STANFORD EXPLORATION PROJECT

Arnaud Berlioux, Dimitri Bevc, Biondo Biondi, Norman Bleistein, Morgan Brown, Nizar Chemingui, Jon Claerbout, Robert Clapp, Joe Dellinger, Christine Ecker, Sergey Fomel, Antoine Guitton, William Harlan, David Lumley, Francis Muir, Marie Prucha, James Rickett, Paul Sava, Yalei Sun, Hector Urdaneta, Louis Vaillant, and Ioan Vlad

Selected papers
# SEP — TABLE OF CONTENTS

## Optimization

*Arnaud Berlioux and William S. Harlan*, Iterative least-square inversion for amplitude balancing.......................... 1

*Sergey Fomel*, Least-square inversion with inexact adjoints. Method of conjugate directions: A tutorial......................... 13

*Sergey Fomel*, Asymptotic pseudounitary stacking operators ............... 27

*Sergey Fomel*, Forward interpolation .................................................. 31

*Jon Claerbout and Sergey Fomel*, Spitz makes a better assumption for the signal PEF ......................................................... 77

*Sergey Fomel*, Seismic reflection data interpolation with differential offset and shot continuation.................................. 85

*Sergey Fomel*, Applications of plane-wave destruction filters .............. 103

*Sergey Fomel*, Inverse B-spline interpolation ................................. 133

## Helix Transform

*Morgan Brown, Jon Claerbout, and Sergey Fomel*, Test case for PEF estimation with sparse data II................................. 163

*Yalei Sun*, Multiple suppression using prediction-error filter ......... 169

*Sergey Fomel and Jon Claerbout*, Multidimensional recursive filter preconditioning in geophysical estimation problems........... 177

*Sergey Fomel and Jon F. Claerbout*, Exploring three-dimensional implicit wavefield extrapolation with the helix transform .......... 199

*Sergey Fomel, Paul Sava, James Rickett, and Jon F. Claerbout*, The Wilson-Burg method of spectral factorization with application to helical filtering............................................. 217

*Paul Sava and Sergey Fomel*, Spectral factorization revisited .......... 237

*Sergey Fomel*, Plane wave prediction in 3-D ................................. 245

*Robert G. Clapp, Sergey Fomel, and Jon Claerbout*, Solution steering with space-variant filters ........................................... 255

*Jon Claerbout*, Random lines in a plane ..................................... 269

*Morgan Brown*, Texture synthesis and prediction error filtering ....... 273
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>James Rickett and Antoine Guittton, Multi-dimensional Fourier transforms in the helical coordinate system</td>
<td>285</td>
</tr>
<tr>
<td><strong>Imaging</strong></td>
<td></td>
</tr>
<tr>
<td>James Rickett and Jon Claerbout, Passive seismic imaging applied to synthetic data</td>
<td>295</td>
</tr>
<tr>
<td>Dimitri Bevc and David E. Lumley, When is anti-aliasing needed in Kirchhoff migration?</td>
<td>303</td>
</tr>
<tr>
<td>Dimitri Bevc, Imaging complex structures with first-arrival traveltimes</td>
<td>313</td>
</tr>
<tr>
<td>Sergey Fomel and Louis Vaillant, Evaluating the Stolt-stretch parameter</td>
<td>323</td>
</tr>
<tr>
<td>Sergey Fomel, Antialiasing of Kirchhoff operators by reciprocal parameterization</td>
<td>339</td>
</tr>
<tr>
<td>Sergey Fomel and Marie Prucha, Angle-gather time migration</td>
<td>355</td>
</tr>
<tr>
<td><strong>Traveltimes</strong></td>
<td></td>
</tr>
<tr>
<td>Hector Urdaneta, Wavefront construction using waverays</td>
<td>365</td>
</tr>
<tr>
<td>Sergey Fomel, Traveltime computation with the linearized eikonal equation</td>
<td>381</td>
</tr>
<tr>
<td>Paul Sava and Sergey Fomel, Huygens wavefront tracing: A robust alternative to conventional ray tracing</td>
<td>389</td>
</tr>
<tr>
<td>Sergey Fomel, A variational formulation of the fast marching eikonal solver</td>
<td>403</td>
</tr>
<tr>
<td>James Rickett and Sergey Fomel, A second-order fast marching eikonal solver</td>
<td>423</td>
</tr>
<tr>
<td><strong>Azimuth Moveout</strong></td>
<td></td>
</tr>
<tr>
<td>Sergey Fomel and Biondo L. Biondi, The time and space formulation of azimuth moveout</td>
<td>429</td>
</tr>
<tr>
<td>Sergey Fomel, Theory of differential offset continuation</td>
<td>441</td>
</tr>
<tr>
<td>Sergey Fomel and Norman Bleistein, Amplitude preservation for offset continuation: Confirmation for Kirchhoff data</td>
<td>473</td>
</tr>
<tr>
<td>Ioan Vlad and Biondo Biondi, Effective AMO implementation in the log-stretch, frequency-wavenumber domain</td>
<td>483</td>
</tr>
<tr>
<td><strong>Velocities</strong></td>
<td></td>
</tr>
<tr>
<td>Jon F. Claerbout, Earthquake stacks at constant offset</td>
<td>493</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Jon Claerbout</td>
<td>A prospect for super resolution</td>
</tr>
<tr>
<td>Joe Dellinger and Francis Muir</td>
<td>The double-elliptic approximation in the group and phase domains</td>
</tr>
<tr>
<td>Sergey Fomel</td>
<td>Velocity continuation and the anatomy of residual prestack time migration</td>
</tr>
<tr>
<td>Sergey Fomel</td>
<td>Velocity continuation by spectral methods</td>
</tr>
<tr>
<td>Sergey Fomel</td>
<td>Time migration velocity analysis by velocity continuation</td>
</tr>
<tr>
<td>James Rickett</td>
<td>Traveltime sensitivity kernels: Banana-doughnuts or just plain bananas?</td>
</tr>
<tr>
<td>Nizar Chemingui</td>
<td>Modeling 3-D anisotropic fractal media</td>
</tr>
</tbody>
</table>

**AVO & Elasticity**

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christine Ecker and David E. Lumley</td>
<td>Seismic AVO analysis of methane hydrate structures</td>
<td>595</td>
</tr>
</tbody>
</table>
Iterative least-square inversion for amplitude balancing

Arnaud Berlioux and William S. Harlan

ABSTRACT
Variations in source strength and receiver amplitude can introduce a bias in the final AVO analysis of prestack seismic reflection data. In this paper we tackle the problem of the amplitude balancing of the seismic traces from a marine survey. We start with a 2-D energy map from which the global trend has been removed. In order to balance this amplitude map, we first invert for the correction coefficients using an iterative least-square algorithm. The coefficients are calculated for each shot position along the survey line, each receiver position in the recording cable, and each offset. Using these coefficients, we then correct the original amplitude map for amplitude variations in the shot, receiver, and offset directions.

INTRODUCTION
In 1994, Mobil provided SEP with a marine dataset on which we were to perform an amplitude variation with offset (AVO) analysis. However, [Berlioux and Lumley (1994)] showed that the amplitude of the traces in the survey present anomalies that need to be preprocessed prior to the AVO analysis.

Fluctuations of the source strength and the receiver amplitudes as well as near-surface irregularities can create amplitude anomalies. It is therefore often necessary to balance the amplitude of each trace in the survey.

To do so we determine the source, receiver, and offset amplitude balancing coefficients by using an iterative least-square algorithm. We then apply these scaling factors to the original 2-D amplitude map to cancel the effect of the impulse response of the defective sources and receivers, and to compensate for irregularities of the sea bottom.

THE PROBLEM OF VARIATIONS OF AMPLITUDE DURING A SURVEY
In a seismic survey, the amplitude recorded at each receiver for each shot depends on the geology of the earth, and on the seismic source and receiver impulse responses.

\[\text{e-mail: arnaud@sep.Stanford.EDU, harlan@sep.Stanford.EDU}\]
Geophysicists and geologists are interested in the variations of the impulse response of the earth. The impulse responses of the receivers and the source may vary, causing anomalous fluctuations of the amplitude recorded during the experiment. It is therefore necessary to correct for such fluctuations, when observed, in order to restore the earth component, which is the valuable information.

Figure 1 shows an amplitude plot in source and offset coordinates for a 2-D seismic survey provided by Mobil in 1994. For each shot and each offset position (i.e., for each trace of the survey) the value of the amplitude has been calculated by taking the root mean square of the trace amplitudes along the time axis. The global trend (low-frequency component of the earth) of the amplitude surface has been estimated by least-square fitting, and removed from the original surface to leave a globally flat 2-D amplitude map (Berlioux and Lumley, 1994).

Figure 1 shows the result after normalization by the root-mean-square value. In this figure the horizontal stripes correspond to offsets where the hydrophone had an impulse response that was weaker (darker stripe) or stronger (brighter stripe) than the average response of the other receivers. Likewise, the vertical stripes indicate where the source had an impulse response that varied from the average. Less obvious, though noticeable, are two other categories of stripes dipping to the left. One is quite visible at the bottom of the plot around the offset -0.5 km, dipping at approximately 10 degrees. An example of the second type of stripe, which dips 20 degrees to the left, is visible at the source position 16 km. Based on the amplitude plot in Figure 1, we have built a model of offset-, source-, midpoint-, and receiver-consistent stripes (Figure 1). Comparing both figures we can identify the first category of dipping stripes as being midpoint-consistent, whereas the less steep stripes are receiver-consistent. The stripes in the receiver directions are broader than the others and therefore not as visible in the source and receiver coordinate system (Figure 3). The stripes in the offset direction follow the descending diagonal in the transformed coordinate system.
Berlioux and Lumley (1994) and Lumley et al. (1995) proposed a method to estimate the source and offset correction coefficients in order to balance the amplitude of each trace in the survey. This method, based on a simple amplitude model, produces good results but does not take into account the receiver-consistent stripes still visible after correction. Because water has a substantially lower velocity than the underlying sediments, the waves travel nearly vertically in the water. The variations of the amplitude caused by the receiver can therefore be associated with near-surface anomalies or irregular sea-bottom topography that affects the receiver recording vertically above it. In the next section, we use a more complex amplitude model, which allows for these variations, and propose an iterative method to estimate the coefficients in order to later correct the amplitude map and balance each trace of the survey.

AN ITERATIVE LEAST-SQUARE INVERSION SCHEME

Our revised amplitude model is

\[ a_{s,h}^{\text{total}} = a_s a_h a_{y=s+h/2} a_{r=s+h} a_{\text{earth}} \]  

(1)

where \( a_{\text{earth}} \) is the earth low-frequency component of the amplitude function; and \( a_s, a_h, a_y, \) and \( a_r \) are the components of the amplitude caused by the source \((s)\), the full offset \((h)\), the midpoint \((y)\), and the receiver \((r)\) variations, respectively.

We now need to invert for the amplitude correction coefficients in order to remove
Figure 3: Portion of the amplitude map in Figure 1 displayed in the source and receiver space. The stripes along the descending diagonal follow the offset direction. Midpoint and receiver stripes are less visible in this coordinate system.
the stripes in Figure 1. To do so, we use the following quadratic objective function

\[ \varphi = \| d(s,h) - a_s a_h a_y a_r \|^2 \]  

(2)

where we assume that the data \( d \) can be modeled as the product of the source, offset, midpoint, and receiver. Normalization allows us to assume \( a_{earth} \approx 1 \).

To estimate the coefficients \( a_s, a_h, a_y, \) and \( a_r \) we choose the Gauss-Seidel algorithm which is an iterative least-square inversion scheme [see Stark (1970)]. We also assume that the coefficients for which we are solving the objective function \( \varphi \) are independent. Therefore, we can get an estimation of one type of coefficient \((a_s, a_h, a_y, \text{or} a_r)\) while keeping the value of the other fixed.

Under these assumptions, after minimizing the objective function with respect to the source coefficients, we derive the following expression, giving the value of the coefficients at iteration \( k \) as a function of the other coefficients at the preceding iteration:

\[ a_s^{(k)} = \frac{\sum_h d^{(k-1)} a_h^{(k-1)} a_y^{(k-1)} a_r^{(k-1)}}{\sum_h \left[ a_h^{(k-1)} a_y^{(k-1)} a_r^{(k-1)} \right]^2} \]  

(3)

We obtain a similar expression for the other three coefficients, where each is expressed as a function of the data and all the other coefficients.

Figures 4, 5, 6, and 7 show the result of the inversion when the algorithm has converged, which required 10 iterations. Comparing the source and offset correction coefficient curves (Figures 4 and 5) with those obtained by Berlioux and Lumley (1994), we can see that the global shape of the curves is the same. The curves in Figures 4 through 7 show identical features: high-frequency variations of the coefficient value around a globally constant value.

The next section shows how we use these estimated correction coefficients to cancel the stripes in the original 2-D amplitude map, and thus balance the traces in the survey.
Figure 5: Estimated offset coefficients.

Figure 6: Estimated midpoint coefficients.

Figure 7: Estimated receiver coefficients.
AMPLITUDE BALANCING

We use the coefficients calculated for the source, offset, and receiver directions to remove the stripes from the plot in Figure 1. Figure 8 shows the 2-D synthetic amplitude map modeled using the source and offset coefficient curves only. The similarity between the stripes in the amplitude plots in Figures 1 and 8 is quite strong.

![Figure 8: Synthetic 2-D amplitude map modeled using the source and offset coefficients.](image)

We divide the original 2-D amplitude function by the source and offset correction coefficients to obtain the amplitude map in Figure 9. In this plot most of the source-and offset-related anomalous amplitude stripes seem to have disappeared, revealing the grey background (the earth impulse response and noise) and other stripes dipping to the left.

![Figure 9: 2-D amplitude map corrected for variations in the source and offset directions using the coefficient curves in Figures 4 and 5.](image)

Comparing Figure 9 and the modeled stripes in Figure 1, we can associate the
now more apparent dipping stripes with the receiver- and midpoint-consistent stripes. In Figure 9, the fine stripes with a steep dip particularly visible at the bottom part of the plot can be regarded as midpoint-consistent, whereas the less steep stripes are receiver-related.

Figure 10 represents a portion of the amplitude map in Figure 9 in the source and receiver coordinate system. The plot shows a bright spot smeared along the offset direction (the descending diagonal). Two very broad darker stripes orthogonal to the receiver axis are visible around the receiver positions at 9.5 km and 12 km. In this coordinate system we can identify these two broad stripes as receiver-consistent. This observation is confirmed by Figure 7 in which the receiver correction coefficient curve shows two local minima at the corresponding receiver location. The same stripes are also noticeable in the center of the plot in Figure 9, though less obvious. They are also clearly visible in Figure 11 which is a synthetic amplitude map modeled from the receiver coefficients only.

Figure 12 shows the amplitude map corrected for variations in the source, offset, and receiver directions. It is the result of the division of the plot in Figure 9 by the estimated receiver coefficients in Figure 7. Figure 13 represents the portion of the corrected amplitude map in Figure 12 in the source and receiver coordinate system. After the correction has been applied, the broad horizontal stripes around the receiver positions at 9.5 km and 12 km have disappeared.

Although midpoint-consistent factors are not used to correct the data amplitudes, we invert for the factors simultaneously to separate the different components more fully. The resulting amplitude map, corrected for anomalous variations in the source, offset, and receiver directions, shows the contribution of the earth and midpoint components to the amplitude of the recorded traces. Some variations may still be present, but not dominant.

**CONCLUSIONS**

Using a more complex amplitude model than that described by Berlioux and Lumley (1994), we have estimated the correction coefficients to balance the trace amplitudes of the Mobil AVO dataset. For this model, we have assumed that the coefficients we are inverting for are independent. We have corrected the original 2-D amplitude map with the source, offset, and receiver coefficients determined by an iterative least-square inversion scheme. This new method gives a more accurate result, taking into account variations of the amplitude caused by the receivers. The stripes in the source, offset, and receiver directions have been successfully removed to reveal the geological grey background of the amplitude plot. We believe the same algorithm can be used for other data sets that present a similar amplitude balancing problem.
Figure 10: Portion of the amplitude map in Figure 9 displayed in the source and receiver space.

[balance/ampl/ srsobalcd]
Figure 11: Synthetic 2-D amplitude map modeled using the receiver coefficients. [balance/ampl/ rvstripes]

Figure 12: 2-D amplitude map corrected for variations in the source, offset, and receiver directions using the coefficient curves in Figures 4, 5, and 7. [balance/ampl/ balanced]
Figure 13: Portion of the corrected amplitude map in Figure 12 displayed in the source and receiver space.
REFERENCES


Least-square inversion with inexact adjoints.  
Method of conjugate directions: A tutorial

Sergey Fomel

ABSTRACT
This tutorial describes the classic method of conjugate directions: the generalization of the conjugate-gradient method in iterative least-square inversion. I derive the algebraic equations of the conjugate-direction method from general optimization principles. The derivation explains the “magic” properties of conjugate gradients. It also justifies the use of conjugate directions in cases when these properties are distorted either by computational errors or by inexact adjoint operators. The extra cost comes from storing a larger number of previous search directions in the computer memory. A simple program and two examples illustrate the method.

INTRODUCTION
This paper describes the method of conjugate directions for solving linear operator equations in Hilbert space. This method is usually described in the numerous textbooks on unconstrained optimization as an introduction to the much more popular method of conjugate gradients. See, for example, Practical optimization by [Gill et al. (1995)] and its bibliography. The famous conjugate-gradient solver possesses specific properties, well-known from the original works of [Hestenes and Stiefel (1952)] and [Fletcher and Reeves (1964)]. For linear operators and exact computations, it guarantees finding the solution after, at most, \( n \) iterative steps, where \( n \) is the number of dimensions in the solution space. The method of conjugate gradients doesn’t require explicit computation of the objective function and explicit inversion of the Hessian matrix. This makes it particularly attractive for large-scale inverse problems, such as those of seismic data processing and interpretation. However, it does require explicit computation of the adjoint operator. [Claerbout (1992, 2003)] shows dozens of successful examples of the conjugate gradient application with numerically precise adjoint operators.

The motivation for this tutorial is to explore the possibility of using different types of preconditioning operators in the place of adjoints in iterative least-square inversion. For some linear or linearized operators, implementing the exact adjoint may pose a difficult problem. For others, one may prefer different preconditioners because of their

---

1e-mail: sergey@sep.stanford.edu
smoothness (Claerbout, 1995a; Crawley, 1995), simplicity (Kleinman and van den Berg, 1991), or asymptotic properties (Sevink and Herman, 1994). In those cases, we could apply the natural generalization of the conjugate gradient method, which is the method of conjugate directions. The cost difference between those two methods is in the volume of memory storage. In the days when the conjugate gradient method was invented, this difference looked too large to even consider a practical application of conjugate directions. With the evident increase of computer power over the last 30 years, we can afford to do it now.

I derive the main equations used in the conjugate-direction method from very general optimization criteria, with minimum restrictions implied. The textbook algebra is illustrated with a simple program and two simple examples.

**IN SEARCH OF THE MINIMUM**

We are looking for the solution of the linear operator equation

\[ d = A m , \]  

(1)

where \( m \) is the unknown model in the linear model space, \( d \) stands for the given data, and \( A \) is the forward modeling operator. The data vector \( d \) belongs to a Hilbert space with a defined norm and dot product. The solution is constructed by iterative steps in the model space, starting from an initial guess \( m_0 \). Thus, at the \( n \)-th iteration, the current model \( m_n \) is found by the recursive relation

\[ m_n = m_{n-1} + \alpha_n s_n , \]  

(2)

where \( s_n \) denotes the step direction, and \( \alpha_n \) stands for the scaling coefficient. The residual at the \( n \)-th iteration is defined by

\[ r_n = d - A m_n . \]  

(3)

Substituting (2) into (3) leads to the equation

\[ r_n = r_{n-1} - \alpha_n A s_n . \]  

(4)

For a given step \( s_n \), we can choose \( \alpha_n \) to minimize the squared norm of the residual

\[ \| r_n \|^2 = \| r_{n-1} \|^2 - 2 \alpha_n (r_{n-1}, A s_n) + \alpha_n^2 \| A s_n \|^2 . \]  

(5)

The parentheses denote the dot product, and \( \| x \| = \sqrt{(x, x)} \) denotes the norm of \( x \) in the corresponding Hilbert space. The optimal value of \( \alpha_n \) is easily found from equation (5) to be

\[ \alpha_n = \frac{(r_{n-1}, A s_n)}{\| A s_n \|^2} . \]  

(6)
Two important conclusions immediately follow from this fact. First, substituting the value of $\alpha_n$ from formula (9) into equation (4) and multiplying both sides of this equation by $r_n$, we can conclude that

$$(r_n, As_n) = 0,$$  \hspace{1cm} (7)

which means that the new residual is orthogonal to the corresponding step in the residual space. This situation is schematically shown in Figure 1. Second, substituting formula (9) into (5), we can conclude that the new residual decreases according to

$$\|r_n\|^2 = \|r_{n-1}\|^2 - \frac{(r_{n-1}, As_n)^2}{\|As_n\|^2},$$  \hspace{1cm} (8)

("Pythagoras’s theorem"), unless $r_{n-1}$ and $As_n$ are orthogonal. These two conclusions are the basic features of optimization by the method of steepest descent. They will help us define an improved search direction at each iteration.

Figure 1: Geometry of the residual in the data space (a scheme).

**IN SEARCH OF THE DIRECTION**

Let’s suppose we have a generator that provides particular search directions at each step. The new direction can be the gradient of the objective function (as in the method of steepest descent), some other operator applied on the residual from the previous step, or, generally speaking, any arbitrary vector in the model space. Let us denote the automatically generated direction by $c_n$. According to formula (8), the residual decreases as a result of choosing this direction by

$$\|r_{n-1}\|^2 - \|r_n\|^2 = \frac{(r_{n-1}, Ac_n)^2}{\|Ac_n\|^2}. $$  \hspace{1cm} (9)

How can we improve on this result?
First step of the improvement

Assuming \( n > 1 \), we can add some amount of the previous step \( s_{n-1} \) to the chosen direction \( c_n \) to produce a new search direction \( s_n^{(n-1)} \), as follows:

\[
s_n^{(n-1)} = c_n + \beta_n^{(n-1)} s_{n-1} ,
\]

(10)

where \( \beta_n^{(n-1)} \) is an adjustable scalar coefficient. According to the fundamental orthogonality principle (7),

\[
(r_{n-1}, A s_{n-1}) = 0.
\]

(11)

As follows from equation (11), the numerator on the right-hand side of equation (9) is not affected by the new choice of the search direction:

\[
(r_{n-1}, A s_n^{(n-1)})^2 = [(r_{n-1}, A c_n) + \beta_n^{(n-1)} (r_{n-1}, A s_{n-1})]^2 = (r_{n-1}, A c_n)^2.
\]

(12)

However, we can use transformation (10) to decrease the denominator in (9), thus further decreasing the residual \( r_n \). We achieve the minimization of the denominator

\[
\|A s_n^{(n-1)}\|^2 = \|A c_n\|^2 + 2 \beta_n^{(n-1)} (A c_n, A s_{n-1}) + (\beta_n^{(n-1)})^2 \|A s_{n-1}\|^2
\]

(13)

by choosing the coefficient \( \beta_n^{(n-1)} \) to be

\[
\beta_n^{(n-1)} = -\frac{(A c_n, A s_{n-1})}{\|A s_{n-1}\|^2}.
\]

(14)

Note the analogy between (15) and (9). Analogously to (7), equation (15) is equivalent to the orthogonality condition

\[
(A s_n^{(n-1)}, A s_{n-1}) = 0.
\]

(15)

Analogously to (8), applying formula (15) is also equivalent to defining the minimized denominator as

\[
\|A c_n^{(n-1)}\|^2 = \|A c_n\|^2 - \frac{(A c_n, A s_{n-1})^2}{\|A s_{n-1}\|^2}.
\]

(16)

Second step of the improvement

Now let us assume \( n > 2 \) and add some amount of the step from the \((n-2)\)-th iteration to the search direction, determining the new direction \( s_n^{(n-2)} \), as follows:

\[
s_n^{(n-2)} = s_n^{(n-1)} + \beta_n^{(n-2)} s_{n-2}.
\]

(17)

We can deduce that after the second change, the value of numerator in equation (9) is still the same:

\[
(r_{n-1}, A s_n^{(n-2)})^2 = [(r_{n-1}, A c_n) + \beta_n^{(n-2)} (r_{n-1}, A s_{n-2})]^2 = (r_{n-1}, A c_n)^2.
\]

(18)
This remarkable fact occurs as the result of transforming the dot product \( \langle \mathbf{r}_{n-1}, \mathbf{A} \mathbf{s}_{n-2} \rangle \) with the help of equation (4):

\[
\langle \mathbf{r}_{n-1}, \mathbf{A} \mathbf{s}_{n-2} \rangle = \langle \mathbf{r}_{n-2}, \mathbf{A} \mathbf{s}_{n-2} \rangle - \alpha_{n-1} \langle \mathbf{A} \mathbf{s}_{n-1}, \mathbf{A} \mathbf{s}_{n-2} \rangle = 0 .
\]

(19)

The first term in (19) is equal to zero according to formula (7); the second term is equal to zero according to formula (15). Thus we have proved the new orthogonality equation

\[
\langle \mathbf{r}_{n-1}, \mathbf{A} \mathbf{s}_{n-2} \rangle = 0 ,
\]

(20)

which in turn leads to the numerator invariance (18). The value of the coefficient \( \beta_{n}^{(n-2)} \) in (17) is defined analogously to (15) as

\[
\beta_{n}^{(n-2)} = -\frac{\langle \mathbf{A} \mathbf{s}_{n}^{(n-1)}, \mathbf{A} \mathbf{s}_{n-2} \rangle}{\| \mathbf{A} \mathbf{s}_{n-2} \|^2} = -\frac{\langle \mathbf{A} \mathbf{c}_{n}, \mathbf{A} \mathbf{s}_{n-2} \rangle}{\| \mathbf{A} \mathbf{s}_{n-2} \|^2} ,
\]

(21)

where we have again used equation (15). If \( \mathbf{A} \mathbf{s}_{n-2} \) is not orthogonal to \( \mathbf{A} \mathbf{c}_{n} \), the second step of the improvement leads to a further decrease of the denominator in (8) and, consequently, to a further decrease of the residual.

**Induction**

Continuing by induction the process of adding a linear combination of the previous steps to the arbitrarily chosen direction \( \mathbf{c}_{n} \) (known in mathematics as the Gram-Schmidt orthogonalization process), we finally arrive at the complete definition of the new step \( \mathbf{s}_{n} \), as follows:

\[
\mathbf{s}_{n} = \mathbf{s}_{n}^{(1)} = \mathbf{c}_{n} + \sum_{j=1}^{j=n-1} \beta_{n}^{(j)} \mathbf{s}_{j} .
\]

(22)

Here the coefficients \( \beta_{n}^{(j)} \) are defined by equations

\[
\beta_{n}^{(j)} = -\frac{\langle \mathbf{A} \mathbf{c}_{n}, \mathbf{A} \mathbf{s}_{j} \rangle}{\| \mathbf{A} \mathbf{s}_{j} \|^2} ,
\]

(23)

which correspond to the orthogonality principles

\[
\langle \mathbf{A} \mathbf{s}_{n}, \mathbf{A} \mathbf{s}_{j} \rangle = 0 , \quad 1 \leq j \leq n - 1
\]

(24)

and

\[
\langle \mathbf{r}_{n}, \mathbf{A} \mathbf{s}_{j} \rangle = 0 , \quad 1 \leq j \leq n .
\]

(25)

It is these orthogonality properties that allowed us to optimize the search parameters one at a time instead of solving the \( n \)-dimensional system of optimization equations for \( \alpha_{n} \) and \( \beta_{n}^{(j)} \).
ALGORITHM

The results of the preceding sections define the method of conjugate directions to consist of the following algorithmic steps:

1. Choose initial model $m_0$ and compute the residual $r_0 = d - A m_0$.
2. At $n$-th iteration, choose the initial search direction $c_n$.
3. If $n$ is greater than 1, optimize the search direction by adding a linear combination of the previous directions, according to equations (22) and (23), and compute the modified step direction $s_n$.
4. Find the step length $\alpha_n$ according to equation (9). The orthogonality principles (24) and (7) can simplify this equation to the form
   \[ \alpha_n = \frac{(r_{n-1}, A c_n)}{\|A s_n\|^2}. \]  
   (26)
5. Update the model $m_n$ and the residual $r_n$ according to equations (2) and (4).
6. Repeat iterations until the residual decreases to the required accuracy or as long as it is practical.

At each of the subsequent steps, the residual is guaranteed not to increase according to equation (8). Furthermore, optimizing the search direction guarantees that the convergence rate doesn’t decrease in comparison with (9). The only assumption we have to make to arrive at this conclusion is that the operator $A$ is linear. However, without additional assumptions, we cannot guarantee global convergence of the algorithm to the least-square solution of equation (1) in a finite number of steps.

WHAT ARE ADJOINTS FOR? THE METHOD OF
CONJUGATE GRADIENTS

The adjoint operator $A^T$ projects the data space back to the model space and is defined by the dot product test
\[ (d, A m) \equiv (A^T d, m) \]  
for any $m$ and $d$. The method of conjugate gradients is a particular case of the method of conjugate directions, where the initial search direction $c_n$ is
\[ c_n = A^T r_{n-1}. \]  
(28)
This direction is often called the gradient, because it corresponds to the local gradient of the squared residual norm with respect to the current model $m_{n-1}$. Aligning the initial search direction along the gradient leads to the following remarkable simplifications in the method of conjugate directions.
Orthogonality of the gradients

The orthogonality principle (25) transforms according to the dot-product test (14) to the form

\[
(r_{n-1}, A s_j) = (A^T r_{n-1}, s_j) = (c_n, s_j) = 0, \quad 1 \leq j \leq n - 1.
\]

(29)

Forming the dot product \((c_n, c_j)\) and applying formula (22), we can see that

\[
(c_n, c_j) = (c_n, s_j - \sum_{i=1}^{i=j-1} \beta_n^{(i)} s_i) = (c_n, s_j) - \sum_{i=1}^{i=j-1} \beta_n^{(i)} (c_n, s_i) = 0, \quad 1 \leq j \leq n - 1.
\]

(30)

Equation (30) proves the orthogonality of the gradient directions from different iterations. Since the gradients are orthogonal, after \(n\) iterations they form a basis in the \(n\)-dimensional space. In other words, if the model space has \(n\) dimensions, each vector in this space can be represented by a linear combination of the gradient vectors formed by \(n\) iterations of the conjugate-gradient method. This is true as well for the vector \(m_0 - m\), which points from the solution of equation (1) to the initial model estimate \(m_0\). Neglecting computational errors, it takes exactly \(n\) iterations to find this vector by successive optimization of the coefficients. This proves that the conjugate-gradient method converges to the exact solution in a finite number of steps (assuming that the model belongs to a finite-dimensional space).

The method of conjugate gradients simplifies formula (26) to the form

\[
\alpha_n = \frac{(r_{n-1}, A c_n)}{\|A s_n\|^2} = \frac{(A^T r_{n-1}, c_n)}{\|A s_n\|^2} = \frac{\|c_n\|^2}{\|A s_n\|^2},
\]

(31)

which in turn leads to the simplification of formula (8), as follows:

\[
\|r_n\|^2 = \|r_{n-1}\|^2 - \frac{\|c_n\|^4}{\|A s_n\|^2}.
\]

(32)

If the gradient is not equal to zero, the residual is guaranteed to decrease. If the gradient is equal to zero, we have already found the solution.

Short memory of the gradients

Substituting the gradient direction (28) into formula (23) and applying formulas (4) and (14), we can see that

\[
\beta_n^{(j)} = \frac{(A c_n, r_j - r_{j-1})}{\alpha_j \|A s_j\|^2} = \frac{(c_n, A^T r_j - A^T r_{j-1})}{\alpha_j \|A s_j\|^2} = \frac{(c_n, c_{j+1} - c_j)}{\alpha_j \|A s_j\|^2}.
\]

(33)

The orthogonality condition (30) and the definition of the coefficient \(\alpha_j\) from equation (31) further transform this formula to the form

\[
\beta_n^{(j)} = 0, \quad 1 \leq j \leq n - 2.
\]

(35)
Equation (35) shows that the conjugate-gradient method needs to remember only the previous step direction in order to optimize the search at each iteration. This is another remarkable property distinguishing that method in the family of conjugate-direction methods.

**PROGRAM**

The program in Table 1 implements one iteration of the conjugate-direction method. It is based upon Jon Claerbout’s cgstep() program and uses an analogous naming convention. Vectors in the data space are denoted by double letters.

In addition to the previous steps \(s_j\) and their conjugate counterparts \(A s_j\) (array \(s\)), the program stores the squared norms \(\|A s_j\|^2\) (variable beta) to avoid recomputation. For practical reasons, the number of remembered iterations can actually be smaller than the total number of iterations.

**EXAMPLES**

**Example 1: Inverse interpolation**

Matthias Schwab has suggested (in a personal communication) an interesting example, in which the cgstep program fails to comply with the conjugate-gradient theory. The inverse problem is a simple one-dimensional data interpolation with a known filter (\(\cdot\)). The known portion of the data is a single spike in the middle. One hundred other data points are considered missing. The known filter is the Laplacian \((1, -2, 1)\), and the expected result is a bell-shaped cubic spline. The forward problem is strictly linear, and the exact adjoint is easily computed by reverse convolution. However, the conjugate-gradient program requires significantly more than the theoretically predicted 100 iterations. Figure 2 displays the convergence to the final solution in three different plots. According to the figure, the actual number of iterations required for convergence is about 300. Figure 3 shows the result of a similar experiment with the conjugate-direction solver cdstep. The number of required iterations is reduced to almost the theoretical one hundred. This indicates that the orthogonality of directions implied in the conjugate-gradient method has been distorted by computational errors. The additional cost of correcting these errors with the conjugate-direction solver comes from storing the preceding 100 directions in memory. A smaller number of memorized steps produces smaller improvements.

**Example 2: Velocity transform**

The next test example is the velocity transform inversion with a CMP gather from the Mobil AVO dataset (Nichols, 1994; Lumley et al., 1994; Lumley, 1994). I use Jon
void sf_cdstep (bool forget /* restart flag */,
    int nx /* model size */,
    int ny /* data size */,
    float* x /* current model [nx] */,
    const float* g /* gradient [nx] */,
    float* rr /* data residual [ny] */,
    const float* gg /* conjugate gradient [ny] */)
/*< Step of conjugate-direction iteration.
   The data residual is rr = A x - dat
   */
{
    float *s, *si, *ss;
    double alpha, beta;
    int i, n, ix, iy;

    s = sf_floatalloc (nx+ny);
    ss = s+nx;

    for (ix=0; ix < nx; ix++) { s[ix] = g[ix]; }
    for (iy=0; iy < ny; iy++) { ss[iy] = gg[iy]; }

    sf_llist_rewind (steps);
    n = sf_llist_depth (steps);

    for (i=0; i < n; i++) {
        sf_llist_down (steps, &si, &beta);
        alpha = - cblas_dsdot (ny, gg, 1, si+nx, 1) / beta;
        cblas_saxpy (nx+ny, alpha, si, 1, s, 1);
    }

    beta = cblas_dsdot (ny, s+nx, 1, s+nx, 1);
    if (beta < DBL_EPSILON) return;

    sf_llist_add (steps, s, beta);
    if (forget) sf_llist_chop (steps);
    alpha = - cblas_dsdot (ny, rr, 1, ss, 1) / beta;
    cblas_saxpy (nx, alpha, s, 1, x, 1);
    cblas_saxpy (ny, alpha, ss, 1, rr, 1);
}

Table 1: The source of this program is RSF/api/c/cdstep.c
Figure 2: Convergence of the missing data interpolation problem with the conjugate-gradient solver. Current models are plotted against the number of iterations. The three plots are different displays of the same data.

Figure 3: Convergence of the missing data interpolation problem with the long-memory conjugate-direction solver. Current models are plotted against the number of iterations. The three plots are different displays of the same data.
Claerbout's veltran program (Claerbout 1995b) for anti-aliased velocity transform with rho-filter preconditioning and compare three different pairs of operators for inversion. The first pair is the CMP stacking operator with the “migration” weighting function \( w = \frac{\tau_0}{\sqrt{\tau}} \) and its adjoint. The second pair is the “pseudo-unitary” velocity transform with the weighting proportional to \( \sqrt{|s \times x|} \), where \( x \) is the offset and \( s \) is the slowness. These two pairs were used in the velocity transform inversion with the iterative conjugate-gradient solver. The third pair uses the weight proportional to \(|x|\) for CMP stacking and \(|s|\) for the reverse operator. Since these two operators are not exact adjoints, it is appropriate to apply the method of conjugate directions for inversion. The convergence of the three different inversions is compared in Figure 4. We can see that the third method reduces the least-square residual error, though it has a smaller effect than that of the pseudo-unitary weighting in comparison with the uniform one. The results of inversion after 10 conjugate-gradient iterations are plotted in Figures 5 and 6, which are to be compared with the analogous results of Lumley (1994) and Nichols (1994).

CONCLUSIONS

The conjugate-gradient solver is a powerful method of least-square inversion because of its remarkable algebraic properties. In practice, the theoretical basis of conjugate gradients can be distorted by computational errors. In some applications of inversion,
Figure 5: Input CMP gather (left) and its velocity transform counterpart (right) after 10 iterations of conjugate-direction inversion.

Figure 6: The modeled CMP gather (left) and the residual data (right) plotted at the same scale.
we may want to do that on purpose, by applying inexact adjoints in preconditioning. In both cases, a safer alternative is the method of conjugate directions. Jon Claerbout’s cgstep() program actually implements a short-memory version of the conjugate-direction method. Extending the length of the memory raises the cost of iterations, but can speed up the convergence.

REFERENCES

Asymptotic pseudounitary stacking operators

Sergey Fomel

ABSTRACT

Stacking operators are widely used in seismic imaging and seismic data processing. Examples include Kirchhoff datuming, migration, offset continuation, DMO, and velocity transform. Two primary approaches exist for inverting such operators. The first approach is iterative least-squares optimization, which involves the construction of the adjoint operator. The second approach is asymptotic inversion, where an approximate inverse operator is constructed in the high-frequency asymptotics. Adjoint and asymptotic inverse operators share the same kinematic properties, but their amplitudes (weighting functions) are defined differently. This paper describes a theory for reconciling the two approaches. I introduce a pair of the asymptotic pseudo-unitary operators, which possess both the property of being adjoint and the property of being asymptotically inverse. The weighting function of the asymptotic pseudo-unitary stacking operators is shown to be completely defined by the derivatives of the operator kinematics. I exemplify the general theory by considering several particular examples of stacking operators. Simple numerical experiments demonstrate a noticeable gain in efficiency when the asymptotic pseudo-unitary operators are applied for preconditioning iterative least-squares optimization.

INTRODUCTION

Integral (stacking) operators play an important role in seismic imaging and seismic data processing. The most common applications are common midpoint stacking, Kirchhoff migration, and dip moveout. Other examples include (listed in random order) Kirchhoff datuming, back-projection tomography, slant stack, velocity transform, offset continuation, and azimuth moveout. The use of the integral methods increases in prestack three-dimensional processing because of their flexibility with respect to irregularities in the data geometry.

An integral operator often is used to represent the forward modeling problem, and we invert it to solve for the model. In this paper, I consider two different approaches to inversion. The first is least-squares inversion, which requires constructing the adjoint counterpart of the modeling operator. The second approach is asymptotic inversion, which aims at reconstructing the high-frequency (discontinuous) parts of the model.
I compare the two approaches and introduce the notion of asymptotic pseudo-unitary operator pair that ties them together.

In practice, least squares inversion is often applied as an iterative process (Ronen and Liner, 2000). The advantage of connecting it with the asymptotic inverse theory is the ability to speed up the iteration. This approach was used, in the context of seismic migration, by Jin et al. (1992) and Lambaré et al. (1992). Asymptotic pseudo-unitary operators, introduced in this paper, provide a more universal theoretical tool. One can use them to construct an appropriate preconditioning operator for accelerating the convergence of the least-squares methods.

The first part of this paper contains a formal definition of a stacking operator and reviews the theory of asymptotic inversion, following the fundamental results of Beylkin (1985) and Goldin (1988, 1990). According to this theory, the high-frequency asymptotic inverse of a stacking operator is also a stacking operator. To connect this theory with the theory of adjoint operators, I show that the adjoint of a stacking operator can also be included in the class of stacking operators. The adjoint operator has the same summation path as the asymptotic inverse but a different weighting function. These two results combine together to form the definition of asymptotic pseudo-unitary integral operators. I apply such operators to define a general preconditioning operator for least-squares inversion. While one can apply Beylkin’s theory directly for constructing an appropriate asymptotic preconditioner, pseudo-unitary operators accomplish the job in a more straightforward and computationally attractive way.

The second part of the paper addresses such examples of commonly used stacking operators as wave-equation datuming, migration, velocity transform, and offset continuation. The theory is specified for these particular applications and accompanied by numerical examples. The examples demonstrate the practical advantages of asymptotic pseudo-unitary operators.

**THEORETICAL DEFINITION OF A STACKING OPERATOR**

In practice, integration of discrete data is performed by stacking. In theory, it is convenient to represent a stacking operator in the form of a continuous integral:

\[
S(t, y) = A[M(z, x)] = \int_{\Omega} w(x; t, y) M(\theta(x; t, y), x) \, dx .
\] (1)

Function \(M(z, x)\) is the input of the operator, \(S(t, y)\) is the output, \(\Omega\) is the summation aperture, \(\theta\) represents the summation path, and \(w\) stands for the weighting function. The range of integration (the operator aperture) may also depend on \(t\) and \(y\). Allowing \(x\) to be a two-dimensional variable, we can use definition (1) to represent an operator applied to three-dimensional data. Throughout this paper, I assume that
t and z belong to a one-dimensional space, and that x and y have the same number of dimensions.

The goal of inversion is to reconstruct some function \( \hat{M}(z, x) \) for a given \( S(t, y) \), so that \( \hat{M} \) is in some sense close to \( M \) in equation (1).

**ASYMPTOTIC INVERSION: RECONSTRUCTING THE DISCONTINUITIES**

Mathematical analysis of the inverse problem for operator (1) shows that only in rare cases can we obtain an analytically exact inversion. A well-known example is the Radon transform, which has acquired a lot of different aliases in geophysical literature: slant stack, tau-p transform, plane wave decomposition, and controlled directional reception (CDR) transform (Gardner and Lu, 1991). In this case,

\[
\theta(x; t, y) = t + x y, \tag{2}
\]

\[
w(x; t, y) = 1. \tag{3}
\]

Radon obtained a result similar to the theoretical inversion of operator (1) with the summation path (2) and the weighting function (3) in 1917, but his result was not widely known until the development of computer tomography. According to Radon (1917), the inverse operator has the form

\[
M(z, x) = A^{-1}[S(t, y)] = |D|^m \int \hat{w} S(\hat{\theta}(y; z, x), y) \, dy, \tag{4}
\]

where

\[
\hat{\theta}(y; z, x) = z - x y, \tag{5}
\]

\[
\hat{w} = \frac{1}{(2 \pi)^m}. \tag{6}
\]

\(|D|\) is a one-dimensional convolution operator with the spectrum \(|\omega|\):

\[
|D|[U(z, x)] = \frac{1}{2 \pi} \int U(\xi, x) \int |\omega| e^{i \omega(z-\xi)} d\omega \, d\xi, \tag{7}
\]

and \( m \) is the dimensionality of \( x \) and \( y \) (usually 1 or 2). In Russian geophysical literature, a similar result for the inversion of the CDR transform was published by Nakhamkin (1969).

Extension of Radon’s result to the general form of integral operator (1) (generalized Radon transform) is possible via asymptotic analysis of the inverse problem. In the general case, Beylkin (1985) and Goldin (1988) have shown that asymptotic inversion can reconstruct discontinuous parts of the model. These are the parts responsible for the asymptotic behavior of the model at high frequencies. Since the discontinuities are
associated with wavefronts and reflection events at seismic sections, there is a certain correspondence between asymptotic inversion and such standard goals of seismic data processing as kinematic equivalence and amplitude preservation.

The main theorem of asymptotic inversion can be formulated as follows (Goldin, 1988). The leading-order discontinuities in $M$ are reconstructed by an integral operator of the form

$$\hat{M}(z, x) = \hat{A}[S(t, y)] = |D|^m \int \hat{w}(y; z, x) S(\hat{\theta}(y; z, x), y) \, dy,$$

where the summation path $\hat{\theta}$ is obtained simply by solving the equation

$$z = \theta(x; t, y)$$

for $t$ (if such an explicit solution is possible). The correctly chosen summation path reconstructs the geometry of the discontinuities. To recover the amplitude, we must choose the correct weighting function, which is constrained by the equation (Beylkin, 1985; Goldin, 1988)

$$w \hat{w} = \frac{1}{(2\pi)^m} \sqrt{|F\hat{F}|} \left| \frac{\partial \hat{\theta}}{\partial z} \right|^m,$$

where

$$F = \frac{\partial \theta}{\partial t} \frac{\partial^2 \theta}{\partial x \partial y} - \frac{\partial \theta}{\partial y} \frac{\partial^2 \theta}{\partial x \partial t},$$

$$\hat{F} = \frac{\partial \hat{\theta}}{\partial z} \frac{\partial^2 \hat{\theta}}{\partial x \partial y} - \frac{\partial \hat{\theta}}{\partial y} \frac{\partial^2 \hat{\theta}}{\partial x \partial z}.$$  

The solution assumes that differential forms $F$ and $\hat{F}$ exist and are bounded and non-vanishing. In the multi-dimensional case ($m \geq 2$), they are replaced by the determinants of the corresponding matrices. To ensure the asymptotic inversion, equation (10) must be satisfied at least in the vicinity of the stationary points of integral (1). Those are the points where the summation path of the form (9) is tangent to the traveltimes of the actual events on the transformed model. In the case of the Radon transform, $|F\hat{F}| = |\frac{\partial \hat{\theta}}{\partial z}| = 1$, and the asymptotic inverse coincides with the exact inversion.

### LEAST-SQUARES INVERSION AND ADJOINT OPERATORS

Least-squares inversion is widely used in practice not only because it is applicable even when the asymptotic results are unavailable but also because of its ability to

---

2This requirement is related to the requirement for the normal $A^T A$ operator, introduced in the next section, to be a pseudo-differential operator (Wong, 1991). Situations where this condition is violated require a special consideration (Nolan and Symes, 1996; Stolk, 2000).
handle finite sampling effects that are difficult to handle in asymptotic theory (Ronen and Liner, 2000).

The theoretical least-squares inverse of operator (1) has the well-known form (Tarantola, 1987)

\[
\tilde{M}(z, x) = \tilde{A}[S(t, y)] = (A^T A)^\dagger A^T [S(t, y)] ,
\]

(13)

where \(\dagger\) denotes pseudo-inverse, and the adjoint operator \(A^T\) is defined by the dot-product test:

\[
(S(t, y), A[M(z, x)]) \equiv (A^T [S(t, y)], M(z, x)) .
\]

(14)

With a specified definition of the dot-product, the generalized inverse minimizes the following quantity, which is the squared \(L_2\) norm of the residual:

\[
(S(t, y) - A[M(z, x)], S(t, y) - A[M(z, x)]) .
\]

(15)

In the case of integral operators, a natural definition of the dot-product is the double integral

\[
(S_1(t, y), S_2(t, y)) = \iint S_1(t, y) S_2(t, y) \, dy \, dt ,
\]

(16)

\[
(M_1(z, x), M_2(z, x)) = \iint M_1(z, x) M_2(z, x) \, dx \, dz .
\]

(17)

The notion of the adjoint operator completely depends on the arbitrarily chosen definition of the dot product and norm in the model and data spaces. A simple way to change those definitions is to find some positive weights \(W_M(z, x)\) in the model space and \(W_S(t, y)\) in the data space that define the dot products as follows:

\[
(S_1(t, y), S_2(t, y)) = \iint W_S(t, y) S_1(t, y) S_2(t, y) \, dy \, dt ,
\]

(18)

\[
(M_1(z, x), M_2(z, x)) = \iint W_M(z, x) M_1(z, x) M_2(z, x) \, dx \, dz .
\]

(19)

To formally define the adjoint of a stacking operator, let us substitute the definition of the stacking operator (1) into the dot product (14), as follows:

\[
(S(t, y), A[M(z, x)]) = \iint w(x; t, y) M(\theta(x; t, y), x) S(t, y) \, dx \, dy \, dt .
\]

(20)

Assuming that the function \(\theta\) is monotone in \(t\) \(3\) we can change the integration variable \(t\) to \(z = \theta(x; t, y)\) and rewrite equation (20) in the form

\[
(S(t, y), A[M(z, x)]) = \iint \tilde{w}(y; z, x) M(z, x) S(\tilde{\theta}(y; z, x), x) \, dy \, dz .
\]

(21)

\(^3\)If this is not the case, a different parameterization of the stacking function is appropriate (Fomel, 2001a).
where $\hat{\theta}$ has the same meaning as in equation (8), and

$$\tilde{w}(y; z, x) = w(x; \hat{\theta}(y; z, x), y) \left. \frac{\partial \hat{\theta}}{\partial z} \right|_{z}. \tag{22}$$

Comparing equations (21) and (14), we conclude that the adjoint operator $A^T$ is defined by the equality

$$A^T[S(t, y)] = \int \tilde{w}(y; z, x) S(\hat{\theta}(y; z, x), y) \, dy. \tag{23}$$

Thus we have proven that the continuous adjoint of a stacking operator is another stacking operator. The adjoint operator has the same summation path as the asymptotic inverse (8), which guarantees the correct reconstruction of the kinematics of the input wavefield. The amplitude (weighting function) of the adjoint operator is directly proportional to the forward weighting according to equation (22). The coefficient of proportionality is the Jacobian of the transformation of the variables $z$ and $t$.

Similar results have been obtained for particular cases of stacking operators: velocity transform (Thorson, 1984; Jedlicka, 1989), Kirchhoff constant-velocity migration (Ji, 1994), and NMO (Crawley, 1995). In the appendix, I exemplify an application of least-squares inversion by reviewing inversion of the Radon operator and showing that it is precisely equivalent to the asymptotic result of the previous section.

**ASYMPTOTIC PSEUDO-UNITARY OPERATOR PAIR**

According to the theory of asymptotic inversion, briefly reviewed in the first part of this paper, the weighting function of the asymptotically inverse operator is inversely proportional to the weighting of the forward operator. On the other hand, the weighting in the adjoint is directly proportional to the forward weighting. This difference allows us to define a hybrid pair of operators that possess both the property of being adjoint and the property of being asymptotic inverse. It is appropriate to call a pair of operators defined in this way asymptotic pseudo-unitary. The definition of asymptotic pseudo-unitary operators follows directly from the combination of definitions (8) and (23). Splitting the derivative operator $|D|$ in (8) into the product of two operators, we can write the forward operator as

$$S(t, y) = A[M(z, x)] = \int w^{(+)}(x; t, y) |D|^{m/2} M(\theta(x; t, y), x) \, dx \tag{24}$$

and its asymptotic pseudo-unitary adjoint as

$$\tilde{M}(z, x) = \tilde{A}[S(t, y)] = |D|^{m/2} \int w^{(-)}(y; z, x) S(\hat{\theta}(y; z, x), y) \, dy. \tag{25}$$
According to equation [10],

\[
    w^{(+)} w^{(-)} = \frac{1}{(2 \pi)^m} \sqrt{|F \hat{F}|} \left| \frac{\partial \hat{\theta}}{\partial z} \right|^m.
\]

(26)

According to equation [22],

\[
    w^{(-)} = w^{(+)} \left| \frac{\partial \hat{\theta}}{\partial z} \right|^\frac{m}{2}.
\]

(27)

Combining equations (26) and (27) uniquely determines both weighting functions, as follows:

\[
    w^{(+)} = \frac{1}{(2 \pi)^{m/2}} \left| F \hat{F} \right|^{1/4} \left| \frac{\partial \hat{\theta}}{\partial z} \right|^{(m-2)/4},
\]

(28)

\[
    w^{(-)} = \frac{1}{(2 \pi)^{m/2}} \left| F \hat{F} \right|^{1/4} \left| \frac{\partial \hat{\theta}}{\partial z} \right|^{(m+2)/4}.
\]

(29)

Equations (28) and (29) complete the definition of asymptotic pseudo-unitary operator pair.

The notion of pseudo-unitary operators is directly applicable in the situations where we can arbitrarily construct both forward and inverse operators. One example of such a situation is the velocity transform considered in the next section of this paper. In the more common case, the forward operator is strictly defined by the physics of a problem. In this case, we can include asymptotic inversion in the iterative least-squares inversion by means of preconditioning [Jin et al., 1992; Lambaré et al., 1992]. The linear preconditioning operator should transform the forward stacking-type operator to the form (24) with the weighting function (28). Theoretically, this form of preconditioning should lead to the fastest convergence of the iterative least-squares inversion with respect to the high-frequency parts of the model.

If the forward pseudo-unitary operator \( A_p \) can be related to the forward modeling operator \( A_m \) as \( A_p = W_s A_m W_m \), where \( W_s \) and \( W_m \) are weighting operators in the data and model domains correspondingly, then preconditioning simply amounts to replacing the least-squares equation

\[
    S \approx A_m [M]
\]

(30)

with the equation

\[
    W_s [S] \approx W_s A_m W_m [P] = A_p [P],
\]

(31)

where \( P \) is the preconditioned model. The advantage of using equation (31) is in the the fact that the normal operator \( A_p^T A_p \) is closer (asymptotically) to identity and therefore should be easier to invert than the original operator \( A_m^T A_m \) in the least-squares solution (17).
EXAMPLES

In this section, I consider several particular examples of stacking operators used in seismic data processing and derive their asymptotic pseudo-unitary versions.

Datuming

Let \( x \) denote a point on the surface at which the propagating wavefield is recorded. Let \( y \) denote a point on another surface, to which the wavefield is propagating. Then the summation path of the stacking operator for the forward wavefield continuation is

\[
\theta(x; t, y) = t - T(x, y),
\]

where \( t \) is the time recorded at the \( y \)-surface, and \( T(x, y) \) is the traveltime along the ray connecting \( x \) and \( y \). The backward propagation reverses the sign in (32), as follows:

\[
\hat{\theta}(y; z, x) = z + T(x, y) .
\]

Substituting the summation path formulas (32) and (33) into the general weighting function formulas (28) and (29), we immediately obtain

\[
w^{(+)} = w^{(-)} = \frac{1}{(2\pi)^{m/2}} \left| \frac{\partial^2 T}{\partial x \partial y} \right|^{1/2} .
\]

Gritsenko’s formula (Gritsenko, 1984; Goldin, 1986) states that the second mixed traveltime derivative \( \frac{\partial^2 T}{\partial x \partial y} \) is connected with the geometric spreading \( R \) along the \( x-y \) ray by the equality

\[
R(x, y) = \frac{\sqrt{\cos \alpha(x) \cos \alpha(y)}}{v(x)} \left| \frac{\partial^2 T}{\partial x \partial y} \right|^{-1/2} ,
\]

where \( v(x) \) is the velocity at the point \( x \), and \( \alpha(x) \) and \( \alpha(y) \) are the angles formed by the ray with the \( x \) and \( y \) surfaces, respectively. In a constant-velocity medium,

\[
R(x, y) = v^{m-1} T(x, y)^{m/2} .
\]

Gritsenko’s formula (35) allows us to rewrite equation (34) in the form (Goldin, 1988)

\[
w^{(+)}(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{\sqrt{\cos \alpha(x) \cos \alpha(y)}}{v(x) R(x, y)},
\]

\[
w^{(-)}(y; z, x) = \frac{1}{(2\pi)^{m/2}} \frac{\sqrt{\cos \alpha(x) \cos \alpha(y)}}{v(y) R(y, x)} .
\]
The weighting functions commonly used in Kirchhoff datuming (Berryhill, 1979; Wiggins, 1984; Goldin, 1985) are defined as

\[
w(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{\cos \alpha(x)}{v(x) R(x, y)}, \quad (39)\\
\]

\[
\hat{w}(y; z, x) = \frac{1}{(2\pi)^{m/2}} \frac{\cos \alpha(y)}{v(y) R(y, x)}. \quad (40)
\]

These two operators appear to be asymptotically inverse according to formula (10). They coincide with the asymptotic pseudo-unitary operators if the velocity \( v \) is constant \( (v(x) = v(y)) \), and the two datum surfaces are parallel \( (\alpha(x) = \alpha(y)) \).

**Migration**

*Least-squares* migration, envisioned by Lailly (1984) and Tarantola (1984), has recently become a practical method and gained a lot of attention in the geophysical literature (Nemeth et al., 1999; Chavent and Plessix, 1999; Duquet and Marfurt, 1999; Fomel et al., 2002). Using the theory of asymptotic pseudo-unitary operators allows us to reconcile this approach with the method of *asymptotic true-amplitude* migration (Bleistein et al., 2001).

As recognized by Tygel et al. (1994), true-amplitude migration (Goldin, 1992; Schleicher et al., 1993) is the asymptotic inversion of seismic modeling represented by the Kirchhoff high-frequency approximation. The Kirchhoff approximation for a reflected wave (Haddon and Buchen, 1981; Bleistein, 1984) belongs to the class of stacking-type operators (1) with the summation path

\[
\theta(x; t, y) = t - T(s(y), x) - T(x, r(y)), \quad (41)
\]

the weighting function

\[
w(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{C(s(y), x, r(y))}{R(s(y), x) R(x, r(y))}, \quad (42)
\]

and the additional time filter \( \left( \frac{\partial}{\partial z} \right)^{m/2} \). Here \( x \) denotes a point at the reflector surface, \( s \) is the source location, and \( r \) is the receiver location at the observation surface. The parameter \( y \) corresponds to the configuration of observation. That is, \( s(y) = s, \quad r(y) = y \) for the common-shot configuration, \( s(y) = r(y) = y \) for the zero-offset configuration, and \( s(y) = y - h, \quad r(y) = y + h \) for the common-offset configuration (where \( h \) is the half-offset). The functions \( T \) and \( R \) have the same meaning as in the datuming example, representing the one-way traveltime and the one-way geometric spreading, respectively. The function \( C(s, x, r) \) is known as the *obliquity factor*. Its definition is

\[
C(s, x, r) = \frac{1}{2} \left( \frac{\cos \alpha_s(x)}{v_s(x)} + \frac{\cos \alpha_r(x)}{v_r(x)} \right), \quad (43)
\]
where the angles $\alpha_s(x)$ and $\alpha_r(x)$ are formed by the incident and reflected waves with the normal to the reflector at the point $x$, and $v_s(x)$ and $v_r(x)$ are the corresponding velocities in the vicinity of this point. In this paper, I leave the case of converted (e.g., P-SV) waves outside the scope of consideration and assume that $v_s(x)$ equals $v_r(x)$ (e.g., in P-P reflection). In this case, it is important to notice that at the stationary point of the Kirchhoff integral, $\alpha_s(x) = \alpha_r(x) = \alpha(x)$ (the law of reflection), and therefore

$$C(s, x, r) = \frac{\cos \alpha(x)}{v(x)}. \quad (44)$$

The stationary point of the Kirchhoff integral is the point where the stacking curve (41) is tangent to the actual reflection traveltime curve. When our goal is asymptotic inversion, it is appropriate to use equation (44) in place of (43) to construct the inverse operator. The weighted function (42) can include other factors affecting the leading-order (WKBJ) ray amplitude, such as the source signature, caustics counter (the KMAH-index), and transmission coefficient for the interfaces (Chapman and Drummond, 1982; Červený, 2001). In the following analysis, I neglect these factors for simplicity.

The model $M$ implied by the Kirchhoff modeling integral is the wavefield with the wavelet shape of the incident wave and the amplitude proportional to the reflector coefficient along the reflector surface. The goal of true-amplitude migration is to recover $M$ from the observed seismic data. In order to obtain the image of the reflectors, the reconstructed model is evaluated at the time $z$ equal to zero. The Kirchhoff modeling integral requires explicit definition of the reflector surface. However, its inverse doesn’t require explicit specification of the reflector location. For each point of the subsurface, one can find the normal to the hypothetical reflector by bisecting the angle between the $s-x$ and $x-r$ rays. Born scattering approximation provides a different physical model for the reflected waves. According to this approximation, the recorded waves are viewed as scattered on smooth local inhomogeneities rather than reflected from sharp reflector surfaces. The inversion of Born modeling (Miller et al., 1987; Bleistein, 1987) closely corresponds with the result of Kirchhoff integral inversion. For an unknown reflector and the correct macro-velocity model, the asymptotic inversion reconstructs the signal located at the reflector surface with the amplitude proportional to the reflector coefficient.

As follows from the form of the summation path (41), the integral migration operator must have the summation path

$$\hat{\theta}(y; z, x) = z + T(s(y), x) + T(x, r(y)) \quad (45)$$

to reconstruct the geometry of the reflector at the migrated section. According to equation (8), the asymptotic reconstruction of the wavelet requires, in addition, the derivative filter $(-\partial^m \partial z)^{m/2}$. The asymptotic reconstruction of the amplitude defines the true-amplitude weighting function in accordance with equation (10), as follows:

$$\hat{w}(y; z, x) = \frac{v(x)R(s(y), x)R(x, r(y))}{(2\pi)^{m/2}\cos \alpha(x)} \left| \frac{\partial^2 T(s(y), x)}{\partial x \partial y} + \frac{\partial^2 T(x, r(y))}{\partial x \partial y} \right|. \quad (46)$$
The weighting function of the asymptotic pseudo-unitary migration is found analogously to equation (34) as

\[ w^{(+)} = w^{(-)} = \frac{1}{(2\pi)^{m/2}} \left| \frac{\partial^2 T(s(y),x)}{\partial x \partial y} + \frac{\partial^2 T(x,r(y))}{\partial x \partial y} \right|^{1/2}. \]  

Unlike true-amplitude migration, this type of migration operator does not change the dimensionality of the input. Several specific cases exist for different configurations of the input data.

1. Common-shot migration

In the case of common-shot migration, we can simplify equation (46) with the help of Gritsenko’s formula (35) to the form

\[ \hat{w}_{CS}(r; z, x) = \frac{1}{(2\pi)^{m/2}} \frac{\cos \alpha(r)}{v(x)} \frac{R(s, x)}{R(x, r)} = \frac{1}{(2\pi)^{m/2}} \frac{\cos \alpha(r)}{v(r)} \frac{R(s, x)}{R(r, x)}, \]  

where the angle \( \alpha(r) \) is measured between the reflected ray and the normal to the observation surface at the reflector point \( r \). Formula (48) coincides with the analogous result of Keho and Beydoun (1988), derived directly from Claerbout’s imaging principle (Claerbout, 1970). An alternative derivation is given by Goldin (1987). Docherty (1991) points out a remarkable correspondence between this formula and the classic results of Born scattering inversion (Bleistein, 1987).

For common-shot migration, pseudo-unitary weighting coincides with the weighting of datuming and corresponds to the downward continuation of the receivers.

2. Zero-offset migration

In the case of zero-offset migration, Gritsenko’s formula simplifies the true-amplitude migration weighting function (46) to the form

\[ \hat{w}_{ZO}(y; z, x) = \frac{2^m}{(2\pi)^{m/2}} \frac{\cos \alpha(y)}{v(y)} . \]  

In a constant-velocity medium, one can accomplish the true-amplitude zero-offset migration by premultiplying the recorded zero-offset seismic section by the factor \( (\frac{v}{2})^{m-1} \left( \frac{1}{2} \right)^{m/2} \) [which corresponds at the stationary point to the geometric spreading \( R(x, y) \)] and downward continuation according to formula (40) with the effective velocity \( \frac{v}{2} \) (Goldin, 1987; Hubral et al., 1991). This conclusion is in agreement with the analogous result of Born inversion (Bleistein et al., 1985), though derived from a different viewpoint.

In the zero-offset case, the pseudo-unitary forward operator reduces to downward pseudo-unitary continuation with a velocity of \( \frac{v}{2} \).
3. Common-offset migration

In the case of common-offset migration in a general variable-velocity medium, the weighting function (46) cannot be simplified to a different form, and all its components need to be calculated explicitly by dynamic ray tracing (Červený and de Castro, 1993). In the constant-velocity case, we can differentiate the explicit expression for the summation path

\[
\hat{\theta}(y; z, x) = z + \frac{\rho_s(x, y) + \rho_r(x, y)}{v},
\]

where \(\rho_s\) and \(\rho_r\) are the lengths of the incident and reflected rays:

\[
\rho_s(y, x) = \sqrt{x_3^2 + (x_1 - y_1 + h_1)^2 + (x_2 - y_2 + h_2)^2},
\]

\[
\rho_r(y, x) = \sqrt{x_3^2 + (x_1 - y_1 - h_1)^2 + (x_2 - y_2 - h_2)^2}.
\]

For simplicity, the vertical component of the midpoint \(y_3\) is set here to zero. Evaluating the second derivative term in formula (46) for the common-offset geometry leads, after some heavy algebra, to the expression

\[
\left| \frac{\partial^2 T(s(y), x)}{\partial x \partial y} + \frac{\partial^2 T(x, r(y))}{\partial x \partial y} \right| = \frac{x_3 (\rho_s^2 + \rho_r^2)}{v \rho_s \rho_r^2} \left( \frac{\rho_s + \rho_r}{v \rho_s \rho_r^2} \right)^{m-1} \cos \alpha(x).
\]

Substituting (53) into the general formula (46) yields the weighting function for the common-offset true-amplitude constant-velocity migration:

\[
\hat{w}_{CO}(y; z, x) = \frac{1}{(2\pi)^{m/2}} \frac{x_3 (\rho_s + \rho_r)^{m-1} (\rho_s^2 + \rho_r^2)}{v (\rho_s \rho_r)^{m/2+1}}.
\]

Equation (54) is similar to the result obtained by Sullivan and Cohen (1987). In the case of zero offset \(h = 0\), it reduces to equation (49). Note that the value of \(m = 1\) in (54) corresponds to the two-dimensional (cylindric) waves recorded on the seismic line. A special case is the 2.5-D inversion, when the waves are assumed to be spherical, while the recording is on a line, and the medium has cylindric symmetry. In this case, the modeling weighting function (42) transforms to (Deregowski and Brown, 1983; Bleistein, 1986)

\[
w(x; t, y) = \frac{1}{(2\pi)^{1/2}} \frac{\sqrt{v} C(s(y), x, r(y))}{\sqrt{\rho_s \rho_r (\rho_s + \rho_r)}},
\]

and the time filter is \(\left(\frac{\rho_s \rho_r}{v}\right)^{1/2}\). Combining this result with formula (53) for \(m = 1\), we obtain the weighting function for the 2.5-D common-offset migration in a constant velocity medium (Sullivan and Cohen, 1987):

\[
\hat{w}_{CO:2.5D}(y; z, x) = \frac{1}{(2\pi)^{1/2}} \frac{x_3 \sqrt{\rho_s + \rho_r (\rho_s^2 + \rho_r^2)}}{\sqrt{v} (\rho_s \rho_r)^{3/2}}.
\]
The corresponding time filter for 2.5-D migration is \((-\frac{\partial}{\partial t})^{1/2}\).

In the common-offset case, the pseudo-unitary weighting is defined from (47) and (53) as follows:

\[ w_{CO}^{(-)}(y; z, x) = \frac{1}{(2\pi v)^{m/2}} \sqrt{x_3 \cos \alpha \left(\rho_s + \rho_r\right)^{m+1}} \sqrt{\rho_s^2 + \rho_r^2}, \]  

(57)

where

\[ \cos \alpha = \left(\frac{(x - y)^2 + \rho_s \rho_r - h^2}{2 \rho_s \rho_r}\right)^{1/2}. \]  

(58)

**Post-Stack Time Migration**

An interesting example of a stacking operator is the hyperbola summation used for time migration in the post-stack domain. In this case, the summation path is defined as

\[ \hat{\theta}(y; z, x) = \sqrt{z^2 + \frac{(x - y)^2}{v^2}}, \]  

(59)

where \(z\) denotes the vertical traveltime, \(x\) and \(y\) are the horizontal coordinates on the migrated and unmigrated sections respectively, and \(v\) stands for the effectively constant root-mean-square velocity \(\text{[Claerbout, 1995]}\). The summation path for the reverse transformation (demigration) is found from solving equation (59) for \(z\). It has the well-known elliptic form

\[ \theta(x; t, y) = \sqrt{t^2 - \frac{(x - y)^2}{v^2}}. \]  

(60)

The Jacobian of transforming \(z\) to \(t\) is

\[ \left|\frac{\partial \hat{\theta}}{\partial z}\right| = \frac{z}{t}. \]  

(61)

If the migration weighting function is defined by conventional downward continuation \(\text{[Schneider, 1978]}\), it takes the following form, which is equivalent to equation (40):

\[ \hat{w}(y; z, x) = \frac{1}{(2\pi v)^{m/2}} \cos \alpha(y) \]  

\[ \times \left(\frac{1}{2\pi v} R(y, x) = \frac{1}{(2\pi v)^{m/2}} \frac{\cos \alpha}{v^m \rho^{m/2}}. \]  

(62)

The simple trigonometry of the reflected ray suggests that the cosine factor in formula (62) is equal to the simple ratio between the vertical traveltime \(z\) and the zero-offset reflected travelttime \(t\):

\[ \cos \alpha = \frac{z}{t}. \]  

(63)

The equivalence of the Jacobian (61) and the cosine factor (13) has important interpretations in the theory of Stolt frequency-domain migration \(\text{[Stolt, 1978]}\). Chun and
According to equation (22), the weighting function of the adjoint operator is the ratio of (62) and (61):

$$\tilde{w}(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{1}{\nu^m t^{m/2}}.$$  

(64)

We can see that the cosine factor $z/t$ disappears from the adjoint weighting. This is completely analogous to the known effect of “dropping the Jacobian” in Stolt migration [Harlan 1983, Levin 1994]. The product of the weighting functions for the time migration and its asymptotic inverse is defined according to formula (10) as

$$w \hat{w} = \frac{1}{(2\pi)^m} \sqrt{\left| \bar{F} \right|^m \left( \frac{\partial \hat{\theta}}{\partial z} \right)^m} = \frac{1}{(\nu^2 t)^m}.$$  

(65)

Thus, the asymptotic inverse of the conventional time migration has the weighting function determined from equations (10) and (62) as

$$w(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{t/z}{\nu^m t^{m/2}}.$$  

(66)

The weighting functions of the asymptotic pseudo-unitary operators are obtained from formulas (28) and (29). They have the form

$$w^{(+)}(x; t, y) = \frac{1}{(2\pi)^{m/2}} \frac{\sqrt{t/z}}{\nu^m t^{m/2}}.$$  

(67)

$$w^{(-)}(y; z, x) = \frac{1}{(2\pi)^{m/2}} \frac{\sqrt{z/t}}{\nu^m t^{m/2}}.$$  

(68)

The square roots of the cosine factor appearing in formulas (67) and (68) correspond to the analogous terms in the pseudo-unitary Stolt migration proposed by [Harlan and Sword 1986].

Figure 1 shows the output of a simple numerical test. The synthetic zero-offset section used in this test is shown in the left plot of Figure 2. The data are taken from [Claerbout 1995] and correspond to a synthetic reflectivity model, which contains several dipping layers, a fault, and an unconformity. The input zero-offset section is inverted using an iterative conjugate-gradient method and two different weighting schemes: the uniform weighting and the asymptotic pseudo-unitary weighting (67-68). I compare the iterative convergence by measuring the least-squares norm of the data residual error at different iterations. Figure 1 shows that the pseudo-unitary weighting provides a significantly faster convergence. The result of inversion after 10 conjugate-gradient iterations is shown in Figures 2 and 3. The right plot in Figure 2 shows the output of the least-squares migration. Figure 3 shows the corresponding modeled data and the residual error. The latter is very close to zero. Although this example has only a pedagogical value, it clearly demonstrates possible advantages of using asymptotic pseudo-unitary operators in least-squares migration.
Figure 1: Comparison of convergence of the iterative least-squares migration. The dashed line corresponds to the unweighted (uniformly weighted) operator. The solid line corresponds to the asymptotic pseudo-unitary operator. The latter provides a noticeably faster convergence.

Figure 2: Input zero-offset section (left) and the corresponding least-squares image (right) after 10 iterations of iterative inversion.
Figure 3: The modeled zero-offset (left) and the residual error (right) plotted at the same scale.

Velocity Transform

Velocity transform is another form of hyperbolic stacking with the summation path

$$\hat{\theta}(h; t_0, s) = \sqrt{t_0^2 + s^2 h^2},$$

where $h$ corresponds to the offset, $s$ is the stacking slowness, and $t_0$ is the estimated zero-offset traveltime. Hyperbolic stacking is routinely applied for scanning velocity analysis in common-midpoint stacking. Velocity transform inversion has proved to be a powerful tool for data interpolation and amplitude-preserving multiple suppression (Thorson, 1984; Ji, 1995; Lumley et al., 1995).

Solving equation (69) for $t_0$, we find that the asymptotic inverse and adjoint operators have the elliptic summation path

$$\theta(s; t, h) = \sqrt{t^2 - s^2 h^2}.$$  

The weighting functions of the asymptotic pseudo-unitary velocity transform are found using formulas (28) and (29) to have the form

$$w^{(+)} = \frac{1}{(2 \pi)^{1/2}} \left| F \hat{F} \right|^{1/4} \frac{\partial \hat{\theta}}{\partial t_0}^{-1/4} = \frac{1}{\sqrt{\pi}} \frac{\sqrt{s h} \sqrt{t/t_0}}{\sqrt{t}} \cdot$$

$$w^{(-)} = \frac{1}{(2 \pi)^{1/2}} \left| F \hat{F} \right|^{1/4} \frac{\partial \hat{\theta}}{\partial t_0}^{3/4} = \frac{1}{\sqrt{\pi}} \frac{\sqrt{s h} \sqrt{t_0/t}}{\sqrt{t}} \cdot$$
The factor $\sqrt{s h}$ for pseudo-unitary velocity transform weighting has been discovered empirically by Claerbout (1995).

Figure 4 shows the output of a numerical test of the least-squares velocity transform inversion using a CMP gather from the Mobil AVO dataset (Lumley et al., 1995). The input CMP gather (shown in the left plot of Figure 5) is inverted using an iterative conjugate-gradient method and two different weighting schemes: the uniform weighting and the asymptotic pseudo-unitary weights (71-72). Analogously to Figure 1, the iterative convergence is measured by the least-squares norm of the data residual error at different iterations. Figure 4 shows that the pseudo-unitary weighting provides a noticeably faster convergence at the first three iterations. At later iterations, the residual errors of the two methods are very close to each other. The use of a pseudo-unitary weighting will be justified in this case if only three iterations are practically affordable. The results of inversion after 10 conjugate-gradient iterations are plotted in Figures 5 and 6. The right plot in Figure 5 shows the output of the velocity transform inversion: an optimized velocity scan. Figure 6 shows the corresponding modeled CMP gather and the residual error. The error is negligible which indicates a successful inversion.

Offset Continuation and DMO

Offset continuation is the operator that transforms seismic reflection data from one offset to another (Bolondi et al., 1982; Salvador and Savelli, 1982). If the data are continued from half-offset $h_1$ to a larger offset $h_2$, the summation path of the post-NMO integral offset continuation has the following form (Biondi and Chemingui, 1994; Stovas and Fomel, 1996; Fomel, 2001b):

$$
\theta(x; t, y) = \frac{t}{h_2} \sqrt{\frac{U + V}{2}},
$$

(73)
Figure 5: Input CMP gather (left) and its velocity transform counterpart (right) after 10 iterations of iterative least-squares inversion.

Figure 6: The modeled CMP gather (left) and the residual error (right) plotted at the same scale.
where $U = h_1^2 + h_2^2 - (x - y)^2$, $V = \sqrt{U^2 - 4 h_1^2 h_2^2}$, and $x$ and $y$ are the midpoint coordinates before and after the continuation. The summation path of the reverse continuation is found from inverting (73) to be

$$\hat{\theta}(y; z, x) = z h_2 \sqrt{\frac{2}{U + V}} = \frac{z}{h_1} \sqrt{\frac{U - V}{2}}. \quad (74)$$

The Jacobian of the time coordinate transformation in this case is simply

$$\left| \frac{\partial \hat{\theta}}{\partial z} \right| = \frac{t}{z}. \quad (75)$$

Differentiating summation paths (73) and (74), we can define the product of the weighting functions according to formula (10), as follows:

$$w \hat{w} = \frac{1}{2 \pi} \left| F \hat{F} \right|^{1/4} \left| \frac{\partial \hat{\theta}}{\partial t} \right|^{1/4} = \frac{t}{2 \pi} \left( \frac{h_2^2 - h_1^2}{V^3} \right)^{1/2} - (x - y)^4. \quad (76)$$

The weighting functions of the amplitude-preserving offset continuation have the form (Fomel, 2001b)

$$w(x; t, y) = \sqrt{\frac{z}{2 \pi}} \frac{h_2^2 - h_1^2 - (x - y)^2}{V^{3/2}}, \quad (77)$$

$$\hat{w}(y; z, x) = \frac{t}{\sqrt{2 \pi}} \frac{h_2^2 - h_1^2 + (x - y)^2}{V^{3/2}}. \quad (78)$$

It easy to verify that they satisfy relationship (76); therefore, they appear to be asymptotically inverse to each other.

The weighting functions of the asymptotic pseudo-unitary offset continuation are defined from formulas (28) and (29), as follows:

$$w(+) = \frac{1}{(2 \pi)^{1/2}} \left| F \hat{F} \right|^{1/4} \left| \frac{\partial \hat{\theta}}{\partial t} \right|^{-1/4} = \sqrt{\frac{z}{2 \pi}} \left( \frac{(h_2^2 - h_1^2)^2 - (x - y)^4}{V^{3/2}} \right)^{1/2}, \quad (79)$$

$$w(-) = \frac{1}{(2 \pi)^{1/2}} \left| F \hat{F} \right|^{1/4} \left| \frac{\partial \hat{\theta}}{\partial t} \right|^{3/4} = \frac{t}{\sqrt{2 \pi}} \left( \frac{(h_2^2 - h_1^2)^2 - (x - y)^4}{V^{3/2}} \right)^{1/2}. \quad (80)$$

The most important case of offset continuation is the continuation to zero offset. This type of continuation is known as dip moveout (DMO). Setting the initial offset $h_1$ equal to zero in the general offset continuation formulas, we deduce that the inverse and forward DMO operators have the summation paths

$$\theta(x; t, y) = t \frac{\sqrt{h_2^2 - (x - y)^2}}, \quad (81)$$

$$\hat{\theta}(y; z, x) = \frac{z h_2}{\sqrt{h_2^2 - (x - y)^2}}. \quad (82)$$
The weighting functions of the amplitude-preserving inverse and forward DMO are

\[
w(x; t, y) = \sqrt{\frac{z}{2\pi}} \frac{1}{h_2}, \tag{83}
\]

\[
\tilde{w}(y; z, x) = \frac{t/\sqrt{z}}{\sqrt{2\pi}} \frac{h_2 (h_2^2 + (x - y)^2)}{(h_2^2 - (x - y)^2)^2}, \tag{84}
\]

and the weighting functions of the asymptotic pseudo-unitary DMO are

\[
w^{(+)} = \sqrt{\frac{z}{2\pi}} \frac{\sqrt{h_2^2 + (x - y)^2}}{h_2^2 - (x - y)^2}, \tag{85}
\]

\[
w^{(-)} = \frac{t/\sqrt{z}}{\sqrt{2\pi}} \frac{\sqrt{h_2^2 + (x - y)^2}}{h_2^2 - (x - y)^2}. \tag{86}
\]

Equations similar to (83) and (84) have been published by Stovas and Fomel (1996). Equation (84) differs from the similar result of Black et al. (1993) by a simple time multiplication factor. This difference corresponds to the difference in definition of the amplitude preservation criterion. Equation (84) agrees asymptotically with the frequency-domain Born DMO operators (Bleistein 1990; Liner 1991; Bleistein and Cohen 1995). Likewise, the stacking operator with the weighting function (83) corresponds to Ronen’s inverse DMO (Ronen 1987), as discussed by Fomel (2001b). Its adjoint, which has the weighting function

\[
\tilde{w}(x; t, y) = \frac{t/\sqrt{z}}{2\pi} \frac{1}{h_2}, \tag{87}
\]

corresponds to Hale’s DMO (Hale 1984).

**CONCLUSIONS**

Stacking operators such as Kirchoff migration, datuming, dip moveout, velocity transform, etc. are widely used in seismic imaging and data processing, and the need often arises to invert them.

This paper fills the gap between the concept of asymptotically inverse operators and the concept of adjoint operators by introducing the notion of asymptotic pseudo-unitary stacking operators. A pair of asymptotic pseudo-unitary operators possesses the property of being both adjoint and asymptotically inverse to each other. The amplitude (weighting) functions of these operators are completely defined by the derivatives of their kinematics (stacking surfaces).

The practical advantage of this unification is in the ability to construct asymptotically optimal preconditioning for iterative least-squares solution of inverse problems. Simple preliminary tests are encouraging, but further practical experience is needed to confirm the theoretical expectations.
ACKNOWLEDGMENTS

I owe my familiarity with the asymptotic inversion theory to Sergey Goldin. A short discussion with Martin Tygel helped me better understand the true-amplitude migration concept.

I thank Jon Claerbout for helpful discussions and the sponsors of the Stanford Exploration Project for the financial support of this work. Comments from three anonymous reviewers helped to improve the paper.

APPENDIX A

LEAST-SQUARES RADON TRANSFORM INVERSION

This appendix exemplifies the application of adjoint operators by reviewing the analytical least-squares inversion of the classic Radon transform (slant stack operator).

Forming the product $A^T A$ for this case leads to the double integral

$$H(z, x) = (A^T A)[M(z, x)] = \iint \tilde{w}(y; z, x) w(\xi; \tilde{\theta}(y; z, x), y) M(\theta(\xi; \tilde{\theta}(y; z, x), y), \xi) \, d\xi \, dy = \iint M(z + y (\xi - x)) \, d\xi \, dy.$$  \hspace{1cm} (A-1)

Applying Fourier transform with respect to $z$, we can rewrite equation \eqref{A-1} in the frequency domain as

$$\hat{H}(\omega, x) = \int \hat{M}(\omega, \xi) \int e^{i\omega y (\xi - x)} \, dy \, d\xi,$$  \hspace{1cm} (A-2)

where

$$\hat{H}(\omega, x) = \int H(z, x) e^{-i\omega z} \, dz,$$  \hspace{1cm} (A-3)

$$\hat{M}(\omega, x) = \int M(z, x) e^{-i\omega z} \, dz.$$  \hspace{1cm} (A-4)

The inner integral in equation \eqref{A-2} reduces to the $m$-dimensional delta function:

$$\hat{H}(\omega, x) = (2 \pi)^m \int \hat{M}(\omega, \xi) \delta(\omega^m (\xi - x)) \, d\xi.$$  \hspace{1cm} (A-5)

As follows from the properties of delta function,

$$\hat{H}(\omega, x) = \frac{(2 \pi)^m}{|\omega|^m} \int \hat{M}(\omega, \xi) \delta(\xi - x) \, d\xi = \frac{(2 \pi)^m}{|\omega|^m} \hat{M}(\omega, x).$$  \hspace{1cm} (A-6)
Inverting $A^{-6}$ for $M$, we conclude that

$$(A^T A)^{-1} = \frac{|D|^m}{(2\pi)^m}. \tag{A-7}$$

Substituting equation $A^{-7}$ into (17) produces the result precisely equivalent to Radon’s inversion $[4]$.

**REFERENCES**

Fomel, S., J. Berryman, R. Clapp, and M. Prucha, 2002, Iterative resolution estimation in least-squares Kirchhoff migration: Geophysical Prospecting, accepted for publication.
——–, 1988, Transformation and recovery of discontinuities in problems of tomographic type: Institute of Geology and Geophysics.
Jin, S., R. Madariaga, J. Virieux, and G. Lambaré, 1992, Two-dimensional asymptotic


ABSTRACT

As I will illustrate in later chapters, the crucial part of data regularization problems is in the choice and implementation of the regularization operator $D$ or the corresponding preconditioning operator $P$. The choice of the forward modeling operator $L$ is less critical. In this chapter, I discuss the nature of forward interpolation, which has been one of the traditional subjects in computational mathematics. Wolberg (1990) presents a detailed review of different conventional approaches. I discuss a simple mathematical theory of interpolation from a regular grid and derive the main formulas from a very general idea of function bases. Forward interpolation plays only a supplementary role in this dissertation, but it has many primary applications, such as trace resampling, NMO, Kirchhoff and Stolt migrations, log-stretch, and radial transform, in seismic data processing and imaging. Two simple examples appear at the end of this chapter.

INTERPOLATION THEORY

Mathematical interpolation theory considers a function $f$, defined on a regular grid $N$. The problem is to find $f$ in a continuum that includes $N$. I am not defining the dimensionality of $N$ and $f$ here because it is not essential for the derivations. Furthermore, I am not specifying the exact meaning of “regular grid,” since it will become clear from the analysis that follows. The function $f$ is assumed to belong to a Hilbert space with a defined dot product.

If we restrict our consideration to a linear case, the desired solution will take the following general form

$$f(x) = \sum_{n \in N} W(x, n) f(n),$$

where $x$ is a point from the continuum, and $W(x, n)$ is a linear weight function that can take both positive and negative values. If the grid $N$ itself is considered as continuous, the sum in formula (1) transforms to an integral in $dn$. Two general properties of the linear weighting function $W(x, n)$ are evident from formula (1).

Property 1

$$W(n, n) = 1.$$  \hspace{1cm} (2)
Equality (2) is necessary to assure that the interpolation of a single spike at some point \( n \) does not change the value \( f(n) \) at the spike.

**Property 2**

\[
\sum_{n \in N} W(x, n) = 1 .
\]

This property is the normalization condition. Formula (3) assures that interpolation of a constant function \( f(n) \) remains constant.

One classic example of the interpolation weight \( W(x, n) \) is the Lagrange polynomial, which has the form

\[
W(x, n) = \frac{(x - i)}{(n - i)} .
\]

The Lagrange interpolation provides a unique polynomial, which goes exactly through the data points \( f(n) \). The local 1-point Lagrange interpolation is equivalent to the nearest-neighbor interpolation, defined by the formula

\[
W(x, n) = \begin{cases} 
1, & \text{for } n - 1/2 \leq x < n + 1/2 \\
0, & \text{otherwise}
\end{cases}
\]

Likewise, the local 2-point Lagrange interpolation is equivalent to the linear interpolation, defined by the formula

\[
W(x, n) = \begin{cases} 
1 - |x - n|, & \text{for } n - 1 \leq x < n + 1 \\
0, & \text{otherwise}
\end{cases}
\]

Because of their simplicity, the nearest-neighbor and linear interpolation methods are very practical and easy to apply. Their accuracy is, however, limited and may be inadequate for interpolating high-frequency signals. The shapes of interpolants (5) and (6) and their spectra are plotted in Figures 1 and 2. The spectral plots show that both interpolants act as low-pass filters, preventing the high-frequency energy from being correctly interpolated.

The Lagrange interpolants of higher order correspond to more complicated polynomials. Another popular practical approach is cubic convolution (Keys, 1981). The cubic convolution interpolant is a local piece-wise cubic function:

\[
W(x, n) = \begin{cases} 
3/2|x - n|^3 - 5/2|x - n|^2 + 1, & \text{for } 0 \leq |x - n| < 1 \\
-1/2|x - n|^3 + 5/2|x - n|^2 - 4|x - n| + 2, & \text{for } 1 \leq |x - n| < 2 \\
0, & \text{otherwise}
\end{cases}
\]

The shapes of interpolant (7) and its spectrum are plotted in Figure 3.

\footnote{It is interesting to note that the interpolation and finite-difference filters developed by Karrenbach (1995) from a general approach of self-similar operators reduce to a localized form of Lagrange polynomials.}
Figure 1: Nearest-neighbor interpolant (left) and its spectrum (right).

Figure 2: Linear interpolant (left) and its spectrum (right).

Figure 3: Cubic-convolution interpolant (left) and its spectrum (right).
I compare the accuracy of different forward interpolation methods on a one-dimensional signal shown in Figure 4. The ideal signal has an exponential amplitude decay and a quadratic frequency increase from the center towards the edges. It is sampled at a regular 50-point grid and interpolated to 500 regularly sampled locations. The interpolation result is compared with the ideal one. Figures 5 and 6 show the interpolation error steadily decreasing as we proceed from 1-point nearest-neighbor to 2-point linear and 4-point cubic-convolution interpolation. At the same time, the cost of interpolation grows proportionally to the interpolant length.

Figure 4: One-dimensional test signal. Top: ideal. Bottom: sampled at 50 regularly spaced points. The bottom plot is the input in a forward interpolation test.

Figure 5: Interpolation error of the nearest-neighbor interpolant (dashed line) compared to that of the linear interpolant (solid line).

FUNCTION BASIS

A particular form of the solution (1) arises from assuming the existence of a basis function set \( \{ \psi_k(x) \} \), \( k \in K \), such that the function \( f(x) \) can be represented by a linear combination of the basis functions in the set, as follows:

\[
 f(x) = \sum_{k \in K} c_k \psi_k(x). 
\] (8)
Figure 6: Interpolation error of the linear interpolant (dashed line) compared to that of the cubic convolution interpolant (solid line).

We can find the linear coefficients $c_k$ by multiplying both sides of equation (8) by one of the basis functions (e.g. $\psi_j(x)$). Inverting the equality

$$\langle \psi_j(x), f(x) \rangle = \sum_{k \in K} c_k \Psi_{jk}, \quad (9)$$

where the parentheses denote the dot product, and

$$\Psi_{jk} = \langle \psi_j(x), \psi_k(x) \rangle, \quad (10)$$

leads to the following explicit expression for the coefficients $c_k$:

$$c_k = \sum_{j \in K} \Psi^{-1}_{kj} (\psi_j(x), f(x)). \quad (11)$$

Here $\Psi^{-1}_{kj}$ refers to the $kj$ component of the matrix, which is the inverse of $\Psi$. The matrix $\Psi$ is invertible as long as the basis set of functions is linearly independent. In the special case of an orthonormal basis, $\Psi$ reduces to the identity matrix:

$$\Psi_{jk} = \Psi^{-1}_{kj} = \delta_{jk}. \quad (12)$$

Equation (11) is a least-squares estimate of the coefficients $c_k$: one can alternatively derive it by minimizing the least-squares norm of the difference between $f(x)$ and the linear decomposition (8). For a given set of basis functions, equation (11) approximates the function $f(x)$ in formula (1) in the least-squares sense.

SOLUTION

The usual (although not unique) mathematical definition of the continuous dot product is

$$\langle f_1, f_2 \rangle = \int \bar{f}_1(x) f_2(x) dx, \quad (13)$$
where the bar over $f_1$ stands for complex conjugate (in the case of complex-valued functions). Applying definition (13) to the dot product in equation (11) and approximating the integral by a finite sum on the regular grid $N$, we arrive at the approximate equality

$$
(\psi_j(x), f(x)) = \int \bar{\psi}_j(x)f(x)dx \approx \sum_{n \in N} \bar{\psi}_j(n)f(n).
$$

We can consider equation (14) not only as a useful approximation, but also as an implicit definition of the regular grid. Grid regularity means that approximation (14) is possible. According to this definition, the more regular the grid is, the more accurate is the approximation.

Substituting equality (14) into equations (11) and (8) yields a solution to the interpolation problem. The solution takes the form of equation (11) with

$$
W(x, n) = \sum_{k \in K} \sum_{j \in K} \Psi_{kj}^{-1} \psi_k(x) \bar{\psi}_j(n).
$$

We have found a constructive way of creating the linear interpolation operator from a specified set of basis functions.

It is important to note that the adjoint of the linear operator in formula (1) is the continuous dot product of the functions $W(x, n)$ and $f(x)$. This simple observation follows from the definition of the adjoint operator and the simple equality

$$
\left( f_1(x), \sum_{n \in N} W(x, n)f_2(n) \right) = \sum_{n \in N} f_2(n) (f_1(x), W(x, n)) = \left( (W(x, n), f_1(x)) , f_2(n) \right).
$$

In the final equality, we have assumed that the discrete dot product is defined by the sum

$$
(f_1(n), f_2(n)) = \sum_{n \in N} \bar{f}_1(n)f_2(n).
$$

Applying the adjoint interpolation operator to the function $f$, defined with the help of formula (15), and employing formulas (8) and (11), we discover that

$$
(W(x, n), f(x)) = \sum_{k \in K} \sum_{j \in K} \Psi_{kj}^{-1} \psi_j(n) (\psi_k(x), f(x)) = \sum_{j \in K} \bar{\psi}_j(n) \sum_{k \in K} \Psi_{jk}^{-1} (\psi_k(x), f(x)) = \sum_{j \in K} c_j \bar{\psi}_j(n) = f(n).
$$

This remarkable result shows that although the forward linear interpolation is based on approximation (14), the adjoint interpolation produces an exact value of $f(n)$! The approximate nature of equation (15) reflects the fundamental difference between adjoint and inverse linear operators (Claerbout, 1992).

When adjoint interpolation is applied to a constant function $f(x) \equiv 1$, it is natural to require the constant output $f(n) = 1$. This requirement leads to yet another general property of the interpolation functions $W(x, n)$:
Property 3

\[ \int W(x, n) dx = 1 . \] (19)

The functional basis approach to interpolation is well developed in the sampling theory (Garcia, 2000). Some classic examples are discussed in the next section.

### INTERPOLATION WITH FOURIER BASIS

To illustrate the general theory with familiar examples, I consider in this section the most famous example of an orthonormal function basis, the Fourier basis of trigonometric functions. What kind of linear interpolation does this basis lead to?

#### Continuous Fourier basis

For the continuous Fourier transform, the set of basis functions is defined by

\[ \psi_{\omega}(x) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} , \] (20)

where \( \omega \) is the continuous frequency. For a 1-point sampling interval, the frequency is limited by the Nyquist condition: \(|\omega| \leq \pi\). In this case, the interpolation function \( W \) can be computed from equation (15) to be

\[ W(x, n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega(x-n)} d\omega = \frac{\sin[\pi(x-n)]}{\pi(x-n)} . \] (21)

The shape of the interpolation function [21] and its spectrum are shown in Figure 3. The spectrum is identically equal to 1 in the Nyquist frequency band.

![Figure 7: Sinc interpolant (left) and its spectrum (right).](image-url)
Function (21) is well-known as the Shannon sinc interpolant. According to the sampling theorem (Kotel’nikov, 1933; Shannon, 1949), it provides an optimal interpolation for band-limited signals. A known problem prohibiting its practical implementation is the slow decay with \((x - n)\), which results in a far too expensive computation. This problem is solved in practice with heuristic tapering (Hale, 1980), such as triangle tapering (Harlan, 1982), or more sophisticated taper windows (Wolberg, 1990). One popular choice is the Kaiser window (Kaiser and Shafer, 1980), which has the form

\[
W(x, n) = \begin{cases} 
\frac{\sin [\pi (x - n)]}{\pi (x - n)} \frac{I_0 \left( a \sqrt{1 - \left( \frac{x-n}{N} \right)^2} \right)}{I_0(a)} & \text{for } n - N < x < n + N \\
0, & \text{otherwise} 
\end{cases}
\]  

(22)

where \(I_0\) is the zero-order modified Bessel function of the first kind. The Kaiser-windowed sinc interpolant (5) has the adjustable parameter \(a\), which controls the behavior of its spectrum. I have found empirically the value of \(a = 4\) to provide a spectrum that deviates from 1 by no more than 1% in a relatively wide band.

While the function \(W\) from equation (21) automatically satisfies properties (3) and (19), where both \(x\) and \(n\) range from \(-\infty\) to \(\infty\), its tapered version may require additional normalization.

Figure 7 compares the interpolation error of the 8-point Kaiser-tapered sinc interpolant with that of cubic convolution on the example from Figure 4. The accuracy improvement is clearly visible.

The differences among the described forward interpolation methods are also clearly visible from the discrete spectra of the corresponding interpolants. The left plots in Figures 8 and 9 show discrete interpolation responses: the function \(W(x, n)\) for a fixed value of \(x = 0.7\). The right plots compare the corresponding discrete spectra. Clearly, the spectrum gets flatter and wider as the accuracy of the method increases.
Figure 9: Discrete interpolation responses of linear and cubic convolution interpolants (left) and their discrete spectra (right) for $x = 0.7$. [forwd/chirp/ speclincub]

Figure 10: Discrete interpolation responses of cubic convolution and 8-point windowed sinc interpolants (left) and their discrete spectra (right) for $x = 0.7$. [forwd/chirp/ speccubkai]
Discrete Fourier basis

Assuming that the range of the variable $x$ is limited in the interval from $-N$ to $N$, the discrete Fourier basis (Fast Fourier Transform) employs a set of orthonormal periodic functions

$$\psi_k(x) = \frac{1}{\sqrt{2N}} e^{i\pi \frac{k}{2N} x}, \quad (23)$$

where the discrete frequency index $k$ also ranges, according to the Nyquist sampling criterion, from $-N$ to $N$. The interpolation function is computed from equation (15) to be

$$W(x, n) = \frac{1}{2N} \sum_{k=-N}^{N-1} e^{i\pi \frac{k}{2N} (x-n)} = \frac{1}{2N} e^{-i\pi (x-n)} \left[ 1 + e^{i\pi \frac{x-n}{N}} + \cdots + e^{i\pi \frac{2N-1}{N} (x-n)} \right] =$$

$$\frac{1}{2N} e^{-i\pi (x-n)} \frac{e^{i\pi \frac{x-n}{2N}} - 1}{e^{i\pi \frac{x-n}{N}} - e^{-i\pi \frac{x-n}{2N}}} = e^{-i\pi \frac{x-n}{2N}} \frac{\sin \left[ \pi (x-n) \right]}{2N \sin \left[ \pi (x-n)/2N \right]} \cdot \quad (24)$$

An interpolation function equivalent to (24) has been found by Muir (Lin et al., 1993; Popovici et al., 1993, 1996). It can be considered a tapered version of the sinc interpolant (21) with smooth tapering function

$$\frac{\pi (x-n)/2N}{\tan \left[ \pi (x-n)/2N \right]} \cdot$$

Unlike most other tapered-sinc interpolants, Muir’s interpolant (24) satisfies not only the obvious property (2), but also properties (3) and (19), where the interpolation function $W(x, n)$ should be set to zero for $x$ outside the range from $n - N$ to $n + N$. The form of this function is shown in Figure 11.

The development of the mathematical wavelet theory (Daubechies, 1992) has opened the door to a whole universe of orthonormal function bases, different from the Fourier basis. The wavelet theory should find many useful applications in geophysical data interpolation, but exploring this interesting opportunity would go beyond the scope of the present work.

The next section carries the analysis to the continuum and compares the mathematical interpolation theory with the theory of seismic imaging.

CONTINUOUS CASE AND SEISMIC IMAGING

Of course, the linear theory is not limited to discrete grids. It is interesting to consider the continuous case because of its connection to the linear integral operators commonly used in seismic imaging. Indeed, in the continuous case, linear decomposition
Figure 11: The left plots show the sinc interpolation function. Note the slow decay in $x$. The middle shows the effective tapering function of Muir’s interpolation; the right is Muir’s interpolant. The top is for $N = 2$ (5-point interpolation); the bottom, $N = 6$ (13-point interpolation).
(8) takes the form of the integral operator

\[ f(y) = \int m(x)G(y; x)dx \]  

(25)

where \( x \) is a continuous analog of the discrete coefficient \( k \) in (8), the continuous function \( m(x) \) is analogous to the coefficient \( c_k \), and \( G(y; x) \) is analogous to one of the basis functions \( \psi_k(x) \). The linear integral operator in (25) has a mathematical form similar to the form of well-known integral imaging operators, such as Kirchhoff migration or “Kirchhoff” DMO. Function \( G(y; x) \) in this case represents the Green’s function (impulse response) of the imaging operator. Linear decomposition of the data into basis functions means decomposing it into the combination of impulse responses (“hyperbolas”).

In the continuous case, equation (15) transforms to

\[ W(y, n) = \int \int \Psi^{-1}(x_1, x_2)G(y; x_1)\bar{G}(n; x_2)dx_1 dx_2 \]  

(26)

where \( \Psi^{-1}(x_1, x_2) \) refers to the inverse of the “matrix” operator

\[ \Psi(x_1, x_2) = \int G(y; x_1)\bar{G}(y; x_2)dy \]  

(27)

When the linear operator, defined by equation (25), is unitary,

\[ \Psi^{-1}(x_1, x_2) = \delta(x_1 - x_2) \]  

(28)

and equation (26) simplifies to the single integral

\[ W(y, n) = \int G(y; x)\bar{G}(n; x)dx \]  

(29)

With respect to seismic imaging operators, one can recognize in the interpolation operator (29) the generic form of azimuth moveout (Biondi et al., 1996), which is derived either as a cascade of adjoint (\( \bar{G}(n; y) \)) and forward (\( G(x; y) \)) DMO or as a cascade of migration (\( \bar{G}(n; y) \)) and modeling (\( G(x; y) \)) (Fomel and Biondi, 1995; Biondi et al., 1998). In the first case, the intermediate variable \( y \) corresponds to the space of zero-offset data cube. In the second case, it corresponds to a point in the subsurface.

**Asymptotically pseudo-unitary operators as orthonormal bases**

It is interesting to note that many integral operators routinely used in seismic data processing have the form of operator (25) with the Green’s function

\[ G(t, y; z, x) = \left| \frac{\partial}{\partial t} \right|^{m/2} A(x; t, y)\delta(z - \theta(x; t, y)) \]  

(30)
where we have split the variable $x$ into the one-dimensional component $z$ (typically depth or time) and the $m$-dimensional component $x$ (typically a lateral coordinate with $m$ equal 1 or 2). Similarly, the variable $y$ is split into $t$ and $y$. The function $\theta$ represents the summation path, which captures the kinematic properties of the operator, and $A$ is the amplitude function. In the case of $m = 1$, the fractional derivative $\left| \frac{\partial}{\partial t} \right|^{m/2}$ is defined as the operator with the frequency response $(i\omega)^{m/2}$, where $\omega$ is the temporal frequency ([Samko et al., 1993]).

The impulse response (30) is typical for different forms of Kirchhoff migration and datuming as well as for velocity transform, integral offset continuation, DMO, and AMO. Integral operators of that class rarely satisfy the unitarity condition, with the Radon transform (slant stack) being a notable exception. In an earlier paper [Fomel, 1996], I have shown that it is possible to define the amplitude function $A$ for each kinematic path $\theta$ so that the operator becomes asymptotically pseudo-unitary. This means that the adjoint operator coincides with the inverse in the high-frequency (stationary-phase) approximation. Consequently, equation (28) is satisfied to the same asymptotic order.

Using asymptotically pseudo-unitary operators, we can apply formula (29) to find an explicit analytic form of the interpolation function $W$, as follows:

$$W(t, y; t_n, y_n) = \int \int \int G(t, y; z, x) G(t_n, y_n; z, x) \delta \left( \theta(x; t, y) - \theta(x; t_n, y_n) \right) dxdz.$$  

(31)

Here the amplitude function $A$ is defined according to the general theory of asymptotically pseudo-inverse operators as

$$A = \frac{1}{(2\pi)^{m/2}} \left| \hat{F} \right|^{1/4} \left| \frac{\partial \theta}{\partial t} \right|^{(m+2)/4},$$  

(32)

where

$$F = \frac{\partial \theta}{\partial t} \frac{\partial^2 \theta}{\partial x \partial y} - \frac{\partial \theta}{\partial y} \frac{\partial^2 \theta}{\partial x \partial t},$$  

(33)

$$\hat{F} = \frac{\partial \tilde{\theta}}{\partial z} \frac{\partial^2 \tilde{\theta}}{\partial x \partial y} - \frac{\partial \tilde{\theta}}{\partial y} \frac{\partial^2 \tilde{\theta}}{\partial x \partial z},$$  

(34)

and $\tilde{\theta}(x; t, y)$ is the dual summation path, obtained by solving equation $z = \theta(x; t, y)$ for $t$ (assuming that an explicit solution is possible).

For a simple example, let us consider the case of zero-offset time migration with a constant velocity $v$. The summation path $\theta$ in this case is an ellipse

$$\theta(x; t, y) = \sqrt{t^2 - \frac{(x - y)^2}{v^2}}.$$  

(35)
and the dual summation path \( \hat{\theta} \) is a hyperbola

\[
\hat{\theta}(y; z, x) = \sqrt{z^2 + \frac{(x - y)^2}{v^2}}.
\]  

(36)

The corresponding pseudo-unitary amplitude function is found from formula (32) to be (Fomel, 1996)

\[
A = \frac{1}{(2\pi)^{m/2}} \sqrt{\frac{t}{z}} \frac{1}{\sqrt{z^m z^{m/2}}}. 
\]  

(37)

Substituting formula (37) into (31), we derive the corresponding interpolation function

\[
W(t, y; t_n, y_n) = \frac{1}{(2\pi)^m} \left| \frac{\partial}{\partial t} \right|^{m/2} \left| \frac{\partial}{\partial t_n} \right|^{m/2} \int \frac{\sqrt{tt_n}}{v^2 z^m z^{m+1}} \delta(z - z_n) \, dx, 
\]  

(38)

where \( z = \theta(x; t, y) \), and \( z_n = \theta(x; t_n, y_n) \). For \( m = 1 \) (the two-dimensional case), we can apply the known properties of the delta function to simplify formula (38) further to the form

\[
W = \frac{\sqrt{tt_n}}{\pi} \left| \frac{\partial}{\partial t} \right|^{1/2} \left| \frac{\partial}{\partial t_n} \right|^{1/2} \frac{\sqrt{tt_n}}{\sqrt{(y - y_n)^2 - v^2(t - t_n)^2} \left[ v^2(t + t_n)^2 - (y - y_n)^2 \right]}.
\]  

(39)

The result is an interpolant for zero-offset seismic sections. Like the sinc interpolant in equation (21), which is based on decomposing the signal into sinusoids, equation (39) is based on decomposing the zero-offset section into hyperbolas.

While opening a curious theoretical possibility, seismic imaging interpolants have an undesirable computational complexity. Following the general regularization framework of Chapter ??, I shift the computational emphasis towards appropriately chosen regularization operators discussed in Chapter ?? For the forward interpolation method, all data examples in this dissertation use either the simplest nearest neighbor and linear interpolation or a more accurate B-spline method, described in the next section.

**INTERPOLATION WITH CONVOLUTIONAL BASES**

Unser et al. (1993) noticed that the basis function idea has an especially simple implementation if the basis is convolutional and satisfies the equation

\[
\psi_k(x) = \beta(x - k).
\]  

(40)

In other words, the basis is constructed by integer shifts of a single function \( \beta(x) \). Substituting expression (7) into equation (8) yields

\[
f(x) = \sum_{k \in K} c_k \beta(x - k).
\]  

(41)
Evaluating the function $f(x)$ in equation (15) at an integer value $n$, we obtain the equation

$$f(n) = \sum_{k \in K} c_k \beta(n - k),$$

which has the exact form of a discrete convolution. The basis function $\beta(x)$, evaluated at integer values, is digitally convolved with the vector of basis coefficients to produce the sampled values of the function $f(x)$. We can invert equation (9) to obtain the coefficients $c_k$ from $f(n)$ by inverse recursive filtering (deconvolution). In the case of a non-causal filter $\beta(n)$, an appropriate spectral factorization will be needed prior to applying the recursive filtering.

According to the convolutional basis idea, forward interpolation becomes a two-step procedure. The first step is the direct inversion of equation (9): the basis coefficients $c_k$ are found by deconvolving the sampled function $f(n)$ with the factorized filter $\beta(n)$. The second step reconstructs the continuous (or arbitrarily sampled) function $f(x)$ according to formula (15). The two steps could be combined into one, but usually it is more convenient to apply them separately. I show a schematic relationship among different variables in Figure 1.

**Figure 12:** Schematic relationship among different variables for interpolation with a convolutional basis.

---

**B-splines**

B-splines represent a particular example of a convolutional basis. Because of their compact support and other attractive numerical properties, B-splines are a good choice of the basis set for the forward interpolation problem and related signal processing problems (Unser, 1999). According to Thévenaz et al. (2000), they exhibit superior performance for any given order of accuracy in comparison with other methods of similar efficiency.

B-splines of the order 0 and 1 coincide with the nearest neighbor and linear interpolants (5) and (6) respectively. B-splines $\beta^n(x)$ of a higher order $n$ can be defined by a repetitive convolution of the zeroth-order spline $\beta^0(x)$ (the box function) with itself:

$$\beta^n(x) = \underbrace{\beta^0(x) \ast \cdots \ast \beta^0(x)}_{(n+1) \text{ times}}.$$
There is also the explicit expression

\[ \beta^n(x) = \frac{1}{n!} \sum_{k=0}^{n+1} C_k^{n+1} (-1)^k (x + \frac{n+1}{2} - k)_+^n, \] (44)

which can be proved by induction. Here \( C_k^{n+1} \) are the binomial coefficients, and the function \( x_+ \) is defined as follows:

\[ x_+ = \begin{cases} x, & \text{for } x > 0 \\ 0, & \text{otherwise} \end{cases} \] (45)

As follows from formula (11), the most commonly used cubic B-spline \( \beta^3(x) \) has the expression

\[ \beta^3(x) = \begin{cases} \frac{(4 - 6|x|^2 + 3|x|^3)}{6}, & \text{for } 1 > |x| \\ \frac{(2 - |x|)^3}{6}, & \text{for } 2 > |x| \\ 0, & \text{elsewhere} \end{cases} \] (46)

The corresponding discrete filter \( \beta^3(n) \) is a centered 3-point filter with coefficients 1/6, 2/3, and 1/6. According to the traditional method, deconvolution with this filter is performed as a tridiagonal matrix inversion (de Boor 1978). One can, however, accomplish the same task more efficiently by spectral factorization and recursive filtering (Unser et al. 1993). The recursive filtering approach generalizes straightforwardly to B-splines of higher orders.

Both the support length and the smoothness of B-splines increase with the order. In the limit, B-splines converge to the Gaussian function. Figures 11 and 12 show the third- and seventh-order splines \( \beta^3(x) \) and \( \beta^7(x) \), respectively, and their continuous spectra.

![Figure 13: Third-order B-spline \( \beta^3(x) \) (left) and its spectrum (right).](image)

It is important to realize the difference between B-splines and the corresponding interpolants \( W(x, n) \), which are sometimes called cardinal splines. An explicit computation of the cardinal splines is impractical, because they have infinitely long support.
Typically, they are constructed implicitly by the two-step interpolation method outlined above. The cardinal splines of orders 3 and 7 and their spectra are shown in Figures 13 and 14. As B-splines converge to the Gaussian function, the corresponding interpolants rapidly converge to the sinc function (21). Good convergence is achieved with the help of the implicitly-generated long support, which results from recursive filtering at the first step of the interpolation procedure.

In practice, the recursive filtering step adds only marginally to the total interpolation cost. Therefore, an $n$-th order B-spline interpolation is comparable in cost with any other method that uses an $(n+1)$-point interpolant. The comparison in accuracy usually turns out in favor of B-splines. Figures 15 and 16 compare interpolation errors of B-splines and other similar-cost methods on the example from Figure 4.

Similarly to the comparison in Figures 8 and 9, we can also compare the discrete responses of B-spline interpolation with those of other methods. The right plots in Figures 17 and 18 show that the discrete spectra of the effective B-spline interpolants are genuinely flat at low frequencies and wider than those of the competitive methods. Although the B-spline responses are infinitely long because of the recursive filtering step, they exhibit a fast amplitude decay.
Figure 16: Effective seventh-order B-spline interpolant (left) and its spectrum (right).

Figure 17: Interpolation error of the cubic-convolution interpolant (dashed line) compared to that of the third-order B-spline (solid line). [forwd/chirp/cubspl4]

Figure 18: Interpolation error of the 8-point windowed sinc interpolant (dashed line) compared to that of the seventh-order B-spline (solid line). [forwd/chirp/kaispl8]
Figure 19: Discrete interpolation responses of cubic convolution and third-order B-spline interpolants (left) and their discrete spectra (right) for $x = 0.7$.

Figure 20: Discrete interpolation responses of 8-point windowed sinc and seventh-order B-spline interpolants (left) and their discrete spectra (right) for $x = 0.7$. 
2-D example

For completeness, I include a 2-D forward interpolation example. Figure [19] shows a 2-D analog of the function in Figure [4] and its coarsely-sampled version.

![Figure 21: Two-dimensional test function (left) and its coarsely sampled version (right).](forw/chirp2/chirp2)

Figure [20] compares the errors of the 2-D nearest neighbor and 2-D linear (bi-linear) interpolation. Switching to bi-linear interpolation shows a significant improvement, but the error level is still relatively high. As shown in Figures [21] and [22], B-spline interpolation again outperforms other methods with comparable cost. In all cases, I constructed 2-D interpolants by orthogonal splitting. Although the splitting method reduces computational overhead, the main cost factor is the total interpolant size, which is squared when the interpolation goes from one to two dimensions.

Beyond B-splines

It is not too difficult to construct a convolutional basis with more accurate interpolation properties than those of B-splines, for example by sacrificing the function smoothness. The following piece-wise cubic function has a lower smoothness than $\beta^3(x)$ in equation [13] but slightly better interpolation behavior:

$$
\mu^3(x) = \begin{cases} 
(10 - 13|x|^2 + 6|x|^3)/16, & \text{for } 1 > |x| \geq 0 \\
(2 - |x|)^2(5 - 2|x|)/16, & \text{for } 2 > |x| \geq 1 \\
0, & \text{elsewhere}
\end{cases}
$$

[47]

Blu et al. [1998] have developed a general approach for constructing non-smooth piece-wise functions with optimal interpolation properties. However, the gain in ac-
Figure 22: 2-D Interpolation errors of nearest neighbor interpolation (left) and linear interpolation (right). The top graphs show 1-D slices through the center of the image. Bi-linear interpolation exhibits smaller error and therefore is more accurate.

Figure 23: 2-D Interpolation errors of cubic convolution interpolation (left) and third-order B-spline interpolation (right). The top graphs show 1-D slices through the center of the image. B-spline interpolation exhibits smaller error and therefore is more accurate.
Figure 24: 2-D Interpolation errors of 8-point windowed sinc interpolation (left) and seventh-order B-spline interpolation (right). The top graphs show 1-D slices through the center of the images. B-spline interpolation exhibits smaller error and therefore is more accurate.

Accuracy is often negligible in practice. In the rest of the dissertation, I use the classic and better tested B-spline method.

**SEISMIC APPLICATIONS OF FORWARD INTERPOLATION**

For completeness, I conclude this section with two simple examples of forward interpolation in seismic data processing. Figure 25 shows a 3-D impulse response of Stolt migration (Stolt, 1978), computed by using 2-point linear interpolation and 8-point B-spline interpolation. As noted by Ronen (1982) and Harlan (1982), inaccurate interpolation may lead to spurious artifact events in Stolt-migrated images. Indeed, we see several artifacts in the image with linear interpolation (the left plots in Figure 25). The artifacts are removed if we use a more accurate interpolation method (the right plots in Figure 25).

Another simple example is the radial trace transform (Ottolini, 1982). Figure 26 shows a land shot gather contaminated by nearly radial ground-roll. As discussed by Claerbout (1983), Henley (1999, 2000), and Brown and Claerbout (2000a), one can effectively eliminate ground-roll noise by applying a radial trace transform followed by high-pass filtering and the inverse radial transform. Figure 27 shows the result of the forward radial transform of the shot gather in Figure 26 in the radial band.
Figure 25: Stolt-migration impulse response. Left: using linear interpolation. Right: using seventh-order B-spline interpolation. Migration artifacts are removed by a more accurate forward interpolation method.

of the ground-roll noise and the transform error after we go back to the original domain. Comparing the results of using linear and third-order B-spline interpolation, we see once again that the transform artifacts are removed with a more accurate interpolation scheme.

ACKNOWLEDGMENTS

A short conversation with Dave Hale led me to a better understanding of different forward interpolation methods. Tamas Nemeth helped me better understand the general interpolation theory.

REFERENCES

Figure 26: Ground-roll-contaminated shot gather used in a radial transform test.

Figure 27: Radial trace transform results. Top: radial trace domain. Bottom: residual error after the inverse transform. The error should be zero in a radial band from 0 to 0.65 km/s radial velocity. Left: using linear interpolation. Right: using third-order B-spline interpolation.
Daubechies, I., 1992, Ten lectures on wavelets: SIAM.
Shannon, C. E., 1949, Communication in the presence of noise: Proc. I.R.E., 37,
Spitz makes a better assumption for the signal PEF

Jon Claerbout and Sergey Fomel

ABSTRACT
In real-world extraction of signal from data we are not given the needed signal prediction-error filter (PEF). Claerbout has taken $S$, the PEF of the signal, to be that of the data, $S \approx D$. Spitz takes it to be $S \approx D/N$. Where noises are highly predictable in time or space, Spitz gets significantly better results. Theoretically, a reason is that the essential character of a PEF is contained where it is small.

INTRODUCTION
Knowledge of signal spectrum and noise spectrum allows us to find filters for optimally separating data $d$ into two components, signal $s$ and noise $n$ (Claerbout, 1999). Actually, it is the inverses of these spectra which are required. In Claerbout’s textbook example (Claerbout, 1999) he estimates these inverse spectra by estimating prediction-error filters (PEFs) from the data. He estimates both a signal PEF and a noise PEF from the same data $d$. A PEF based on data $d$ might be expected to be named the data PEF $D$, but Claerbout estimates two different PEFS from $d$ and calls them the signal PEF $S$ and the noise PEF $N$. They differ by being estimated with different number of adjustable coefficients, one matching a signal model (two plane waves) having three positions on the space axis, the other matching a noise model having one position on the space axis.

Meanwhile, using a different approach, Spitz (1999) concludes that the signal, noise, and data inverse spectra should be related by $D = SN$. The conclusion we reach in this paper is that Claerbout’s estimate of $S$ is more appropriately an estimate of the data PEF $D$. To find the most appropriate $S$ and $N$ we should use both the “variable templates” idea of Claerbout and the $D \approx SN$ idea of Spitz. Here we first provide a straightforward derivation of the Spitz insight and then we show some experimental results.

1e-mail: claerbout@stanford.edu, sergey@sep.stanford.edu
BASIC THEORY

Signal spectrum plus the noise spectrum gives the data spectrum. Since a prediction-error filter tends to the inverse of a spectrum we have

\[
\frac{1}{DD} = \frac{1}{SS} + \frac{1}{NN} \tag{1}
\]

\[
\frac{1}{DD} = \frac{SS + NN}{SNSN} \tag{2}
\]

or

\[
DD = \frac{SNSN}{SS + NN} \tag{3}
\]

Now we are ready for the Spitz approximation. Spitz builds his applications upon the assumption that we can estimate \(D\) and \(N\) from suitable chunks of raw data. His result may be obtained from (3) by ignoring its denominator getting \(D \approx SN\) or

\[
S \approx \frac{D}{N} \quad (4)
\]

Ignoring the denominator in equation (3), is not so terrible an approximation as it might seem. Remember that PEFs are important where they are small because they are used as weighting functions. Where weighting functions are small, solutions are expected to be large. Although Claerbout’s assumption \(S \approx D\) might be somewhat valid for signal and data spectra, it is much less valid for their PEFs. In practice, signal unpolluted with noise is usually not available. Even a very good chunk of data tends to yield a poor estimate of the signal PEF \(S\) because the holes in the signal spectrum are easily intruded with noise.

Obviously the major difference between \(S \approx D\) and \(S \approx D/N\) is where the noise is large. Thus it is for “organized and predictable” noises (small \(N\)) where we expect to see the main difference.

Theoretically, we need not make the Spitz approximation. We could solve (1) for \(S\) by spectral factorization. Although the \(S\) obtained would be more theoretically satisfying, there would be some practical disadvantages. Getting the signal spectrum by subtracting that of the noise from that of the data leaves the danger of a negative result (which explodes the factorization). Thus, maintaining spectral positivity would require extra care. All these extra burdens are avoided by making the Spitz approximation. All the more so in applications with continuously varying estimates.

SIGNAL AND NOISE SEPARATION

We assume that the data vector \(\mathbf{d}\) is composed of the signal and noise components \(\mathbf{s}\) and \(\mathbf{n}\):

\[
\mathbf{d} = \mathbf{s} + \mathbf{n} . \tag{5}
\]
If both the signal and noise prediction-error filters $S$ and $N$ are known, then the signal can be extracted from the data by solving the following system by the least squares method:

\begin{align}
0 & \approx \mathbf{N} \mathbf{n} = \mathbf{N}(\mathbf{d} - \mathbf{s}) ; \\
0 & \approx \epsilon \mathbf{S} \mathbf{s} ,
\end{align}

where $\epsilon$ is a scalar scaling coefficient, reflecting the presumed signal-to-noise ratio (Claerbout, 1999).

The formal solution of system (6-7) has the form of a projection filter:

$$
\mathbf{s} = \left( \frac{\mathbf{N}' \mathbf{N}}{\mathbf{N}' \mathbf{N} + \epsilon^2 \mathbf{S}' \mathbf{S}} \right) \mathbf{d} .
$$

(8)

Analogously, the signal vector is expressed as

$$
\mathbf{n} = \mathbf{d} - \mathbf{s} = \left( \frac{\epsilon^2 \mathbf{S}' \mathbf{S}}{\mathbf{N}' \mathbf{N} + \epsilon^2 \mathbf{S}' \mathbf{S}} \right) \mathbf{d} .
$$

(9)

In 1-D or $F$-$X$ setting, one can accomplish the division in formulas (8) and (9) directly by spectral factorization and inverse recursive filtering (Soubaras, 1995, 1994). A similar approach can be applied in the case of $T$-$X$ or $F$-$XY$ filtering with the help of the helix transform (Claerbout, 1998; Ozdemir et al., 1999) or by solving system (6-7) directly with an iterative method (Abma, 1995).

Claerbout’s approach, implemented in the examples of GEE (Claerbout, 1999), is to estimate the signal and noise PEFs $S$ and $N$ from the data $\mathbf{d}$ by specifying different shape templates for these two filters. The filter estimates can be iteratively refined after the initial signal and noise separation. In some examples, such as those shown in this paper, the signal and noise templates are not easily separated. When the signal template behaves as an extension of the noise template so that the shape of $S$ completely embeds the shape of $N$, our estimate of $S$ serves as a predictor of both signal and noise. We might as well consider it as $\mathbf{D}$, the prediction-error filter for the data.

Spitz (1999) argues that the data PEF $\mathbf{D}$ can be regarded as the convolution of the signal and noise PEFs $S$ and $N$.

This assertion suggests the following algorithm:

1. Estimate $\mathbf{D}$ and $\mathbf{N}$.

2. Estimate $\mathbf{S}$ by deconvolving (polynomial division) $\mathbf{D}$ by $\mathbf{N}$.

3. Solve the least-square system (6-7).
To avoid the division step, we suggest a simple modification of Spitz's algorithm, which results from multiplying both equations in system (6-7) by the noise filtering operator $N$. The resulting system has the form

$$0 \approx N^2n = N^2(d - s) ;$$  \hspace{1cm} (10)  

$$0 \approx \epsilon NSs = \epsilon Ds .$$  \hspace{1cm} (11)

The modified algorithm is

1. Estimate $D$ and $N$.
2. Convolve $N$ with itself.
3. Solve the least-square system (10-11).

The formal least-squares solution of system (10-11) is

$$s = \left( \frac{N'NN}{N'NN + \epsilon^2D'D} \right) d = \left( \frac{N'NN}{N'NN + \epsilon^2N'S'SN} \right) d .$$  \hspace{1cm} (12)

Comparing (12) with (8), we can see that both the numerator and the denominator in the two expressions differ by the same multiplier $N'N$. This multiplication should not effect the result of projection filtering.

Figure 1 shows a simple example of signal and noise separation taken from GEE (Claerbout, 1999). The signal consists of two crossing plane waves with random amplitudes, and the noise is spatially random. The data and noise $T$-$X$ prediction-error filters were estimated from the same data by applying different filter templates. The template for $D$ is

$$\begin{align*}
a & a \\
1 & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
a & a & a \\
\end{align*}$$

where the $a$ symbol represents adjustable coefficients. The data filter shape has three columns, which allows it to predict two plane waves with different slopes. The noise filter $N$ has only one column. Its template is

$$\begin{align*}
1 \\
a \\
a \\
a \\
\end{align*}$$
The noise PEF can estimate the temporal spectrum but would fail to capture the signal predictability in the space direction. Figure 2 shows the result of applying the modified Spitz method according to equations (10-11). Comparing figures 1 and 2, we can see that using a modified system of equations brings a slightly modified result with more noise in the signal but more signal in the noise. It is as if $\epsilon$ has changed, and indeed this could be the principal effect of neglecting the denominator in equation (3).

![Figure 1: Signal and noise separation with the original GEE method. The input signal is on the left. Next is that signal with random noise added. Next are the estimated signal and the estimated noise.](image)

To illustrate a significantly different result using the Spitz insight we examine the new situation shown in Figures 3 and 4. The wave with the positive slope is considered to be regular noise; the other wave is signal. The noise PEF $N$ was estimated from the data by restricting the filter shape so that it could predict only positive slopes. The corresponding template is

$$
\begin{array}{c}
a \\
1 & a
\end{array}
$$

The data PEF template is

$$
\begin{array}{ccc}
a & a & a \\
a & a \\
1 & a & a \\
a & a & a \\
a & a & a
\end{array}
$$
Figure 2: Signal and noise separation with the modified Spitz method. The input signal is on the left. Next is that signal with random noise added. Next are the estimated signal and the estimated noise. Using the data PEF as a substitute for the signal PEF produces a poor result, shown in Figure 3. We see a part of the signal sneaking into the noise estimate. Using the modified Spitz method, we obtain a clean separation of the plane waves (Figure 4).

Clapp and Brown (1999, 2000) and Brown et al. (1999) show applications of the least-squares signal-noise separation to multiple and ground-roll elimination.

ACKNOWLEDGMENTS

Conversations with our colleagues Bob Clapp and Morgan Brown led us to a better understanding of the Spitz approach.

REFERENCES


——–, 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project.
Figure 3: Plane wave separation with the GEE method. The input signal is on the left. Next is that signal with noise added. Next are the estimated signal and the estimated noise. [spitz/sign/planes90]

Figure 4: Plane wave separation with the modified Spitz method. The input signal is on the left. Next is that signal with noise added. Next are the estimated signal and the estimated noise. [spitz/sign/planes]


Seismic reflection data interpolation with differential offset and shot continuation

Sergey Fomel

ABSTRACT
I propose a finite-difference offset continuation filter for interpolating seismic reflection data. The filter is constructed from the offset continuation differential equation and is applied on frequency slices in the log-stretch frequency domain. Synthetic and real data tests demonstrate that the proposed method succeeds in structurally complex situations where more simplistic approaches fail.

INTRODUCTION
Data interpolation is one of the most important problems of seismic data processing. In 2-D exploration, the interpolation problem arises because of missing near and far offsets, spatial aliasing and occasional bad traces. In 3-D exploration, the importance of this problem increases dramatically because 3-D acquisition almost never provides a complete regular coverage in both midpoint and offset coordinates (Biondi, 1999). Data regularization in 3-D can solve the problem of Kirchoff migration artifacts (Gardner and Canning, 1994), prepare the data for wave-equation common-azimuth imaging (Biondi and Palacharla, 1996), or provide the spatial coverage required for 3-D multiple elimination (van Dedem and Verschuur, 1998).

Claerbout (1992, 1999) formulates the following general principle of missing data interpolation:

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

How can one specify an appropriate filtering for a given interpolation problem? Smooth surfaces are conveniently interpolated with Laplacian filters (Briggs, 1974). Steering filters help us interpolate data with predefined dip fields (Clapp et al., 1998). Prediction-error filters in time-space or frequency-space domain successfully interpolate data composed of distinctive plane waves (Spitz, 1991; Claerbout, 1999). Local plane waves are handled with plane-wave destruction filters (Fomel, 2002). Because prestack seismic data is not stationary in the offset direction, non-stationary prediction-error filters need to be estimated, which leads to an accurate but relatively expensive method with many adjustable parameters (Crawley et al., 1999).

1e-mail: claerbout@stanford.edu, sergey@sep.stanford.edu
A simple model for reflection seismic data is a set of hyperbolic events on a common midpoint gather. The simplest filter for this model is the first derivative in the offset direction applied after the normal moveout correction. Going one step beyond this simple approximation requires taking the dip moveout (DMO) effect into account [Deregowski, 1986]. The DMO effect is fully incorporated in the offset continuation differential equation [Fomel, 1994, 2003].

Offset continuation is a process of seismic data transformation between different offsets [Deregowski and Rocca, 1981; Bolondi et al., 1982; Salvador and Savelli, 1982]. Different types of DMO operators [Hale, 1991] can be regarded as continuation to zero offset and derived as solutions of an initial-value problem with the revised offset continuation equation [Fomel, 2003]. Within a constant-velocity assumption, this equation not only provides correct traveltimes on the continued sections, but also correctly transforms the corresponding wave amplitudes [Fomel et al., 1996]. Integral offset continuation operators have been derived independently by Chemingui and Biondi [1994], Bagaini and Spagnolini [1996], and Stovas and Fomel [1996]. The 3-D analog is known as azimuth moveout (AMO) [Biondi et al., 1998]. In the shot-record domain, integral offset continuation transforms to shot continuation [Schwab, 1993; Bagaini and Spagnolini, 1993; Spagnolini and Opreni, 1996]. Integral continuation operators can be applied directly for missing data interpolation and regularization [Bagaini et al., 1994; Mazzucchelli and Rocca, 1999]. However, they don’t behave well for continuation at small distances in the offset space because of limited integration apertures and, therefore, are not well suited for interpolating neighboring records. Additionally, as all integral (Kirchoff-type) operators they suffer from irregularities in the input geometry. The latter problem is addressed by accurate but expensive inversion to common offset [Chemingui, 1999].

In this paper, I propose an application of offset continuation in the form of a finite-difference filter for Claerbout’s method of missing data interpolation. The filter is designed in the log-stretch frequency domain, where each frequency slice can be interpolated independently. Small filter size and easy parallelization among different frequencies assure a high efficiency of the proposed approach. Although the offset continuation filter lacks the predictive power of non-stationary prediction-error filters, it is much simpler to handle and serves as a good a priori guess of an interpolative filter for seismic reflection data. I first test the proposed method by interpolating randomly missing traces in a constant-velocity synthetic dataset. Next, I apply offset continuation and related shot continuation field to a real data example from the North Sea. Using a pair of offset continuation filters, operating in two orthogonal directions, I successfully regularize a 3-D marine dataset. These tests demonstrate that the offset continuation can perform well in complex structural situations where more simplistic approaches fail.
OFFSET CONTINUATION

A particularly efficient implementation of offset continuation results from a log-stretch transform of the time coordinate (Bolondi et al., 1982), followed by a Fourier transform of the stretched time axis. After these transforms, the offset continuation equation from (Fomel, 2003) takes the form

$$h \left( \frac{\partial^2 \tilde{P}}{\partial y^2} - \frac{\partial^2 \tilde{P}}{\partial h^2} \right) - i \Omega \frac{\partial \tilde{P}}{\partial h} = 0,$$

(1)

where $\Omega$ is the corresponding frequency, $h$ is the half-offset, $y$ is the midpoint, and $\tilde{P}(y, h, \Omega)$ is the transformed data. As in other F-X methods, equation (1) can be applied independently and in parallel on different frequency slices.

We can construct an effective offset-continuation finite-difference filter by studying first the problem of wave extrapolation between neighboring offsets. In the frequency-wavenumber domain, the extrapolation operator is defined by solving the initial-value problem on equation (1). The solution takes the following form (Fomel, 2003):

$$\hat{\tilde{P}}(h_2) = \hat{\tilde{P}}(h_1) Z_\lambda(k h_2)/Z_\lambda(k h_1),$$

(2)

where $\lambda = (1 + i \Omega)/2$, and $Z_\lambda$ is the special function defined as

$$Z_\lambda(x) = \Gamma(1 - \lambda) \left( \frac{x}{2} \right)^\lambda J_{-\lambda}(x) = {}_0F_1 \left( ; 1 - \lambda; -\frac{x^2}{4} \right)$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{\Gamma(1 - \lambda)}{\Gamma(n + 1 - \lambda)} \left( \frac{x}{2} \right)^{2n},$$

(3)

where $\Gamma$ is the gamma function, $J_{-\lambda}$ is the Bessel function, and $_0F_1$ is the confluent hypergeometric limit function (Petkovsek et al., 1996). The wavenumber $k$ in equation (2) corresponds to the midpoint $y$ in the original data domain. In the high-frequency asymptotics, operator (2) takes the form

$$\hat{\tilde{P}}(h_2) = \hat{\tilde{P}}(h_1) F(2kh_2/\Omega)/F(2kh_1/\Omega) \exp \left[ i \Omega \psi(2kh_2/\Omega - 2kh_1/\Omega) \right],$$

(4)

where

$$F(\epsilon) = \sqrt{\frac{1 + \sqrt{1 + \epsilon^2}}{2\sqrt{1 + \epsilon^2}}} \exp \left( \frac{1 - \sqrt{1 + \epsilon^2}}{2} \right),$$

(5)

and

$$\psi(\epsilon) = \frac{1}{2} \left( 1 - \sqrt{1 + \epsilon^2} + \ln \left( \frac{1 + \sqrt{1 + \epsilon^2}}{2} \right) \right).$$

(6)

Returning to the original domain, we can approximate the continuation operator with a finite-difference filter with the $Z$-transform

$$\hat{P}_{h+1}(Z_y) = \hat{P}_h(Z_y) \frac{G_1(Z_y)}{G_2(Z_y)}.$$

(7)
The coefficients of the filters $G_1(Z_y)$ and $G_2(Z_y)$ are found by fitting the Taylor series coefficients of the filter response around the zero wavenumber. In the simplest case of 3-point filters, this procedure uses four Taylor series coefficients and leads to the following expressions:

\[
G_1(Z_y) = 1 - \frac{1 - c_1(\Omega) h_2^2 + c_2(\Omega) h_1^2}{6} + \frac{1 - c_1(\Omega) h_2^2 + c_2(\Omega) h_1^2}{12} (Z_y + Z_y^{-1}) ,
\]

\[
G_2(Z_y) = 1 - \frac{1 - c_1(\Omega) h_1^2 + c_2(\Omega) h_2^2}{6} + \frac{1 - c_1(\Omega) h_1^2 + c_2(\Omega) h_2^2}{12} (Z_y + Z_y^{-1}) ,
\]

where

\[
c_1(\Omega) = \frac{3 (\Omega^2 + 9 - 4i \Omega)}{\Omega^2 (3 + i \Omega)}
\]

and

\[
c_2(\Omega) = \frac{3 (\Omega^2 - 27 - 8i \Omega)}{\Omega^2 (3 + i \Omega)} .
\]

Figure 1 compares the phase characteristic of the finite-difference extrapolators with the phase characteristics of the exact operator and the asymptotic operator. The match between different phases is poor for very low frequencies (left plot in Figure 1) but sufficiently accurate for frequencies in the typical bandwidth of seismic data (right plot in Figure 1).

Figure 2 compares impulse responses of the inverse DMO operator constructed by the asymptotic $\Omega - k$ operator with those constructed by finite-difference offset continuation. Neglecting subtle phase inaccuracies at large dips, the two images look similar, which provides an experimental evidence of the accuracy of the proposed finite-difference scheme.

When applied on the offset-midpoint plane of an individual frequency slice, the one-dimensional implicit filter transforms to a two-dimensional explicit filter with the 2-D $Z$-transform

\[
G(Z_y, Z_h) = G_1(Z_y) - Z_h G_2(Z_y) .
\]

Convolution with filter is the regularization operator that I propose to use for interpolating prestack seismic data.

\section*{APPLICATION}

I start numerical testing of the proposed regularization first on a constant-velocity synthetic, where all the assumptions behind the offset continuation equation are valid.
Figure 1: Phase of the implicit offset-continuation operators in comparison with the exact solution. The offset increment is assumed to be equal to the midpoint spacing. The left plot corresponds to $\Omega = 1$, the right plot to $\Omega = 10$.

Figure 2: Inverse DMO impulse responses computed by the Fourier method (left) and by finite-difference offset continuation (right). The offset is 1 km.
Constant-velocity synthetic

A sinusoidal reflector shown in Figure 3 creates complicated reflection data, shown in Figures 1 and 5. To set up a test for regularization by offset continuation, I removed 90% of randomly selected shot gathers from the input data. The syncline parts of the reflector lead to traveltime triplications at large offsets. A mixture of different dips from the triplications would make it extremely difficult to interpolate the data in individual common-offset gathers, such as those shown in Figure 1. The plots of time slices after NMO (Figure 5) clearly show that the data are also non-stationary in the offset direction. Therefore, a simple offset interpolation scheme is also doomed.

Figure 6 shows the reconstruction process on individual frequency slices. Despite the complex and non-stationary character of the reflection events in the frequency domain, the offset continuation equation is able to accurately reconstruct them from the decimated data.

Figure 7 shows the result of interpolation after the data are transformed back to the time domain. The offset continuation result (right plots in Figure 7) reconstructs the ideal data (left plots in Figure 1) very accurately even in the complex triplication zones, while the result of simple offset interpolation (left plots in Figure 7) fails as expected. The simple interpolation scheme applied the offset derivative \( \frac{\partial}{\partial h} \) in place of the offset continuation equation and thus did not take into account the movement of the events across different midpoints.

The constant-velocity test results allow us to conclude that, when all the assumptions of the offset continuation theory are met, it provides a powerful method of data regularization.

Being encouraged by the synthetic results, I proceed to a three-dimensional real data test.

---

2 An analogous technique applied to the case of wavefield depth extrapolation with the wave equation would lead to the famous 45-degree implicit finite-difference operator (Claerbout, 1985).
Figure 4: Prestack common-offset gathers for the constant-velocity test. Left: ideal data (after NMO). Right: input data (90% of shot gathers removed). Top, center, and bottom plots correspond to different offsets.
Figure 5: Time slices of the prestack data for the constant-velocity test. Left: ideal data (after NMO). Right: input data (90% of random gathers removed). Top, center, and bottom plots correspond to time slices at 0.3, 0.4, and 0.5 s.
Figure 6: Interpolation in frequency slices. Left: input data (90% of the shot gathers removed). Right: interpolation output. Top, bottom, and middle plots correspond to different frequencies. The real parts of the complex-valued data are shown.
Figure 7: Interpolation in common-offset gathers. Left: output of simple offset interpolation. Right: output of offset continuation interpolation. Compare with Figure [7]. Top, center, and bottom plots correspond to different common-offset gathers.
Analogously to integral azimuth moveout operator (Biondi et al., 1998), differential offset continuation can be applied in 3-D for regularizing seismic data prior to prestack imaging.

In the next section, I return to the 2-D case to consider an important problem of shot gather interpolation.

**SHOT CONTINUATION**

Missing or under-sampled shot records are a common example of data irregularity (Crawley, 2000). The offset continuation approach can be easily modified to work in the shot record domain. With the change of variables \( s = y - h \), where \( s \) is the shot location, the frequency-domain equation (1) transforms to the equation

\[
\begin{aligned}
    h \left( 2 \frac{\partial^2 \tilde{P}}{\partial s \partial h} - \frac{\partial^2 \tilde{P}}{\partial h^2} \right) - i \Omega \left( \frac{\partial \tilde{P}}{\partial h} - \frac{\partial \tilde{P}}{\partial s} \right) &= 0 .
\end{aligned}
\]

(11)

Unlike equation (1), which is second-order in the propagation variable \( h \), equation (11) contains only first-order derivatives in \( s \). We can formally write its solution for the initial conditions at \( s = s_1 \) in the form of a phase-shift operator:

\[
\tilde{P}(s_2) = \tilde{P}(s_1) \exp \left[ i k_h (s_2 - s_1) \frac{k_h h - \Omega}{2 k_h h - \Omega} \right],
\]

(12)

where the wavenumber \( k_h \) corresponds to the half-offset \( h \). Equation (12) is in the mixed offset-wavenumber domain and, therefore, not directly applicable in practice. However, we can use it as an intermediate step in designing a finite-difference shot continuation filter. Analogously to the cases of plane-wave destruction and offset continuation, shot continuation leads us to the rational filter

\[
\hat{P}_{s+1}(Z_h) = \hat{P}_s(Z_h) \frac{S(Z_h)}{S(1/Z_h)},
\]

(13)

The filter is non-stationary, because the coefficients of \( S(Z_h) \) depend on the half-offset \( h \). We can find them by the Taylor expansion of the phase-shift equation (12) around zero wavenumber \( k_h \). For the case of the half-offset sampling equal to the shot sampling, the simplest three-point filter is constructed with three terms of the Taylor expansion. It takes the form

\[
S(Z_h) = - \left( \frac{1}{12} + i \frac{h}{2 \Omega} \right) Z_h^{-1} + \left( \frac{2}{3} - i \frac{\Omega^2 + 12 h^2}{12 \Omega h} \right) + \left( \frac{5}{12} + i \frac{\Omega^2 + 18 h^2}{12 \Omega h} \right) Z_h .
\]

(14)

Let us consider the problem of doubling the shot density. If we use two neighboring shot records to find the missing record between them, the problem reduces to the least-squares system

\[
\begin{bmatrix}
    S \\ \bar{S}
\end{bmatrix} p_s \approx \begin{bmatrix}
    S p_{s-1} \\ \bar{S} p_{s+1}
\end{bmatrix},
\]

(15)
where $S$ denotes convolution with the numerator of equation (13), $\bar{S}$ denotes convolution with the corresponding denominator, $p_{s-1}$ and $p_{s+1}$ represent the known shot gathers, and $p_s$ represents the gather that we want to estimate. The least-squares solution of system (15) takes the form

$$p_s = \left( S^T S + \bar{S}^T \bar{S} \right)^{-1} \left( S^T \bar{S} p_{s-1} + \bar{S}^T S p_{s+1} \right).$$  

(16)

If we choose the three-point filter (14) to construct the operators $S$ and $\bar{S}$, then the inverted matrix in equation (16) will have five non-zero diagonals. It can be efficiently inverted with a direct banded matrix solver using the $LDL^T$ decomposition (Golub and Van Loan, 1996). Since the matrix does not depend on the shot location, we can perform the decomposition once for every frequency so that only a triangular matrix inversion will be needed for interpolating each new shot. This leads to an extremely efficient algorithm for interpolating intermediate shot records.

Sometimes, two neighboring shot gathers do not fully constrain the intermediate shot. In order to add an additional constraint, I include a regularization term in equation (16), as follows:

$$p_s = \left( S^T S + \bar{S}^T \bar{S} + \epsilon^2 A^T A \right)^{-1} \left( S^T \bar{S} p_{s-1} + \bar{S}^T S p_{s+1} \right),$$

(17)

where $A$ represents convolution with a three-point prediction-error filter (PEF), and $\epsilon$ is a scaling coefficient. The appropriate PEF can be estimated from $p_{s-1}$ and $p_{s+1}$ using Burg’s algorithm (Burg, 1972, 1975; Claerbout, 1976). A three-point filter does not break the five-diagonal structure of the inverted matrix. The PEF regularization attempts to preserve offset dip spectrum in the under-constrained parts of the estimated shot gather.

Figure 9 shows the result of a shot interpolation experiment using the constant-velocity synthetic from Figure 1. In this experiment, I removed one of the shot gathers from the original NMO-corrected data and interpolated it back using equation (17). Subtracting the true shot gather from the reconstructed one shows a very insignificant error, which is further reduced by using the PEF regularization (right plots in Figure 9). The two neighboring shot gathers used in this experiment are shown in the top plots of Figure 8. For comparison, the bottom plots in Figure 8 show the simple average of the two shot gathers and its corresponding prediction error. As expected, the error is significantly larger than the error of shot continuation. An interpolation scheme based on local dips in the shot direction would probably achieve a better result, but it is significantly more expensive than the shot continuation scheme introduced above.

A similar experiment with real data from a North Sea marine dataset is reported in Figure 11. I removed and reconstructed a shot gather from the two neighboring gathers shown in Figure 10. The lower parts of the gathers are complicated by salt dome reflections and diffractions with conflicting dips. The simple average of the two input shot gathers (bottom plots in Figure 11) works reasonably well for nearly flat reflection events but fails to predict the position of the back-scattered diffractions.
Figure 8: Top: Two synthetic shot gathers used for the shot interpolation experiment. An NMO correction has been applied. Bottom: simple average of the two shot gathers (left) and its prediction error (right).
Figure 9: Synthetic shot interpolation results. Left: interpolated shot gathers. Right: prediction errors (the differences between interpolated and true shot gathers), plotted on the same scale. 

Analogously to the case of offset continuation, it is possible to extend the shot continuation method to three dimensions. A simple modification of the proposed technique would also allow us to use more than two shot gathers in the input or to extrapolate missing shot gathers at the end of survey lines.

CONCLUSIONS

Differential offset continuation provides a valuable tool for interpolation and regularization of seismic data. Starting from analytical frequency-domain solutions of the offset continuation differential equation, I have designed accurate finite-difference filters for implementing offset continuation as a local convolutional operator. A similar technique works for shot continuation across different shot gathers. Missing data are efficiently interpolated by an iterative least-squares optimization. The differential filters have an optimally small size, which assures high efficiency.

Differential offset continuation serves as a bridge between integral and convolutional approaches to data interpolation. It shares the theoretical grounds with the
integral approach but is applied in a manner similar to that of prediction-error filters in the convolutional approach.

Tests with synthetic and real data demonstrate that the proposed interpolation method can succeed in complex structural situations where more simplistic methods fail.

ACKNOWLEDGMENTS

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

For the shot continuation test, I used a North Sea dataset, released to SEP by Elf Aquitaine.

I thank Jon Claerbout and Biondo Biondi for helpful discussions about the practical application of differential offset continuation.

REFERENCES

Figure 11: Real-data shot interpolation results. Top: interpolated shot gather (left) and its prediction error (right). Bottom: simple average of the two input shot gathers (left) and its prediction error (right).
Estimating the signal PEF


Biondi, B. L., 1999, 3-D Seismic Imaging: Stanford Exploration Project.


———, 1985, Imaging the Earth’s Interior: Blackwell Scientific Publications.


Fomel, S. B., 1994, Kinematically equivalent differential operator for offset continu-
tion of seismic sections: Russian Geology and Geophysics, 35, 122–134.
Applications of plane-wave destruction filters

Sergey Fomel

ABSTRACT
Plane-wave destruction filters originate from a local plane-wave model for characterizing seismic data. These filters can be thought of as a $T$-$X$ analog of $F$-$X$ prediction-error filters and as an alternative to $T$-$X$ prediction-error filters. The filters are constructed with the help of an implicit finite-difference scheme for the local plane-wave equation. On several synthetic and real-data examples, I demonstrate that finite-difference plane-wave destruction filters perform well in applications such as fault detection, data interpolation, and noise attenuation.

INTRODUCTION
Plane-wave destruction filters, introduced by Claerbout (1992), serve the purpose of characterizing seismic images by a superposition of local plane waves. They are constructed as finite-difference stencils for the plane-wave differential equation. In many cases, a local plane-wave model is a very convenient representation of seismic data. Unfortunately, early experiences with applying plane-wave destructors for interpolating spatially aliased data (Nichols, 1990; Claerbout, 1992) demonstrated their poor performance in comparison with that of industry-standard $F$-$X$ prediction-error filters (Spitz, 1991).

For each given frequency, an $F$-$X$ prediction-error filter (PEF) can be thought of as a $Z$-transform polynomial. The roots of the polynomial correspond precisely to predicted plane waves (Canales, 1984). Therefore, $F$-$X$ PEFs simply represent a spectral (frequency-domain) approach to plane-wave destruction. This powerful and efficient approach is, however, not theoretically adequate when the plane-wave slopes or the boundary conditions vary both spatially and temporally. In practice, this limitation is addressed by breaking the data into windows and assuming that the slopes are stationary within each window.

Multidimensional $T$-$X$ prediction-error filters (Claerbout, 1992, 1999) share the same purpose of predicting local plane waves. They work well with spatially aliased data and allow for both temporal and spatial variability of the slopes. In practice, however, $T$-$X$ filters appear as very mysterious objects, because their construction

1 e-mail: sergey@sep.stanford.edu
2 The filters are designed to destruct local plane waves. However, in applications such as data interpolation, they are often used to reconstruct the missing parts of local waves. The choice of terminology should not confuse the reader.
involves many non-intuitive parameters. The user needs to choose a raft of parameters, such as the number of filter coefficients, the gap and the exact shape of the filter, the size, number, and shape of local patches for filter estimation, the number of iterations, and the amount of regularization. Recently developed techniques for handling non-stationary PEFs (Crawley et al., 1999) performed well in a variety of applications (Crawley, 2000; Guitton et al., 2001), but the large number of adjustable parameters still requires a significant level of human interaction and remains the drawback of the method.

Clapp et al. (1998) have recently revived the original plane-wave destructors for preconditioning tomographic problems with a predefined dip field (Clapp, 2001). The filters were named steering filters because of their ability to steer the solution in the direction of the local dips. The name is also reminiscent of steerable filters used in medical image processing (Freeman and Adelson, 1991; Simoncelli and Farid, 1996).

In this paper, I revisit Claerbout’s original technique of finite-difference plane-wave destruction. First, I develop an approach for increasing the accuracy and dip bandwidth of the method. Applying the improved filter design to several data regularization problems, I discover that the finite-difference filters often perform as well as, or even better than, T-X PEFs. At the same time, they keep the number of adjustable parameters to a minimum, and the only estimated quantity has a clear physical meaning of the local plane-wave slope. No local windows are required, because the slope is estimated as a smoothly variable continuous function of the data coordinates.

Conventional methods for estimating plane-wave slopes are based on picking maximum values of stacking semblance and other cumulative coherency measures (Neidell and Taner, 1971). The differential approach to slope estimation, employed by plane-wave destruction filters, is related to the differential semblance method (Symes and Carazzone, 1991). Its theoretical superiority to conventional semblance measures for the problem of local plane wave detection has been established by Symes (1994) and Kim and Symes (1998).

HIGH-ORDER PLANE-WAVE DESTRUCTORS

Following the physical model of local plane waves, we can define the mathematical basis of the plane-wave destruction filters as the local plane differential equation

\[
\frac{\partial P}{\partial x} + \sigma \frac{\partial P}{\partial t} = 0 ,
\]

(1)

where \( P(t, x) \) is the wave field, and \( \sigma \) is the local slope, which may also depend on \( t \) and \( x \). In the case of a constant slope, equation (2) has the simple general solution

\[
P(t, x) = f(t - \sigma x) ,
\]

(2)

where \( f(t) \) is an arbitrary waveform. Equation (2) is nothing more than a mathematical description of a plane wave.
If we assume that the slope $\sigma$ does not depend on $t$, we can transform equation (2) to the frequency domain, where it takes the form of the ordinary differential equation

$$\frac{d\hat{P}}{dx} + i\omega \sigma \hat{P} = 0$$

and has the general solution

$$\hat{P}(x) = \hat{P}(0) e^{i\omega \sigma x},$$

where $\hat{P}$ is the Fourier transform of $P$. The complex exponential term in equation (4) simply represents a shift of a $t$-trace according to the slope $\sigma$ and the trace separation $x$.

In the frequency domain, the operator for transforming the trace at position $x-1$ to the neighboring trace and at position $x$ is a multiplication by $e^{i\omega \sigma}$. In other words, a plane wave can be perfectly predicted by a two-term prediction-error filter in the F-X domain:

$$a_0 \hat{P}(x) + a_1 \hat{P}(x-1) = 0,$$

where $a_0 = 1$ and $a_1 = -e^{i\omega \sigma}$. The goal of predicting several plane waves can be accomplished by cascading several two-term filters. In fact, any F-X prediction-error filter represented in the Z-transform notation as

$$A(Z_x) = 1 + a_1 Z_x + a_2 Z_x^2 + \cdots + a_N Z_x^N$$

(6)

can be factored into a product of two-term filters:

$$A(Z_x) \left( 1 - \frac{Z_x}{Z_1} \right) \left( 1 - \frac{Z_x}{Z_2} \right) \cdots \left( 1 - \frac{Z_x}{Z_N} \right),$$

(7)

where $Z_1, Z_2, \ldots, Z_N$ are the zeroes of polynomial (6). According to equation (5), the phase of each zero corresponds to the slope of a local plane wave multiplied by the frequency. Zeroes that are not on the unit circle carry an additional amplitude gain not included in equation (3).

In order to incorporate time-varying slopes, we need to return to the time domain and look for an appropriate analog of the phase-shift operator and the plane-prediction filter. An important property of plane-wave propagation across different traces is that the total energy of the propagating wave stays invariant throughout the process: the energy of the wave at one trace is completely transmitted to the next trace. This property is assured in the frequency-domain solution by the fact that the spectrum of the complex exponential $e^{i\omega \sigma}$ is equal to one. In the time domain, we can reach an equivalent effect by using an all-pass digital filter. In the Z-transform notation, convolution with an all-pass filter takes the form

$$\hat{P}_{x+1}(Z_t) = \hat{P}_x(Z_t) \frac{B(Z_t)}{B(1/Z_t)},$$

(8)

For simplicity, it is assumed that $x$ takes integer values that correspond to trace numbering.
where $\hat{P}_x(Z_t)$ denotes the $Z$-transform of the corresponding trace, and the ratio $B(Z_t)/B(1/Z_t)$ is an all-pass digital filter approximating the time-shift operator $e^{i\omega \sigma}$. In finite-difference terms, equation (8) represents an implicit finite-difference scheme for solving equation (2) with the initial conditions at a constant $x$. The coefficients of filter $B(Z_t)$ can be determined, for example, by fitting the filter frequency response at low frequencies to the response of the phase-shift operator. The Taylor series technique (equating the coefficients of the Taylor series expansion around zero frequency) yields the expression

$$B_3(Z_t) = \frac{(1 - \sigma)(2 - \sigma)}{12} Z_t^{-1} + \frac{(2 + \sigma)(2 - \sigma)}{6} + \frac{(1 + \sigma)(2 + \sigma)}{12} Z_t$$

(9)

for a three-point centered filter $B_3(Z_t)$ and the expression

$$B_5(Z_t) = \frac{(1 - \sigma)(2 - \sigma)(3 - \sigma)(4 - \sigma)}{1680} Z_t^{-2} + \frac{(4 - \sigma)(2 - \sigma)(3 - \sigma)(4 - \sigma)}{420} Z_t^{-1} + \frac{(4 - \sigma)(3 - \sigma)(3 + \sigma)(4 + \sigma)}{280} + \frac{(4 - \sigma)(2 + \sigma)(3 + \sigma)(4 + \sigma)}{420} Z_t + \frac{(1 + \sigma)(2 + \sigma)(3 + \sigma)(4 + \sigma)}{1680} Z_t^2$$

(10)

for a five-point centered filter $B_5(Z_t)$. The derivation of equations (9-10) is detailed in the appendix. It is easy to generalize these equations to longer filters.

![Figure 1](image)

Figure 1 shows the phase of the all-pass filters $B_3(Z_t)/B_3(1/Z_t)$ and $B_5(Z_t)/B_5(1/Z_t)$ for two values of the slope $\sigma$ in comparison with the exact linear function of equation (4). As expected, the phases match the exact line at low frequencies, and the accuracy of the approximation increases with the length of the filter.

Taking both dimensions into consideration, equation (8) transforms to the prediction equation analogous to (5) with the 2-D prediction filter

$$A(Z_t, Z_x) = 1 - Z_x \frac{B(Z_t)}{B(1/Z_t)}$$

(11)

In order to characterize several plane waves, we can cascade several filters of the form (11) in a manner similar to that of equation (7). In the examples of this paper, I use a modified version of the filter $A(Z_t, Z_x)$, namely the filter

$$C(Z_t, Z_x) = A(Z_t, Z_x) B(1/Z_t) = B(1/Z_t) - Z_x B(Z_t)$$

(12)

which avoids the need for polynomial division. In case of the 3-point filter (9), the 2-D filter (12) has exactly six coefficients. It consists of two columns, each column having three coefficients and the second column being a reversed copy of the first one. When filter (12) is used in data regularization problems, it can occasionally cause undesired high-frequency oscillations in the solution, resulting from the near-Nyquist zeroes of the polynomial $B(Z_t)$. The oscillations are easily removed in practice with appropriate low-pass filtering.

In the next section, I address the problem of estimating the local slope $\sigma$ with filters of form (12). Estimating the slope is a necessary step for applying the finite-difference plane-wave filters on real data.
Figure 1: Phase of the implicit finite-difference shift operators in comparison with the exact solution. The left plot corresponds to the slope of $\sigma = 0.5$, the right plot to $\sigma = 0.8$.

**SLOPE ESTIMATION**

Let us denote by $C(\sigma)$ the operator of convolving the data with the 2-D filter $C(Z_t, Z_x)$ of equation (12), assuming the local slope $\sigma$ is known. In order to determine the slope, we can define the least-squares goal

$$C(\sigma) \mathbf{d} \approx 0,$$

where $\mathbf{d}$ is the known data and the approximate equality implies that the solution is found by minimizing the power of the left-hand side. Equations (9) and (10) show that the slope $\sigma$ enters in the filter coefficients in an essentially non-linear way. However, one can still apply the linear iterative optimization methods by an analytical linearization of equation (13). The linearization (also known as the Gauss-Newton iteration) implies solving the linear system

$$C'(\sigma_0) \Delta \sigma \mathbf{d} + C(\sigma_0) \mathbf{d} \approx 0$$

for the slope increment $\Delta \sigma$. Here $\sigma_0$ is the initial slope estimate, and $C'(\sigma)$ is a convolution with the filter, obtained by differentiating the filter coefficients of $C(\sigma)$ with respect to $\sigma$. After system (14) is solved, the initial slope $\sigma_0$ is updated by adding $\Delta \sigma$ to it, and one can solve the linear problem again. Depending on the starting solution, the method may require several non-linear iterations to achieve an acceptable convergence.

The slope $\sigma$ in equation (14) does not have to be constant. We can consider it as varying in both time and space coordinates. This eliminates the need for local
windows but may lead to undesirably rough (oscillatory) local slope estimates. Moreover, the solution will be undefined in regions of unknown or constant data, because for these regions the local slope is not constrained. Both these problems are solved by adding a regularization (styling) goal to system (14). The additional goal takes the form

$$\epsilon D \Delta \sigma \approx 0,$$

(15)

where $D$ is an appropriate roughening operator and $\epsilon$ is a scaling coefficient. For simplicity, I chose $D$ to be the gradient operator. More efficient and sophisticated helical preconditioning techniques are available (Claerbout 1998; Fomel 2001; Fomel and Claerbout 2002).

In theory, estimating two different slopes $\sigma_1$ and $\sigma_2$ from the available data is only marginally more complicated than estimating a single slope. The convolution operator becomes a cascade of $C(\sigma_1)$ and $C(\sigma_2)$, and the linearization yields

$$C'(\sigma_1) C(\sigma_2) \Delta \sigma_1 d + C(\sigma_1) C'(\sigma_2) \Delta \sigma_2 d + C(\sigma_1) C(\sigma_2) d \approx 0.$$  

(16)

The regularization condition should now be applied to both $\Delta \sigma_1$ and $\Delta \sigma_2$:

$$\epsilon D \Delta \sigma_1 \approx 0;$$  

(17)

$$\epsilon D \Delta \sigma_2 \approx 0.$$  

(18)

The solution will obviously depend on the initial values of $\sigma_1$ and $\sigma_2$, which should not be equal to each other. System (16) is generally underdetermined, because it contains twice as many estimated parameters as equations: The number of equations corresponds to the grid size of the data $d$, while characterizing variable slopes $\sigma_1$ and $\sigma_2$ on the same grid involves two gridded functions. However, an appropriate choice of the starting solution and the additional regularization (17-18) allow us to arrive at a practical solution.

The application examples of the next section demonstrate that when the system of equations (14-15) or (16-18) are optimized in the least-squares sense in a cycle of several linearization iterations, it leads to smooth and reliable slope estimates. The regularization conditions (15) and (17-18) assure a smooth extrapolation of the slope to the regions of unknown or constant data.

**APPLICATION EXAMPLES**

In this section, I examine the performance of the finite-difference plane-destruction filters on several test applications. The general framework for applying these filters consists of the two steps:

1. Estimate the dominant local slope (or a set of local slopes) from the data. This step follows the least-squares optimization embedded in equations (14) or (16). Thanks to the general regularization technique of equations (15) and (17-18),
locally smooth slope estimates are obtained without any need for breaking the data into local windows. Of course, local windows can be employed for other purposes (parallelization, memory management, etc.) Selecting appropriate initial values for the local slopes can speed up the computation and steer it towards desirable results. It is easy to incorporate additional constraints on the local slope values.

2. Using the estimated slope, apply non-stationary plane-wave destruction filters for the particular application purposes. In the fault detection application, we simply look at the output of plane-wave destruction. In the interpolation application, the filters are used to constrain the missing data. In the noise attenuation application, they characterize the coherent signal and noise components in the data.

A description of these particular applications follows next.

**Fault detection**

The use of prediction-error filters in the problem of detecting local discontinuities was suggested by Claerbout (1994, 1999), and further refined by Schwab et al. (1996) and Schwab (1998). Bednar (1998) used simple plane-destruction filters in a similar setting to compute coherency attributes.

To test the performance of the improved plane-wave destructors, I chose several examples from Claerbout (1999). Figure 2 introduces the first example. The left plot of the figure shows a synthetic model, which resembles sedimentary layers with a plane unconformity and a curvilinear fault. The model contains 200 traces of 200 samples each. The right plot shows the corresponding texture (Claerbout and Brown, 1999), obtained by convolving a field of random numbers with the inverse of plane-wave destruction filters. The inverses are constructed using helical filtering techniques (Claerbout, 1998; Fomel, 2001). Texture plots allow us to quickly access the ability of the destruction filters to characterize the main locally plane features in the data. The dip field was estimated by the linearization method of the previous section. The dip field itself and the prediction residual [the left-hand side of equation (13)] are shown in the left and right plots of Figure 3 respectively. We observe that the texture plot does reflect the dip structure of the input data, which indicates that the dip field was estimated correctly. The fault and unconformity are clearly visible both in the dip estimate and in the residual plots. Anywhere outside the slope discontinuities and the boundaries, the residual is close to zero. Therefore, it can be used directly as a fault detection measure. Comparing the residual plot in Figure 3 with the analogous plot of Claerbout (1994, 1999), reproduced in Figure 4, establishes a superior performance of the improved finite-difference destructors in comparison with that of the local T-X prediction-error filters.
Figure 2: Synthetic sedimentary model. Left plot: Input data. Right plot: Its texture. The texture is computed by convolving a field number with the inverse of plane-wave destruction filters. It highlights the position of estimated local plane waves.

Figure 3: Synthetic sedimentary model. Left plot: Estimated dip field. Right plot: Prediction residual. Large absolute residual indicates the location of faults.
The left plot in Figure 5 introduces a simpler synthetic test. The model is composed of linear events with two conflicting slopes. A regularized dip field estimation attempts to smooth the estimated dip in the places where it is not constrained by the data (the left plot of Figure 6). The effect of smoothing is clearly seen in the texture image (the right plot in Figure 5). The corresponding residual (the right plot of Figure 6) shows suppressed linear events and highlights the places of their intersection. Residuals are large at intersections because a single dominant dip model fails to adequately represent both conflicting dips.

The left plot in Figure 7 shows a real shot gather: a portion of Yilmaz and Cumro (1983) data set 27. The initial dip in the dip estimation program was set to zero. Therefore, the texture image (the right plot in Figure 7) contains zero-dipping plane waves in the places of no data. Everywhere else the dip is accurately estimated from the data. The data contain a missing trace at about 0.7 km offset and a slightly shifted (possibly mispositioned) trace at about 1.1 km offset. The mispositioned trace is clearly visible in the dip estimate (the left plot in Figure 8), and the missing trace is emphasized in the residual image (the right plot in Figure 8). Additionally, the residual image reveals the forward and back-scattered surface waves, hidden under more energetic reflections in the input data.

Figure 9 shows a stacked time section from the Gulf of Mexico and its corresponding texture. The texture plot demonstrates that the estimated dip (the left plot of Figure 10) reflects the dominant local dip in the data. After the plane waves with the dominant dip are removed, many hidden diffractions appear in the residual image (the right plot in Figure 10). The enhanced diffraction events can be used, for example, for estimating the medium velocity (Harlan et al., 1984).
Figure 5: Conflicting dips synthetic. Left plot: Input data. Right plot: Its texture.

Figure 6: Conflicting dips synthetic. Left plot: Estimated dip field. Right plot: Prediction residual. Large absolute residual indicates the location of conflicting dips.
Figure 7: Real shot gather. Left plot: Input data. Right plot: Its texture.

[pwd/lomo/yc27-txr]
Figure 8: Real shot gather. Left plot: Estimated dip field. Right plot: Prediction residual. The residual highlights surface waves hidden under dominant reflection events in the original data. [pwd/lomo/ yc27-dip]
Figure 9: Time section from the Gulf of Mexico. Left plot: Input data. Right plot: Its texture. The texture plot shows dominant local dips estimated from the data.

[pwd/lomo/ dgulf-txr]
Figure 10: Time section from the Gulf of Mexico. Left plot: Estimated dip field. Right plot: Prediction residual. The residual highlights diffraction events hidden under dominant reflections in the original data.
Overall, the examples of this subsection show that the finite-difference plane-wave destructors provide a reliable tool for enhancement of discontinuities and conflicting slopes in seismic images. The estimation step of the fault detection procedure produces an image of the local dominant dip field, which may have its own interpretational value. An extension to 3-D is possible, as outlined by Schwab (1998), Clapp (2001), and Fomel (2001).

Trace interpolation beyond aliasing

Spitz (1991) popularized the application of prediction-error filters to regular trace interpolation and showed how the spatial aliasing restriction can be overcome by scaling the lower frequencies of $F$-$X$ PEFs. An analogous technique for $T$-$X$ filters was developed by Claerbout (1992, 1999) and was applied for 3-D interpolation with non-stationary PEFs by Crawley (2000). The $T$-$X$ technique implies stretching the filter in all directions so that its dip spectrum is preserved while the coefficients are estimated at alternating traces. After the filter is estimated, it is scaled back and used for interpolating missing traces between the known ones. A very similar method works for finite-difference plane wave destructors, only we need to take special care of the aliased dips at the dip estimation stage.

A simple synthetic example of interpolation beyond aliasing is shown in Figure 11. The input data are clearly aliased and non-stationary. To take the aliasing into account, I estimate the two dips present in the data with the slope estimation technique of equations (16) and (17-18). The first dip corresponds to the true slope, while the second dip corresponds to the aliased dip component. In this example, the true dip is non-negative everywhere and is easily distinguished from the aliased one. In the more general case, an additional interpretation may be required to determine which of the dip components is contaminated by aliasing. Throwing away the aliased dip and interpolating intermediate traces with the true dip produces the accurate interpolation result shown in the right plot of Figure 11. Three additional traces were inserted between each of the neighboring input traces.

Figure 12 shows a marine 2-D shot gather from a deep water Gulf of Mexico survey before and after subsampling in the offset direction. The data are similar to those used by Crawley (2000). The shot gather has long-period multiples and complicated diffraction events caused by a salt body. The amplitudes of the hyperbolic events are not as uniformly distributed as in the synthetic case of Figure 11. Subsampling by a factor of two (the right plot in Figure 12) causes clearly visible aliasing in the steeply dipping events. The goal of the experiment is to interpolate the missing traces in the subsampled data and to compare the result with the original gather shown in the left plot of Figure 12.

A straightforward application of the dip estimation equations (16-18) applied to aliased data can easily lead to erroneous aliased dip estimation because the aliased dip may get picked instead of the true dip. In order to avoid this problem, I chose a
slightly more complex strategy. The algorithm for trace interpolation of aliased data consists of the following steps:

1. Applying Claerbout’s $T$-$X$ methodology, stretch a two-dip plane-wave destruction filter and estimate the dips from decimated data.

2. The second estimated dip will be degraded by aliasing. Ignore this initial second-dip estimate.

3. Estimate the second dip component again by fixing the first dip component and using it as the initial estimate of the second component. This trick prevents the nonlinear estimation algorithm from picking the wrong (aliased) dip in the data.

4. Downscale the estimated two-dip filter and use it for interpolating missing traces.

The two estimated dip components are shown in Figure 13. The first component contains only positive dips. The second component coincides with the first one in the areas where only a single dip is present in the data. In other areas, it picks the complementary dip, which has a negative value for back-dipping hyperbolic diffractions.

Figure 14 shows the interpolation result and the difference between the interpolated traces and the original traces, plotted at the same clip value. The method succeeded in the sense that it is impossible to distinguish interpolated traces from
Figure 12: 2-D marine shot gather. Left: original. Right: subsampled by a factor of two in the offset direction.
Figure 13: Two components of the estimated dip field for the decimated 2-D marine shot gather.
the interpolation result alone. However, it is not ideal, because some of the original energy is missing in the output. A close-up comparison between the original and the interpolated traces in Figure [15] shows that imperfection in more detail. Some of the steepest events in the middle of the section are poorly interpolated, and in some of the other places, the second dip component is continued instead of the first one.

One could improve the interpolation result considerably by including another dimension. To achieve a better result, we can use a pair of plane-wave destructors, one predicting local plane waves in the offset direction and the other predicting local plane waves in the shot direction.

Signal and noise separation

Signal and noise separation and noise attenuation are yet another important application of plane-wave prediction filters. A random noise attention has been successfully addressed by Canales (1984), Gulunay (1986), Abma and Claerbout (1995), Soubaras (1995), and others. A more challenging problem of coherent noise attenuation has only recently joined the circle of the prediction technique applications (Spitz, 1999; Brown and Clapp, 2000; Guitton et al., 2001).

The problem has a very clear interpretation in terms of the local dip components. If two components, $s_1$ and $s_2$ are estimated from the data, and we can interpret the first component as signal, and the second component as noise, then the signal and noise separation problem reduces to solving the least-squares system

$$C(s_1)d_1 \approx 0$$  \hspace{1cm} (19)

$$\epsilon C(s_2)d_2 \approx 0$$  \hspace{1cm} (20)

for the unknown signal and noise components $d_1$ and $d_2$ of the input data $d$:

$$d_1 + d_2 = d.$$  \hspace{1cm} (21)

The scalar parameter $\epsilon$ in equation (20) reflects the signal to noise ratio. We can combine equations (19-20) and (21) in the explicit system for the noise component $d_2$:

$$C(s_1)d_2 \approx C(s_1)d,$$  \hspace{1cm} (22)

$$\epsilon C(s_2)d_2 \approx 0.$$  \hspace{1cm} (23)

Figure [16] shows a simple example of the described approach. I estimated two dip components from the input synthetic data and separated the corresponding events by solving the least-squares system (22-23). The separation result is visually perfect.

Figure [17] presents a significantly more complicated case: a receiver line from of a 3-D land shot gather from Saudi Arabia, contaminated with three-dimensional ground-roll, which appears hyperbolic in the cross-section. The same dataset has been
Figure 14: Left: 2-D marine shot gather after trace interpolation. Right: Difference between the interpolated and the original gather. The error is zero at the location of original traces and fairly random at the location of inserted traces.
Figure 15: Close-up comparison of the interpolated (right) and the original data (left).
used previously by Brown and Clapp (2000). The ground-roll noise and the reflection events have a significantly different frequency content, which might suggest separating them on the base of frequency alone. The result of frequency-based separation, shown in Figure 18 is, however, not ideal: part of the noise remains in the estimated signal after the separation. Changing the $\epsilon$ parameter in equation (23) could clean up the signal estimate, but it would also bring some of the signal into the subtracted noise.

A better strategy is to separate the events by using both the difference in frequency and the difference in slope. For that purpose, I adopted the following algorithm:

1. Use a frequency-based separation (or, alternatively, a simple low-pass filtering) to obtain an initial estimate of the ground-roll noise.

2. Select a window around the initial noise. The further separation will happen only in that window.

3. Estimate the noise dip from the initial noise estimate.

4. Estimate the signal dip in the selected data window as the complimentary dip component to the already known noise dip.

5. Use the signal and noise dips together with the signal and noise frequencies to perform the final separation. This is achieved by cascading single-dip plane-wave destruction filters with local 1-D three-coefficient PEFs aimed at destroying a particular frequency.

The separation result is shown in Figure 19. The separation goal has been fully achieved: the estimated ground-roll noise is free of the signal components, and the estimated signal is free of the noise.
CONCLUSIONS

Plane-wave destruction filters with an improved finite-difference design can be a valuable tool in processing multidimensional seismic data. On several examples, I showed their good performance in such problems as fault detection, missing data interpolation, and noise attenuation. Although only 2-D examples were demonstrated, it is straightforward to extend the method to 3-D applications by considering two orthogonal plane-wave slopes.

The similarities and differences between plane-wave destructors and $T$-$X$ prediction-error filters can be summarized as follows:

Similarities:

- Both types of filters operate in the original time-and-space domain of recorded data.
- Both filters aim to predict local plane-wave events in the data.
Figure 18: Signal and noise separation based on frequency. Top: estimated signal. Bottom: estimated noise.
Figure 19: Signal and noise separation based on both apparent dip and frequency in the considered receiver cable. Top: estimated signal. Bottom: estimated noise.
• In most problems, one filter type can be replaced by the other, and certain
techniques, such as Claerbout’s trace interpolation method, are common for
both approaches.

Differences:

• The design of plane-wave destructors is purely deterministic and follows the
plane-wave differential equation. The design of $T$-$X$ PEF has statistical roots
in the framework of the maximum-entropy spectral analysis (Burg, 1975). In
principle, $T$-$X$ PEF can characterize more complex signals than local plane
waves.

• In the case of PEF, we estimate filter coefficients. In the case of plane-wave
destructors, the estimated quantity is the local plane-wave slope. Several im-
portant distinctions follow from that difference:

  – The filter-estimation problem is linear. The slope estimation problem, in
the case of the improved filter design, is non-linear, but allows for an it-
erative linearization. In general, non-linearity is an undesirable feature
because of local minima and the dependence on initial conditions. How-
ever, we can sometimes use it creatively. For example, it helped to avoid
aliased dips in the trace interpolation example.

  – Non-stationarity is handled gracefully in the local slope estimation. No
local windows are required to produce a smoothly varying estimate of the
local slope. This is a much more difficult issue for PEFs because of the
largely under-determined problem.

  – Local slope has a clearly interpretable physical meaning, which allows for
easy quality control of the results. The coefficients of $T$-$X$ PEFs are much
more difficult to interpret.

• The efficiency of the two approaches is difficult to compare. Plane-wave destruc-
tors are generally more efficient to apply because of the small number of filter
coefficients. However, they may require more computation at the estimation
stage because of the non-linearity problem.

ACKNOWLEDGMENTS

This work was partially accomplished at the Stanford Exploration Project (SEP).

I would like to thank Jon Claerbout, Robert Clapp, Matthias Schwab, and other
SEP members for developing and maintaining the reproducible research technology,
which helped this research.

Suggestions from two anonymous reviewers helped to improve the paper.
REFERENCES

——–, 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project.
Harlan, W. S., J. F. Claerbout, and F. Rocca, 1984, Signal/noise separation and
**APPENDIX A**

**DETERMINING FILTER COEFFICIENTS BY TAYLOR EXPANSION**

This appendix details the derivation of equations (9) and (10). The main idea is to match the frequency responses of the approximate plane-wave filters to the response of the exact phase-shift operator at low frequencies.

The Taylor series expansion of the phase-shift operator $e^{i\omega \sigma}$ around the zero frequency $\omega = 0$ takes the form

$$e^{i\omega \sigma} \approx 1 + i \sigma \omega - \frac{\sigma^2 \omega^2}{2} - i \frac{\sigma^3 \omega^3}{6} + O(\omega^4) \tag{A-1}$$

The Taylor expansion of the six-point implicit finite-difference operator takes the form

$$\frac{B_3(Z_t)}{B_3(1/Z_t)} = \frac{b_{-1} Z_t^{-1} + b_0 + b_1 Z_t}{b_1 Z_t + b_0 + b_{-1} Z_t^{-1}} = \frac{b_{-1} e^{-i\omega} + b_0 + b_1 e^{i\omega} - b_{-1} e^{i\omega} + b_0 + b_{-1} e^{i\omega}}{b_1 e^{i\omega} + b_0 + b_{-1} e^{i\omega}} \approx 1 - 2i \frac{(b_{-1} - b_1) \omega}{b_0 + b_{-1} + b_1} - \frac{2 (b_{-1} - b_1)^2 \omega^2}{(b_0 + b_{-1} + b_1)^2} + i \frac{(b_{-1} - b_1) [b_0^2 - b_0 (b_{-1} + b_1) 4 (b_{-1}^2 - 4 b_{-1} b_1 + b_1^2)] \omega^3}{3 (b_0 + b_{-1} + b_1)^3} + O(\omega^4) \tag{A-2}$$

Matching the corresponding terms of expansions (A-1) and (A-2), we arrive at the system of nonlinear equations

$$\sigma = \frac{2 (b_1 - b_{-1})}{b_0 + b_{-1} + b_1} \tag{A-3}$$
$$\sigma^2 = \frac{4 (b_1 - b_{-1})^2}{(b_0 + b_{-1} + b_1)^2} \tag{A-4}$$
$$\sigma^3 = \frac{2 (b_1 - b_{-1}) [b_0^2 - b_0 (b_{-1} + b_1) 4 (b_{-1}^2 - 4 b_{-1} b_1 + b_1^2)] \omega^3}{(b_0 + b_{-1} + b_1)^3} \tag{A-5}$$

System (A-3-A-5) does not uniquely constrain the filter coefficients $b_{-1}$, $b_0$, and $b_1$ because equation (A-4) simply follows from (A-3) and because all the coefficients can be multiplied simultaneously by an arbitrary constant without affecting the ratios in equation (A-2). I chose an additional constraint in the form

$$B_3(1) = b_{-1} + b_0 + b_1 = 1 \tag{A-6}$$

which ensures that the filter $B_3(Z_t)$ does not alter the zero frequency component. System (A-3-A-5) with the additional constraint (A-6) resolves uniquely to the coefficients.
ficients of filter (9) in the main text:

\[
\begin{align*}
    b_{-1} &= \frac{(1 - \sigma)(2 - \sigma)}{12} ; \\
    b_0 &= \frac{(2 + \sigma)(2 - \sigma)}{6} ; \\
    b_1 &= \frac{(1 + \sigma)(2 + \sigma)}{12} .
\end{align*}
\]

(A-7) \hspace{1cm} (A-8) \hspace{1cm} (A-9)

The \( B_5 \) filter of equation (10) is constructed in a completely analogous way, using longer Taylor expansions to constrain the additional coefficients. Generalization to longer filters is straightforward.

The technique of this appendix aims at matching the filter responses at low frequencies. One might construct different filter families by employing other criteria for filter design (least squares fit, equiripple, etc.)
Inverse B-spline interpolation

Sergey Fomel

ABSTRACT

B-splines provide an accurate and efficient method for interpolating regularly spaced data. In this paper, I study the applicability of B-spline interpolation in the context of the inverse interpolation method for regularizing irregular data. Numerical tests show that, in comparison with lower-order linear interpolation, B-splines lead to a faster iterative conversion in under-determined problems and a more accurate result in over-determined problems. In addition, they provide a constructive method for creating discrete regularization operators from continuous differential equations.

INTRODUCTION

The problem of interpolating irregularly sampled data to regular grid (data regularization) can be recast as the inverse process with respect to interpolating regularly sampled data to irregular locations. Claerbout (1999) describes an iterative least-squares optimization approach to data regularization. The optimization is centered around two goals. The first goal is to minimize the power of the residual difference between the observed and predicted data. The second goal is to style the solution according to some predefined regularization criterion.

The ability of inverse interpolation to reach the data fitting goal depends on the accuracy of the forward interpolation operator. Forward interpolation is one of the classic problems in numerical analysis and has been studied extensively by generations of theoreticians and practitioners (Fomel, 1997b). The two simplest and most widely used methods are the nearest neighbor interpolation and linear interpolation. There are several approaches for constructing more accurate (albeit more expensive) linear forward interpolation operators: cubic convolution (Keys, 1981), local Lagrange, tapered sinc (Harlan, 1982), etc. (Wolberg, 1990) presents a detailed review of different conventional approaches.

Spline interpolation, based on representing the interpolated function by smooth piece-wise polynomials, has been in use for a long time (de Boor, 1978), but only recently Unser et al. (1993a,b) have discovered a way of implementing forward B-spline interpolation with an arbitrary order of accuracy in an efficient signal-processing fashion. The key idea is to implement the B-spline transform with recursive filtering. First, an efficient recursive filtering transforms regularly spaced data into spline
coefficients, then the spline coefficients are interpolated onto irregular locations. B-spline interpolants exhibit a superior performance for any given order of accuracy in comparison with other methods of similar efficiency [Thévenaz et al., 2000].

In this paper, I study the applicability of B-spline interpolation in the context of the inverse interpolation method. In the first section, I review the forward interpolation problem and confirm the observations of Thévenaz et al. (2000) about the superior performance of B-splines. The second section introduces a constructive method of creating discrete regularization operators from B-splines and helical filtering (Claerbout, 1998). The method performance is evaluated with a simple numerical test. In conclusion, I summarize the benefits of using B-splines for data regularization.

FORWARD INTERPOLATION

Forward interpolation plays only a supplementary role in this paper, but it has many applications of its own in the seismic processing practice. It is sufficient to mention such applications as trace resampling, NMO, Kirchoff and Stolt migrations, log-stretch, radial transform, etc. Two simple examples appear at the end of this section.

The general form of a linear forward interpolation operator is

$$f(x) = \sum_{n\in N} W(x,n) f(n),$$

where $n$ is a point on a given regular grid $N$, $x$ is a point in the continuum, $f(x)$ is the reconstructed continuous function, and $W(x,n)$ is a linear weight. Although in the discussion that follows, I refer to only the one-dimensional theory, a generalization to many dimensions is straightforward.

Nearest neighbor and beyond

The two simplest forms of the forward interpolation operators are the 1-point nearest neighbor interpolation with the weight

$$W(x,n) = \begin{cases} 1, & \text{for } n - 1/2 \leq x < n + 1/2 \\ 0, & \text{otherwise} \end{cases}$$

and the 2-point linear interpolation with the weight

$$W(x,n) = \begin{cases} 1 - |x-n|, & \text{for } n - 1 \leq x < n + 1 \\ 0, & \text{otherwise} \end{cases}$$

Because of their simplicity, the nearest neighbor and linear interpolation methods are very practical and easy to apply. Their accuracy is, however, limited and may be inadequate for interpolating high-frequency signals. The shapes of interpolants [2]...
Inverse interpolation

and (12) and their spectra are plotted in Figures 1 and 2. The spectra plots show that both interpolants act as low-pass filters, preventing the high-frequency energy from being correctly interpolated.

On the other side of the accuracy scale, there is the infinitely long sinc interpolant:

\[
W(x, n) = \frac{\sin[\pi(x - n)]}{\pi(x - n)}. \tag{4}
\]

According to the sampling theorem [Kotel’nikov 1933, Shannon 1949], equation (4) provides an optimal interpolation for any band-limited signal. In practice, it is not directly applicable because of a prohibitively expensive computation. The shape of the sinc function and its spectrum are shown in Figure 3. The spectrum is identically equal to one in the Nyquist frequency band.

Several approaches exist for extending the nearest neighbor and linear interpolation to more accurate (albeit more expensive) methods. One example is the 4-point cubic convolution suggested by Keys (1981). The cubic convolution interpolant is a local piece-wise cubic function, which approximates the ideal sinc equation (4). Another popular approach is to taper the ideal sinc function in a local window. For
example, one can use the Kaiser window \cite{Kaiser:1980}

\[
W(x, n) = \begin{cases} 
\sin[\pi(x - n)] \frac{I_0 \left( \alpha \sqrt{1 - \left(\frac{x-n}{N}\right)^2} \right)}{I_0(\alpha)} & \text{for } n - N < x < n + N \\
0, & \text{otherwise}
\end{cases}
\]

where \(I_0\) is the zero-order modified Bessel function of the first kind. The Kaiser-windowed sinc interpolant \cite{Kaiser:1980} has the adjustable parameter \(\alpha\), which controls the behavior of its spectrum. I have found empirically the value of \(\alpha = 4\) to provide a spectrum that deviates from 1 by no more than 1% in a relatively wide band.

I compare the accuracy of different forward interpolation methods on a one-dimensional signal shown in Figure 4. The ideal signal has an exponential amplitude decay and a quadratic frequency increase from the center towards the edges. It is sampled at a regular 50-point grid and interpolated to 500 regularly sampled locations. The interpolation result is compared with the ideal one. Observing Figures 5, 6, and 7, we can see the interpolation error steadily decreasing as we go subsequently from 1-point nearest neighbor to 2-point linear, 4-point cubic convolution, and 8-point windowed sinc interpolation. At the same time, the cost of interpolation grows proportionally to the interpolant length.

The differences among different methods are also clearly visible from the discrete spectra of the corresponding interpolants. The left plots in figures 8 and 9 show discrete interpolation responses: the function \(W(x, n)\) for a fixed value of \(x = 0.7\). The right plots compare the corresponding discrete spectra. We can see that the spectrum gets flatter and wider as the accuracy of the method increases.
Figure 4: One-dimensional test signal. Top: ideal. Bottom: sampled at 50 regularly spaced points. The bottom plot is the input in a forward interpolation test.

Figure 5: Interpolation error of the nearest neighbor interpolant (dashed line) compared to that of the linear interpolant (solid line).

Figure 6: Interpolation error of the linear interpolant (dashed line) compared to that of the cubic convolution interpolant (solid line).
Figure 7: Interpolation error of the cubic convolution interpolant (dashed line) compared to that of the 8-point windowed sinc interpolant (solid line).

Figure 8: Discrete interpolation responses of linear and cubic convolution interpolants (left) and their discrete spectra (right) for $x = 0.7$.

Figure 9: Discrete interpolation responses of cubic convolution and 8-point windowed sinc interpolants (left) and their discrete spectra (right) for $x = 0.7$. 

\texttt{bspl/chirp/cubkai}

\texttt{bspl/chirp/speclincub}

\texttt{bspl/chirp/speccubkai}
Interpolation and convolution

As I discussed in an earlier paper (Fomel, 1997b), a general approach for constructing the interpolant function $W(x, n)$ in equation (9) is to select an appropriate function basis for representing the function $f(x)$. The functional basis representation has the general form

$$f(x) = \sum_{k \in K} c_k \psi_k(x), \quad (6)$$

where $\psi_k(x)$ are basis function, and $c_k$ are the corresponding coefficients. Once an appropriate basis is selected, one can define the $W(x, n)$ function by means of the least squares method.

Unser et al. (1993a) noticed that the function basis idea has an especially simple implementation if the basis is convolutional and satisfies the equation

$$\psi_k(x) = \beta(x - k). \quad (7)$$

In other words, the basis is constructed by integer shifts of a single function $\beta(x)$. Substituting formula (7) into equation (6) yields

$$f(x) = \sum_{k \in K} c_k \beta(x - k). \quad (8)$$

Evaluating the function $f(x)$ in equation (15) at an integer value $n$, we obtain the equation

$$f(n) = \sum_{k \in K} c_k \beta(n - k), \quad (9)$$

which has the exact form of a discrete convolution. The basis function $\beta(x)$, evaluated at integer values, is digitally convolved with the vector of basis coefficients to produce the sampled values of the function $f(x)$. We can invert equation (9) to obtain the coefficients $c_k$ from $f(n)$ by inverse recursive filtering (deconvolution). In the case of a non-causal filter $\beta(n)$, an appropriate spectral factorization will be needed prior to applying the recursive filtering.

According to the convolutional basis idea, forward interpolation becomes a two-step procedure. The first step is the direct inversion of equation (9): the basis coefficients $c_k$ are found by deconvolving the sampled function $f(n)$ with the factorized filter $\beta(n)$. The second step reconstructs the continuous (or arbitrarily sampled) function $f(x)$ according to formula (15). The two steps could be combined into one, but usually it is more convenient to apply them separately. I show a schematic relationship among different variables in Figure 1.

B-splines

B-splines represent a particular example of a convolutional basis. Because of their compact support and other attractive numerical properties, B-splines are a good basis
choice for the forward interpolation problem and related signal processing problems (Unser 1999).

B-splines of the order 0 and 1 coincide with the nearest neighbor and linear interpolants (2) and (12) respectively. B-splines $\beta^n(x)$ of a higher order $n$ can be defined by a repetitive convolution of the zeroth-order spline $\beta^0(x)$ (the box function) with itself:

$$\beta^n(x) = \beta^0(x) * \cdots * \beta^0(x) \quad \text{times} \quad (n+1)$$

There is also the explicit expression

$$\beta^n(x) = \frac{1}{n!} \sum_{k=0}^{n+1} C_k^{n+1} (-1)^k (x + \frac{n+1}{2} - k)_+^n,$$

which can be proved by induction. Here $C_k^{n+1}$ are the binomial coefficients, and the function $x_+$ is defined as follows:

$$x_+ = \begin{cases} x, & \text{for } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

As follows from formula (11), the most commonly used cubic B-spline $\beta^3(x)$ has the expression

$$\beta^3(x) = \begin{cases} (4 - 6|x|^2 + 3|x|^3)/6, & \text{for } 1 > |x| \geq 0 \\ (2 - |x|^3)/6, & \text{for } 2 > |x| \geq 1 \\ 0, & \text{elsewhere} \end{cases} \quad (13)$$

The corresponding discrete filter $\beta^3(n)$ is a centered 3-point filter with coefficients 1/6, 2/3, and 1/6. According to the traditional method, a deconvolution with this filter is performed as a tridiagonal matrix inversion (de Boor 1978). One can accomplish it more efficiently by spectral factorization and recursive filtering (Unser et al. 1993a). The recursive filtering approach generalizes straightforwardly to B-splines of higher orders.

Both the support length and the smoothness of B-splines increase with the order. In the limit, B-splines converge to the Gaussian function. Figures [11] and [12] show the third- and seventh-order splines $\beta^3(x)$ and $\beta^7(x)$ and their continuous spectra.
Figure 11: Third-order B-spline $\beta^3(x)$ (left) and its spectrum (right).

Figure 12: Seventh-order B-spline $\beta^7(x)$ (left) and its spectrum (right).
It is important to realize the difference between B-splines and the corresponding interpolants \( W(x, n) \), which are sometimes called cardinal splines. An explicit computation of the cardinal splines is impractical, because they have infinitely long support. Typically, they are constructed implicitly by the two-step interpolation method, outlined in the previous subsection. The cardinal splines of orders 3 and 7 and their spectra are shown in Figures 13 and 14. As B-splines converge to the Gaussian function, the corresponding interpolants rapidly converge to the sinc function \( [4] \). A good convergence is achieved with the help of the infinitely long support, which results from recursive filtering at the first step of the interpolation procedure.

![Figure 13: Effective third-order B-spline interpolant (left) and its spectrum (right).](image1)

![Figure 14: Effective seventh-order B-spline interpolant (left) and its spectrum (right).](image2)

In practice, the recursive filtering step adds only marginally to the total interpolation cost. Therefore, an \( n \)-th order B-spline interpolation is comparable in cost with any other method with an \( (n + 1) \)-point interpolant. The comparison in accuracy usually turns out in favor of B-splines. Figures 15 and 16 compare interpolation errors of B-splines and other similar-cost methods on the example from Figure 4.

Similarly to Figures 8 and 9, we can also compare the discrete responses of B-spline interpolation with those of other methods. The right plots in Figures 17 and 18 show that the discrete spectra of the effective B-spline interpolants are genuinely flat at low frequencies and wider than those of the competitive methods. Although the B-spline
Figure 15: Interpolation error of the cubic convolution interpolant (dashed line) compared to that of the third-order B-spline (solid line).

Figure 16: Interpolation error of the 8-point windowed sinc interpolant (dashed line) compared to that of the seventh-order B-spline (solid line).
responses are infinitely long because of the recursive filtering step, they exhibit a fast amplitude decay.

Figure 17: Discrete interpolation responses of cubic convolution and third-order B-spline interpolants (left) and their discrete spectra (right) for $x = 0.7$.

Figure 18: Discrete interpolation responses of 8-point windowed sinc and seventh-order B-spline interpolants (left) and their discrete spectra (right) for $x = 0.7$.

2-D example

For completeness, I include a 2-D forward interpolation example. Figure 19 shows a 2-D analog of function in Figure 4 and its coarsely-sampled version.

Figure 20 compares the errors of the 2-D nearest neighbor and 2-D linear (bi-linear) interpolation. Switching to bi-linear interpolation shows a significant improvement, but the error level is still relatively high. As shown in Figures 21 and 22, B-spline interpolation again outperforms other methods with comparable cost complexity. In all cases, I constructed 2-D interpolants by orthogonal splitting. Although the splitting method reduces computational overhead, the main cost factor is the total interpolant size, which squares when going from 1-D to 2-D.
Figure 19: Two-dimensional test function (left) and its coarsely sampled version (right). [bspl/chirp2/ chirp2]

Figure 20: 2-D Interpolation errors of nearest neighbor interpolation (left) and linear interpolation (right). Top graphs show 1-D slices through the center of the image. [bspl/chirp2/ plcbinlin]
Figure 21: 2-D Interpolation errors of cubic convolution interpolation (left) and third-order B-spline interpolation (right). Top graphs show 1-D slices through the center of the image.

bspl/chirp2/ plccubspl

Figure 22: 2-D Interpolation errors of 8-point windowed sinc interpolation (left) and seventh-order B-spline interpolation (right). Top graphs show 1-D slices through the center of the images.

bspl/chirp2/ plckaispl
Beyond B-splines

It is not too difficult to construct a convolutional basis with better interpolation properties than those of B-splines, for example by sacrificing their smoothness. The following piece-wise cubic function has a lower smoothness than $\beta^3(x)$ in equation (13) but slightly better interpolation behavior:

$$
\mu^3(x) = \begin{cases} 
(10 - 13|x|^2 + 6|x|^3) / 16, & \text{for } 1 > |x| \geq 0 \\
(2 - |x|)^2(5 - 2|x|)/16, & \text{for } 2 > |x| \geq 1 \\
0, & \text{elsewhere}
\end{cases}
$$

(14)

Figures 23 and 24 compare the test interpolation errors and discrete responses of methods based on the B-spline function $\beta^3(x)$ and the lower smoothness function $\mu^3(x)$. The latter method has a slight but visible performance advantage and a slightly wider discrete spectrum.

Figure 23: Interpolation error of the third-order B-spline interpolant (dashed line) compared to that of the lower smoothness spline interpolant (solid line).

[bspl/chirp/spl4mom4]

Figure 24: Discrete interpolation responses of third-order B-spline and lower smoothness spline interpolants (left) and their discrete spectra (right) for $x = 0.7$.

[bspl/chirp/specspl4mom4]

Blu et al. (1998) have developed a general approach for constructing non-smooth piece-wise functions with optimal interpolation properties. However, the gain in accu-
racy is often negligible in practice. In the rest of this paper, I use the classic B-spline method.

**Seismic applications of forward interpolation**

For completeness, I conclude this section with two simple examples of forward interpolation in seismic data processing. Figure 25 shows a 3-D impulse response of Stolt migration (Stolt, 1978), computed by using 2-point linear interpolation and 8-point B-spline interpolation. As noted by Ronen (1982) and Harlan (1982), inaccurate interpolation may lead to spurious artifact events in Stolt-migrated image. Indeed, we see several artifacts for the image with linear interpolation (the left plots in Figure 25). The artifacts are removed by a more accurate interpolation method (the right plots in Figure 25).

![Figure 25: Stolt migration impulse response. Left: using linear interpolation. Right: using seventh-order B-spline interpolation. Migration artifacts are removed by a more accurate forward interpolation method.](bspl/stolt/stolt)

Another simple example is radial trace transform (Ottolini, 1982) Figure 26 shows a land shot gather contaminated by nearly radial ground-roll. As discussed by Claerbout (1983), Henley (1999), and Brown and Claerbout (2000), one can effectively eliminate ground-roll noise by applying radial trace transform, followed by high-pass filtering and the inverse radial transform. Figure 27 shows the result of the forward radial transform of the shot gather in Figure 26 in the radial band of the ground-roll noise and the transform error after going back to the original domain. Comparing
the results of using linear and third-order B-spline interpolation, we see once again that the transform artifacts are removed with a more accurate interpolation scheme.

Figure 26: Ground-roll-contaminated shot gather used in a radial transform test [bspl/radial/ radialdat]

**INVERSE INTERPOLATION AND DATA REGULARIZATION**

In the notation of Claerbout (1999), inverse interpolation amounts to a least-squares solution of the system

\[ Lm \approx d ; \]  
\[ \epsilon Am \approx 0 , \]  

where \( d \) is a vector of known data \( f(x_i) \) at irregular locations \( x_i \), \( m \) is a vector of unknown function values \( f(n) \) at a regular grid \( n \), \( L \) is a linear interpolation operator of the general form \( (\text{9}) \), \( A \) is an appropriate regularization (model styling) operator, and \( \epsilon \) is a scaling parameter. In the case of B-spline interpolation, the forward interpolation operator \( L \) becomes a cascade of two operators: recursive deconvolution \( B^{-1} \), which converts the model vector \( m \) to the vector of spline coefficients \( c \), and a spline basis construction operator \( F \). System \((\text{15}-\text{13})\) transforms to

\[ FB^{-1}m \approx d ; \]  
\[ \epsilon Am \approx 0 . \]  

We can rewrite \((\text{17}-\text{18})\) in the form that involves only spline coefficients:

\[ Wc \approx d ; \]  
\[ \epsilon ABc \approx 0 . \]  

After we find a solution of system \((\text{19}-\text{20})\), the model \( m \) will be reconstructed by the simple convolution

\[ m = Bc . \]
Figure 27: Radial trace transform results. Top: radial trace domain. Bottom: residual error after the inverse transform. The error should be zero in a radial band from 0 to 0.65 km/s radial velocity. Left: using linear interpolation. Right: using third-order B-spline interpolation.
This approach resembles a more general method of model preconditioning (Fomel, 1997a).

The inconvenient part of system (19-20) is the complex regularization operator $AB$. Is it possible to avoid the cascade of $B$ and $A$ and to construct a regularization operator directly applicable to the spline coefficients $c$? In the following subsection, I develop a method for constructing spline regularization operators from differential equations.

Spline regularization

In many cases, the regularization (styling) condition originates in a continuous differential operator. For example, one can think of the gradient or Laplacian operator for regularizing smooth functions (Fomel, 2000b), plane-wave destructor for regularizing local plane waves (Fomel, 2000a), or the offset continuation equation for regularizing seismic reflection data (Fomel, 2000c).

Let us denote the continuous regularization operator by $D$. Regularization implies seeking a function $f(x)$ such that the least-squares norm of $D[f(x)]$ is minimum. Using the usual expression for the least-squares norm of continuous functions and substituting the basis decomposition (15), we obtain the expression

$$\|D[f(x)]\| = \int (D[f(x)])^2 \, dx = \int \left( \sum_{k \in K} c_k D[\beta(x-k)] \right)^2 \, dx.$$  \hspace{1cm} (22)

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 (22) with respect to the basis coefficients $c_k$. This leads to the system of linear equations

$$\sum_{k \in K} c_k \int D[\beta(x-k)] D[\beta(x-j)] \, dx = \sum_{k \in K} c_k d_{j-k} = 0,$$  \hspace{1cm} (23)

where

$$d_j = \int D[\beta(x)] D[\beta(x-j)] \, dx.$$  \hspace{1cm} (24)

Equation (23) is clearly a discrete convolution of the spline coefficients $c_k$ with the filter $d_j$ defined in equation (24). To transform the system (23) to a regularization condition of the form

$$Dc \approx 0,$$  \hspace{1cm} (25)

we need to treat the digital filter $d_j$ as an autocorrelation and find its minimum-phase factor. Equation (25) replaces equation (20) in the inverse interpolation problem setting.

We have found a constructive way of creating B-spline regularization operators from continuous differential equations.
A simple regularization example is shown in Figure 28. The continuous operator $D$ in this case comes from the theoretical plane-wave differential equation. I constructed the auto-correlation filter $d_j$ according to formula (24) and factorized it with the efficient Wilson-Burg method on a helix (Sava et al., 1998). The figure shows three plane waves constructed from three distant spikes by applying an inverse recursive filtering with two different plane-wave regularizers. The left plot corresponds to using first-order B-splines (equivalent to linear interpolation). This type of regularizer is identical to Clapp’s steering filters (Clapp et al., 1997) and suffers from numerical dispersion effects. The right plot was obtained with third-order splines. Most of the dispersion is suppressed by using a more accurate interpolation.

![Spline waves B-1 (linear)](image1.png) ![Spline waves B-3 (cubic)](image2.png)

Figure 28: B-spline regularization. Three plane waves constructed by 2-D recursive filtering with the B-spline plane-wave regularizer. Left: using first-order B-splines (linear interpolation). Right: using third-order B-splines.

### Test example

Now that we have all the problem pieces together, we can test the performance gain in the inverse interpolation problem (19)-(25) from the application of B-splines.

For a simple 1-D test, I chose the function shown in Figure 4 but sampled at irregular locations. To create two different regimes for the inverse interpolation problem, I chose 50 and 500 random locations. The two sets of points were interpolated 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 29 and 30 show the input data for the two test after normalized binning to the selected regular bins.
Figure 29: 50 random points binned to 500 regular grid points. The random data are used for testing inverse interpolation in an under-determined situation. [bspl/bintest/bin500]

Figure 30: 500 random points binned to 50 regular grid points. The random data are used for testing inverse interpolation in an over-determined situation. [bspl/bintest/bin50]
I solved system (19)-(25) by the iterative conjugate-gradient method, utilizing a recursive filter preconditioning (Fomel, 1997a) for faster convergence. The regularization operator $D$ was constructed by using the method of the previous subsection with the tension-spline differential equation (Smith and Wessel, 1990; Fomel, 2000b) and the tension parameter of 0.01.

The least-squares differences between the true and the estimated model are plotted in Figures 31 and 32. 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 at earlier iterations. 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 31: Model convergence in the under-determined case. Dashed line: using linear interpolation. Solid line: using third-order B-spline.

application to 3-D seismic data regularization

In this subsection, I demonstrate an application of B-spline inverse interpolation for regularizing three-dimensional seismic reflection data. The dataset of this example comes from the North Sea and was used before for testing AMO (Biondi et al., 1998) and common-azimuth migration (Biondi, 1996). Figure 33 shows the midpoint
geometry and the corresponding bin fold for a selected range of offsets and azimuths. The goal of data regularization is to create a regular data cube at the specified bins from the irregular input data, preprocessed by NMO. As typical of marine acquisition, the fold distribution is fairly regular but has occasional gaps caused by the cable feathering effect.

The data cube after normalized binning (inverse nearest neighbor interpolation) is shown in Figure 34. Binning works reasonably well in the areas of large fold but fails to fill the zero fold gaps and has an overall limited accuracy.

Inverse interpolation using bi-linear interpolants significantly improves the result (Figure 35), and inverse B-spline interpolation improves the accuracy even further (Figure 36). In both cases, I regularized the data in constant time slices, using recursive filter preconditioning with plane-wave destructor filters analogous to those in Figure 28. The plane wave slope was estimated from the binned data with the method of Fomel (2000a). The inverse interpolation results preserve both flat reflection events in the data and steeply-dipping diffractions. When data regularization is used as a preprocessing step for common-azimuth migration (Biondi and Palacharla, 1996), preserving diffractions is important for correct imaging of sharp edges in the subsurface structure.

**CONCLUSIONS**

I have reviewed the B-spline forward interpolation method and confirmed the observation of Thévenaz et al. (2000) about its superior performance in comparison with other methods of similar cost. Whenever an accurate forward interpolation scheme is desired, B-splines can be an extremely valuable tool. B-spline forward interpolation involves two steps. The first step is recursive filtering, which results in a set of spline coefficients. The second step is a linear spline interpolation operator.

Analyzing the role of B-spline interpolation in data regularization, I have in-
Figure 33: Midpoint geometry (left) and fold distribution (right) for the 3-D data test [bspl/sei3d/cmpfold]

introduced a method of constructing B-spline discrete regularization operators from continuous differential equations.

Simple numerical experiments with B-spline inverse interpolation show that the main advantage of using a more accurate interpolation scheme occurs in an over-determined setting, where B-splines lead to a more accurate model estimates. In an under-determined setting, the B-spline inverse interpolation scheme converges faster at early iterations, but the total computational gain may be insignificant.

I have shown on a simple real data example that inverse B-spline interpolation can be used as an accurate method of data regularization for processing 3-D seismic reflection data.

ACKNOWLEDGMENTS

A short conversation with Dave Hale brought me to a better understanding of different forward interpolation methods.

The 3-D North Sea dataset was released to SEP by Conoco and its partners, BP and Mobil.
Figure 34: 3-D data after normalized binning

Nearest Neighbor

Figure 34: 3-D data after normalized binning
Figure 35: 3-D data after inverse interpolation with bi-linear interpolants

[Image of 3-D data with labels: B-spline 2, Time (s), In-line midpoint, Cross-line]
Figure 36: 3-D data after inverse interpolation with third-order B-spline interpolants

\texttt{bspl/sei3d/int4}
REFERENCES

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

The two-stage missing data interpolation approach of Claerbout (1998) (henceforth, the GEE approach) has been applied with great success (Fomel et al., 1997; Clapp et al., 1998; Crawley, 2000) in the past. The main strength of the approach lies in the ability of the prediction error filter (PEF) to find multiple, hidden correlation in the known data, and then, via regularization, to impose the same correlation (covariance) onto the unknown model. Unfortunately, the GEE approach may break down in the face of very sparsely-distributed data, as the number of valid regression equations in the PEF estimation step may drop to zero. In this case, the most common approach is to simply retreat to regularizing with an isotropic differential filter (e.g., Laplacian), which leads to a minimum-energy solution and implicitly assumes an isotropic model covariance.

A pressing goal of many SEP researchers is to find a way of estimating a PEF from sparse data. Although new ideas are certainly required to solve this interesting problem, Claerbout (2000) proposes that a standard, simple test case first be constructed, and suggests using a known model with vanishing Gaussian curvature. In this paper, we present the following, simpler test case, which we feel makes for a better first step.

- **Model:** Deconvolve a 2-D field of random numbers with a simple dip filter, leading to a “plane-wave” model.
- **Filter:** The ideal interpolation filter is simply the dip filter used to create the model.
- **Data:** Subsample the known model randomly and so sparsely as to make conventional PEF estimation impossible.

We use the aforementioned dip filter to regularize a least squares estimation of the missing model points and show that this filter is ideal, in the sense that the model residual is relatively small. Interestingly, we found that the characteristics of the true model and interpolation result depended strongly on the accuracy (dip spectrum localization) of the dip filter. We chose the 8-point truncated sinc filter presented by Fomel (2000). We discuss briefly the motivation for this choice.

METHODOLOGY

Claerbout (1998) presents a two-stage methodology for missing data interpolation. In the first stage of the so-called GEE approach, a prediction error filter (PEF) is estimated from known data. In the second stage, the PEF is used in a least squares interpolation scheme to regularize the undetermined (missing) model points. Crawley (2000) extends the two-stage procedure to use nonstationary PEF’s.

The first stage (PEF estimation) of the two-stage procedure consists of convolving the unknown filter coefficients with the known data, and adjusting the coefficients such
that the residual is minimized. Conceptually, in the process of convolution, a filter template is slid past every point in the data domain. The GEE approach adheres to the following convention: unless every point in the filter template overlies known data, the regression equation for that output point is ignored, and will not contribute to the PEF estimation.

Unfortunately, when the known data is very sparsely distributed, all the regression equations may depend on missing data, making PEF estimation impossible. The motivation of this paper is not to present a new methodology for estimating a PEF from sparse data, but instead to create a very simple test case which fulfills the following criteria:

1. The known data is distributed so sparsely as to render the traditional GEE two-stage approach ineffective.

2. The underlying model is conceptually simple and stationary.

3. The ideal PEF for the underlying model is obtainable by common sense.

The Test Case

Claerbout (2000) proposes a test case for which the Gaussian curvature of the model vanishes. In this paper, we present an even simpler test case. Given a 2-D random field, we deconvolve with a known dip (or steering) filter to obtain a “plane wave” model, as shown in Figure 1. To simulate collected “data”, we sampled the model of Figure 1 at about 60 points randomly, and about two-thirds of the way down one trace in the center. The results are shown in Figure 1.

Figure 1: True model - plane waves dipping at 22.5°.
Digression: Accuracy of Dip Filters

Given a pure plane wave section, i.e., a wavefield where all events have linear moveout, designing a discrete multichannel filter to annihilate events with a given dip seems a trivial task. In fact, it is quite a tricky task; an exercise in interpolation. For many applications, accuracy considerations make the choice of interpolation algorithm critical. *Accuracy* here means localization of the filter’s dip spectrum — ideally the filter should annihilate only the desired dip, or a narrow range of dips.

The problem is illustrated in Figure 1. Given a plane wave with dip $p$, we must set the filter coefficients $a_i$ to best annihilate the plane wave. Achieving good dip spectrum localization implies a filter with many coefficients, by the uncertainly principle ([Bracewell, 1986](#)). If computational cost was not an issue, the best choice would be a sinc function with as many coefficients as time samples. Realistically, however,
a compromise must be found between pure sinc and simple linear interpolation. The reader is referred to (Fomel, 2000) paper, which discusses these issues much more thoroughly. The model of Figure 1 was computed using an 8-point tapered sinc function. Figure 4 compares the result of using, for the same task, dip filters computed via four different interpolation schemes: 8-point tapered sinc, 6-point local Lagrange, 4-point cubic convolution, and simple 2-point linear interpolation. As expected, we see that the more accurate interpolation schemes lead to increased spatial coherency in the model panel. Clapp (2000) has been successful in using as few as 3 coefficients in steering filters for regularizing tomography problems, so we see that the needed amount of steering filter accuracy is a problem-dependent parameter.

Figure 4: Interpolation schemes compared. Deconvolution of random image with labeled steering filters.

INTERPOLATION RESULTS

The plane wave model of Figure 1 dips at 22.5°, so we can easily design a filter to annihilate it. Using the GEE approach for interpolating missing data (Claerbout, 2000)
we interpolate the data of Figure 1 using the 8-point tapered sinc steering filter discussed above. The results are shown in Figure 5. We see that the interpolation is quite good in the center region, where the filter can “see” one or more known data points, as evidenced by a nearly uncorrelated model residual. In the corners, the result is imperfect in regions in which no known data points exist along diagonal tracks. In order to suppress helix wraparoun and other edge effects, we apply zero-padding around the edges of the study region.

Figure 5: Clockwise from upper left: Known data; Interpolation regularized with 8-point tapered sinc steering filter; Difference between known model and interpolated result; known model.  

sparse2/test/correct
CONCLUSIONS

We presented a 2-D test case for sparse data interpolation and give a good PEF with which to do it. The test case renders the traditional GEE two-stage interpolation scheme inapplicable. [Claerbout (2000)] suggests a nonlinear iteration, where filter and model are taken as unknown, but the best solution is still a subject of discussion among many SEP researchers. Regardless of the chosen interpolation strategy, the “correct” PEF and model are both known in this test case, so it should prove a useful starting point.

REFERENCES

Crawley, S., 2000, Seismic trace interpolation with nonstationary prediction-error filters, in SEP-104: Stanford Exploration Project.
Multiple suppression using prediction-error filter

Yalei Sun

ABSTRACT

I present an approach to multiple suppression, that is based on the moveout between primary and multiple events in the CMP gather. After normal moveout correction, primary events will be horizontal, whereas multiple events will not be. For each NMOed CMP gather, I reorder the offset in random order. Ideally, this process has little influence on the primaries, but it destroys the shape of the multiples. In other words, after randomization of the offset order, the multiples appear as random noise. This “man-made” random noise can be removed using prediction-error filter (PEF). The randomization of the offset order can be regarded as a random process, so we can apply it to the CMP gather many times and get many different samples. All the samples can be arranged into a 3-D cube, which is further divided into many small subcubes. A 3-D PEF can then be estimated from each subcube and re-applied to it to remove the multiple energy. After that, all the samples are averaged back into one CMP gather, which is supposed to be free of multiple events. In order to improve the efficiency of the algorithm of estimating the PEF for each subcube, except for the first subcube which starts with a zero-valued initial guess, all the subsequent subcubes take the last estimated PEF as an initial guess. Therefore, the iteration count can be reduced to one step for all the subsequent subcubes with little loss of accuracy. Three examples demonstrate the performance of this new approach, especially in removing the near-offset multiples.

INTRODUCTION

One of the essential differences between multiples and primaries is moveout. On the basis of this difference, a velocity-stacking multiple suppression approach was developed by Lumley et al. (1994). In the velocity-stacking domain, the multiple and primary energy are separated and a masking function is applied to the data in the velocity-stacking domain to remove the multiple energy. Then the data can be inverse-transformed back to the time domain with only primary energy left.

Since the velocity-stacking operator is time-variant, it does not have the Toeplitz structure in the frequency domain. Therefore, given an operator and its adjoint, the inverse transform has to be formulated as an $L_p$ norm optimization problem, which makes this approach fairly expensive. In addition, this approach cannot handle the
near-offset multiple events very well. Because the near-offset multiple event is parallel to the primary event, it is very difficult to separate it from primary energy in the velocity stacking domain.

In this paper, I propose a new approach, which also makes use of the moveout difference between primary and multiple events. Instead of velocity-stacking transform, normal-moveout correction is applied to the CMP gather. Ideally, the hyperbolic primary events will be flattened after NMO correction, whereas the multiple events will not be. There is a residual moveout for the multiple events. I then randomize the order of offsets. This process will have little influence over the primary, but the shape of the multiple event will be destroyed by the residual moveout. In other words, after this randomization, the primary energy is still coherent and the multiple energy will look like random noise. Therefore, the multiple energy can be removed using a prediction-error filter (PEF).

The randomization of the offset order can be regarded as a random process, which can be applied to a single CMP gather many times to produce an ensemble of samples. The ensemble of samples forms a 3-D cube, which can be further decomposed into many small 3-D subcubes. A 3-D PEF is estimated from each subcube and then convolved with the cube to remove the multiple energy. After that, all the samples are averaged back into one CMP gather, which is ideally free of multiple. Here, the estimation of the PEF for each subcube is posed as an inversion problem.

In order to improve the efficiency of the algorithm, I set the initial guess of the PEF for one subcube to be the PEF of its preceding subcube. In other words, except for the first subcube, which starts with a zero-valued initial guess, all the subsequent subcubes take last estimated PEF as initial guess. Therefore, the iteration steps can be reduced to one step for all the subsequent subcubes with little loss of accuracy. Since we are mainly interested in preserving the horizontal events, there are some requirements regarding the shape of the PEF.

Two synthetic and one real example are presented to show that this multiple suppression approach can remove the multiples from near to far offset. Especially in the near offset, its performance is very promising. Since this approach is based on PEF, other kinds of random noise will be removed as well as multiples. Therefore, the signal-to-noise ratio will be increased.

**REORDERING THE OFFSET ORDER AND SIMULATING A RANDOM PROCESS**

Starting with a NMOed CMP gather, we can rearrange the offset order in a random manner. Figure 1 illustrates a synthetic NMOed CMP gather which contains one primary and one multiple event. After offset randomization, the primary event remains horizontal and the multiple event turns into "random noise".

If we apply the offset randomization multiple times, all the different samples of
this random process can generate a 3-D data cube. As shown in Figure 2, the primary energy is coherent along the cube, whereas the multiple part is incoherent. Therefore, not only can we estimate a 2-D PEF in the CMP gather, but also estimate a 3-D PEF in the data cube. It is very natural to expect that the 3-D result will be better than the 2-D, since the 3-D approach exploits the difference between the primary and multiple more thoroughly.

**T-X-Y Domain Prediction Filtering**

The theory of T-X-Y prediction filtering can be found in Claerbout (1994). Abma (1995) applies the T-X-Y prediction-error filter to signal/noise separation. The algorithm used in this paper is the same as the one in Abma’s thesis. Here, I show three examples of the application of T-X-Y prediction filtering to two synthetic CMP gathers and one real CMP gather.

**A Simple Synthetic CMP Gather**

The synthetic CMP gather consists of a horizontal event which represents the primary and a curved event representing the multiple. After offset randomization, the CMP gather is shown in Figure 1. Figure 3 illustrates the result of PEF-based multiple suppression in 2-D case. In this simple example, although there are strong aliasing
Figure 2: A NMOed CMP gather is transformed into a 3-D data cube by conducting the offset randomization for multiple times. The 2-D prediction-error filter separates the horizontal and curved events very well.

A more complicated synthetic CMP gather

This synthetic gather is generated by the Thompson-Haskell method, which has been used as an example in Lumley et al. (1994). The gather includes both primary and multiple events. Velocity stacking approach can remove the far offset multiples successfully, but it is not so successful in near offset, since the multiple event is nearly horizontal in the near offset. I show that PEF based approach can handle the near-offset multiples effectively.

Figure 4 illustrates that most of the multiple energy has been well-separated from primary energy, from near-offset to far-offset. I also extract three traces from the near offset and compare the difference between the two approaches, as shown in Figure 5. It is very clear that the PEF scheme has removed most of the multiple energy in the near offset. The signal-to-noise ratio of PEF based approach higher than the other two figures.
Figure 3: A simple synthetic CMP gather composed of one primary event and one multiple event. This simple example demonstrates the applicability of the PEF based multiple suppression scheme.

Figure 4: A synthetic CMP gather generated by Haskell-Thompson method. After applying the 3-D PEF, both multiple events and random noise have been removed out of the input CMP gather.
Figure 5: Comparison of the PEF scheme and velocity-stacking scheme in the near offset. Three traces are extracted from the near offset. The left one is the original input, the middle one is the PEF result, and the right one is the result of velocity stacking approach. It is very easy to identify the primary events in the middle one.

A real CMP gather

This real CMP gather is extracted from Mobil AVO dataset. The multiple energy is very strong in this dataset. It is not very difficult to locate several high amplitude horizontal events in the original events, as shown in Figure 6. After convolving with the 3-D PEF, these events have been preserved. Most of the multiple energy has been removed. As for the near-offset traces, the signal-to-noise ratio of the PEF approach is also higher than that of the velocity-stacking approach.

Obviously this result is not as convincing as last one. The reason is actually the assumption of the PEF based approach. The PEF based multiple suppression scheme has an assumption over the amplitude of the primary events, that is, the change of amplitude versus offset should be as small as possible. The first example is generated following this assumption strictly. Since this real data has strong AVO effect, the PEF based approach will remove this effect by forcing the amplitude to be invariant along offset. Therefore, some of the primary energy will be lost for this reason. Even for this, the improvement of signal-to-noise ratio in the near offset is still obvious in the near offset, as shown in Figure 7.
Figure 6: A real CMP gather. This is from Mobil’s AVO dataset. The high amplitude events in the input section are primary energy. The middle section is the result of the 3-D PEF scheme. Although most of the multiple events have been removed, the AVO effect is also damaged by the filtering. This shows that the PEF approach has an assumption of constant amplitude along the offset.

Figure 7: Comparison of the PEF scheme and velocity-stacking scheme in the near offset. Although the AVO effect has been damaged by the filtering, the signal-to-noise ratio is improved in the near offset.
CONCLUSION AND DISCUSSION

Two synthetic and one real CMP gathers show that this multiple suppression scheme can remove both near-offset and far-offset multiples. This multiple suppression approach has three features. The first one is offset randomization, which destroys the shape of multiple events. The idea of converting coherent noise like multiples into random noise is very novel, it may find applications in other fields. The second is the assumption of primary events being horizontal after normal moveout correction. This assumption makes it difficult in dealing with nonhyperbolic moveout. The third is that this approach cannot handle the amplitude variation along offset. Therefore, it will lose some primary energy caused by AVO or NMO stretch. Obviously, the second and third features will limit the application range of this approach.

I used a trick to improve the efficiency in using T-X-Y PEF. That is, taking the preceding subcube’s PEF as an initial guess when estimating a new PEF. In my application, this trick can make the algorithm at least ten times faster than the algorithm without using this trick. The error is less than one percent.

ACKNOWLEDGEMENTS

I thank Dr. Ray Abma of Shell and Dr. Jiuying Guo of Tsinghua University for stimulating discussions during the 66th SEG Denver convention.

REFERENCES

Multidimensional recursive filter preconditioning in geophysical estimation problems

Sergey Fomel and Jon F. Claerbout

ABSTRACT

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

INTRODUCTION

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

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

---

1 Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas at Austin, University Station, Box X, Austin, TX 78713-8972, USA
2 Stanford Exploration Project, Mitchell Bldg., Department of Geophysics, Stanford University, Stanford, California 94305-2215, USA
1e-mail: yalei@sep.stanford.edu
second formulation is intrinsically related to the concept of model preconditioning \cite{Vandecar_1994}. We illustrate the general regularization theory with simple synthetic examples.

Though the final results of the model-space and data-space regularization are theoretically identical, the behavior of iterative gradient-based methods, such as the method of conjugate gradients, is different for the two cases. The obvious difference is in the case where the number of model parameters is significantly larger than the number of data measurements. In this case, the dimensions of the inverted matrix in the case of the data-space regularization are smaller than those of the model-space matrix, and the convergence of the iterative conjugate-gradient iteration requires correspondingly smaller number of iterations. But even in the case where the number of model and data parameters are comparable, preconditioning changes the iteration behavior. This follows from the fact that the objective function gradients with respect to the model parameters are different in the two cases. Since iteration to the exact solution is rarely achieved in large-scale geophysical applications, the results of iterative optimization may turn out quite differently. \cite{Harlan_1995} points out that the goals of the model-space regularization conflict with each other: the first one emphasizes “details” in the model, while the second one tries to smooth them out. He describes the advantage of preconditioning as follows:

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

In this paper, we illustrate the two approaches on synthetic and real data examples from simple environmental data sets. All examples show that when we solve the optimization problem iteratively and take the output only after a limited number of iterations, it is preferable to use the preconditioning approach. When the regularization operator is convolution with a filter, a natural choice for preconditioning is inverse filtering (recursive deconvolution). We show how to extend the method of preconditioning by recursive filtering to multiple dimensions. The extension is based on modifying the boundary conditions with the helix transform \cite{Claerbout_1998}.
**REVIEW OF REGULARIZATION IN ESTIMATION PROBLEMS**

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

\[ \mathbf{d} - \mathbf{Lm} \approx \mathbf{0}, \]  

(1)

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

**Model-space regularization**

Model-space regularization implies adding equations to system (17) to obtain a fully constrained (well-posed) inverse problem. The additional equations take the form

\[ \epsilon \mathbf{Dm} \approx \mathbf{0}, \]  

(2)

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

The full system of equations (17-2) can be written in a short notation as

\[ \mathbf{G}_m \mathbf{m} = \begin{bmatrix} \mathbf{L} \\ \epsilon \mathbf{D} \end{bmatrix} \mathbf{m} \approx \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} = \hat{\mathbf{d}}, \]  

(3)

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

\[ \hat{\mathbf{d}} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}, \]  

(4)

and \( \mathbf{G}_m \) is a column operator:

\[ \mathbf{G}_m = \begin{bmatrix} \mathbf{L} \\ \epsilon \mathbf{D} \end{bmatrix}. \]  

(5)

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

\[ \hat{\mathbf{r}} = \hat{\mathbf{d}} - \mathbf{G}_m \mathbf{m} = \begin{bmatrix} \mathbf{d} - \mathbf{Lm} \\ -\epsilon \mathbf{Dm} \end{bmatrix}. \]  

(6)
The formal solution of the regularized optimization problem has the known form (Parker, 1994)

\[
<m> = (L^T L + \epsilon^2 D^T D)^{-1} L^T d ,
\]

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

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

**Data-space regularization (model preconditioning)**

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

\[
m = Pp ,
\]

where \( P \) is a preconditioning operator. The residual vector \( r \) for the data-fitting goal (17) can be defined by the relationship

\[
\epsilon r = d - Lm = d - LPp ,
\]

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

\[
\hat{p} = \begin{bmatrix} p \\ r \end{bmatrix} .
\]

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

\[
\begin{bmatrix} LP & \epsilon I \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} = G_d \hat{p} = d ,
\]

where \( G_d \) is a *row* operator:

\[
G_d = \begin{bmatrix} LP & \epsilon I \end{bmatrix} ,
\]

and \( I \) represents the data-space identity operator.

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

\[
<\hat{p}> = \frac{<d>}{<r>} = G_d^T (G_d G_d^T)^{-1} d = \frac{P^T L^T (LPP^T L^T + \epsilon^2 I)^{-1} d}{\epsilon (LPP^T L^T + \epsilon^2 I)^{-1} d} .
\]

(13)
Applying equation (21), we obtain the corresponding estimate \(<m>\) for the initial model \(m\), as follows:

\[
<m> = PP^T L^T (LP^T L^T + \epsilon^2 I)^{-1} d.
\]  

(14)

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

\[
C = PP^T = (DT D)^{-1},
\]

(15)

where we need to assume the self-adjoint operator \(D^T D\) to be invertible.

To prove the equivalence, consider the operator

\[
G = L^T L C L^T + \epsilon^2 L^T,
\]

(16)

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

\[
G = L^T (LC L^T + \epsilon^2 I) = (L^T L + \epsilon^2 C^{-1}) C L^T,
\]

(17)

or, in another form,

\[
CL^T (LC L^T + \epsilon^2 I)^{-1} \equiv (L^T L + \epsilon^2 C^{-1})^{-1} L^T.
\]

(18)

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

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

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

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

**ONE-DIMENSIONAL SYNTHETIC EXAMPLES**

In the first example, the input data \(d\) were randomly subsampled (with decreasing density) from a sinusoid (Figure 1). The forward operator \(L\) in this case is linear interpolation. In other words, we seek a regularly sampled model \(m\) on 200 grid points that could predict the data with a forward linear interpolation. Sparse irregular distribution of the input data makes the regularization necessary. We applied convolution with the simple \((1, -1)\) difference filter as the operator \(D\) that forces model continuity (the first-order spline). An appropriate preconditioner \(P\) in this...
Table 1: Comparison between model-space and data-space regularization

<table>
<thead>
<tr>
<th>Effective Model</th>
<th><em>Model-space</em></th>
<th><em>Data-space</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Data</td>
<td>[ \hat{d} = \begin{bmatrix} d \ 0 \end{bmatrix} ]</td>
<td>[ d ]</td>
</tr>
<tr>
<td>Effective Operator</td>
<td>[ G_m = \begin{bmatrix} L \ \epsilon D \end{bmatrix} ]</td>
<td>[ G_d = \begin{bmatrix} LP \ \epsilon I \end{bmatrix} ]</td>
</tr>
<tr>
<td>Optimization Problem</td>
<td>minimize ( \hat{\mathbf{r}}^T \hat{\mathbf{r}} ), where ( \hat{\mathbf{r}} = \hat{d} - G_m \mathbf{m} )</td>
<td>minimize ( \hat{\mathbf{p}}^T \hat{\mathbf{p}} ) under the constraint ( G_d \hat{\mathbf{p}} = d )</td>
</tr>
<tr>
<td>Formal Estimate for ( \mathbf{m} )</td>
<td>( (L^T L + \epsilon^2 C^{-1}) L^T d ), where ( C^{-1} = D^T D )</td>
<td>( C L^T (L CL^T + \epsilon^2 I)^{-1} d ), where ( C = \mathbf{P} \mathbf{P}^T ).</td>
</tr>
</tbody>
</table>

Figure 1: The input data (right) are irregularly spaced samples of a sinusoid (left).
case is the inverse of $D$, which amounts to recursive causal integration (Claerbout, 1999). Figures 2 and 3 show the results of inverse interpolation after exhaustive 300 iterations of the conjugate-direction method. The results from the model-space and data-space regularization look similar except for the boundary conditions outside the data range. As a result of using the causal integration for preconditioning, the rightmost part of the model in the data-space case stays at a constant level instead of decreasing to zero. If we specifically wanted a zero-value boundary condition, we could easily implement it by adding a zero-value data point at the boundary.

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

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

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

Figure 5: The top figure is the exact solution found in 250 iterations. The middle is with data-space regularization after 5 iterations. The bottom is with model-space regularization after 5 iterations.
Changing the preconditioning operator changes the regularization result. Figure 6 shows the result of data-space regularization after a triangle smoother is applied as the model preconditioner. Triangle smoother is a filter with the $Z$-transform $(1-z^N)(1-z^{-N})/(1-z)(1-z^{-1})$ (Claerbout, 1992). We chose the filter length $N = 6$.

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

If, instead of looking for a smooth interpolation, we want to limit the number of frequency components, then the optimal choice for the model-space regularization operator $D$ is a prediction-error filter (PEF). To obtain a mono-frequency output, we can use a three-point PEF, which has the $Z$-transform representation $D(Z) = 1 + a_1Z + a_2Z^2$. In this case, the corresponding preconditioner $P$ can be the three-point recursive filter $P(Z) = 1/(1 + a_1Z + a_2Z^2)$. To test this idea, we estimated the PEF $D(Z)$ from the output of inverse linear interpolation (Figure 3), and ran the data-space regularized estimation again, substituting the recursive filter $P(Z) = 1/D(Z)$ in place of the causal integration. We repeated this two-step procedure three times to get a better estimate for the PEF. The result, shown in Figure 7, exhibits the desired mono-frequency output. One can accommodate more frequency components in the model using a longer filter.

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

Regularization after binning: missing data interpolation

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

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

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

Mathematically, this principle can be expressed by the simple equation

\[ Dm \approx 0 , \]  

where \( m \) is the model vector and \( D \) is the specified filter. Equation (19) is completely equivalent to equation (2). The approximate equality sign means that equation (19) is solved by minimizing the squared norm (the power) of its left side. Additionally, the known data values must be preserved in the optimization scheme. Introducing the mask operator \( K \), which is a diagonal matrix with zeros at the missing data locations and ones elsewhere, we can rewrite equation (19) in the extended form

\[ D(I - K)m \approx -DKm = -Dm_k , \]  

where \( I \) is the identity operator and \( m_k \) represents the known portion of the data. It is important to note that equation (20) corresponds to the regularized linear system

\[
\begin{cases}
Km = m_k , \\
\epsilon Dm \approx 0
\end{cases}
\]  

in the limit of the scaling coefficient \( \epsilon \) approaching zero. System (13) is equivalent to system (21) with the masking operator \( K \) playing the role of the forward interpolation operator \( L \). Setting \( \epsilon \) to zero implies associating more weight on the first equation in (13) and using the second equation only to constrain the null space of the solution. Applying the general theory of data-space regularization from the previous section, we can immediately transform system (13) to the equation

\[ KPp \approx m_k , \]  

where \( P \) is a preconditioning operator and \( p \) is the preconditioning variable, connected with \( m \) by the simple relationship (21). According to equations (14) and (7) from the previous section, equations (9) and (20) have exactly the same solutions if condition (15) is satisfied. If \( D \) is represented by a discrete convolution, the natural choice for \( P \) is the corresponding deconvolution (inverse recursive filtering) operator:

\[ P = D^{-1} . \]  

(23)
We illustrate the missing data problem with a simple 1-D synthetic data test taken from [Claerbout (1999)]. Figure 8 shows the interpolation results of the unpreconditioned technique with two different filters $D$. For comparison with the preconditioned scheme, we changed the boundary convolution conditions from internal to truncated transient convolution. As in the previous example, the system was solved with a conjugate-gradient iterative optimization.

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

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

Preconditioned interpolation behaves differently (Figure 9). At the early iterations, the model is simple. As the iteration proceeds, new details are added into the model. After a surprisingly small number of iterations, the output closely resembles the final output. The final output of interpolation with preconditioning by recursive deconvolution is exactly the same as that of the original method.

The next section extends the idea of preconditioning by inverse recursive filtering to multiple dimensions.
Figure 9: Interpolation with preconditioning. Left plot: the top shows the input data; the middle, the result of interpolation; the bottom, the filter impulse response. The right plot shows the convergence process for the first four iterations.
MULTIDIMENSIONAL RECURSIVE FILTER PRECONDITIONING

Claerbout (1998) proposed a helix transform for mapping multidimensional convolution operators to their one-dimensional equivalents. This transform proves the feasibility of multidimensional deconvolution, an issue that has been in question for more than 25 years (Claerbout [1976]). By mapping discrete convolution operators to one-dimensional space, the inverse filtering problem can be conveniently recast in terms of recursive filtering, a well-known part of the digital filtering theory.

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

The helix filtering idea is schematically illustrated in Figure 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 skipped in a helical convolution algorithm. The gain is not in the convolution itself, but in the ability to perform recursive inverse filtering (deconvolution) in multiple dimensions. A multi-dimensional filter is mapped to its 1-D analog by imposing helical boundary conditions on the appropriate axes. After that, inverse filtering is applied recursively in a one-dimensional manner. Neglecting parallelization and indexing issues, the cost of inverse filtering is equivalent to the cost of convolution. It is proportional to the data size and to the number of non-zero filter coefficients.
An example of two-dimensional recursive filtering is shown in Figure 11. The left plot contains two spikes and two filter impulse responses with different polarity. After deconvolution with the given filter, the filter responses turn into spikes, and the initial spikes turn into long-tailed inverse impulse responses (right plot in Figure 11). Helical wrap-around, visible on the top and bottom boundaries, indicates the direction of the helix. Claerbout (1999) presents more examples and discusses all the issues of multidimensional helical deconvolution in detail.

**MULTIDIMENSIONAL EXAMPLES**

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

We can obtain a significantly better image (Figure 13) by replacing the Laplacian filter with a two-dimensional prediction-error filter estimated from the input data. The result in the left plot of Figure 13 was obtained after 200 conjugate-gradient iterations. If we stop after 20 iterations, the output (the right plot in Figure 13) shows only a small deviation from the input data. Large areas of the image remain unfilled. At each iteration, the interpolation process progresses only to the length of
Figure 12: On the left, the SeaBeam data: the depth of the ocean under ship tracks; on the right, an interpolation with the Laplacian filter.

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

Inverting the PEF convolution with the help of the helix transform, we can now apply the inverse filtering operator to precondition the interpolation problem. As expected, the result after 200 iterations (the left plot in Figure 14) is similar to the result of the corresponding unpreconditioned (model-space) interpolation. However, the output after just 20 iterations (the right plot in Figure 14) is already fairly close to the solution.

For a more practical test, we chose the North Sea seismic reflection dataset, pre-
Figure 14: SeaBeam interpolation with the inverse prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations.

Previously used for testing azimuth moveout and common-azimuth migration (Biondi et al., 1998; Biondi, 1997). Figure 15 shows the highly irregular midpoint geometry for a selected in-line and cross-line offset bin in the data. The data irregularity is also evident in the bin fold map, shown in Figure 16. The goal of data regularization is to create a regular data cube at the specified bins from the irregular input data, which have been preprocessed by normal moveout without stacking.

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

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

For efficiency, we perform regularization on individual time slices. Figure 18 shows...
Figure 16: Map of the fold distribution for the 3-D data test.

Figure 17: 3-D data after normalized binning.
the result of regularization using bi-linear interpolation and smoothing preconditioning (data-space regularization) with the minimum-phase Laplacian filter (Fomel et al., 2002). The empty bins are filled in a consistent manner but the data quality is distorted because simple smoothing fails to characterize the complicated data structure. Instead of continuous events, we see smoothed blobs in the time slices. The events in the in-line and cross-line sections are also not clearly pronounced.

![Smoothing Preconditioning](image)

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

We can use the smoothing regularization result to estimate the local dips in the data, design invertible local plane-wave destruction filters (Fomel, 2001), and repeat the regularization process. Inverse interpolation with plane-wave data-space regularization is shown in Figure 19. The result is noticeably improved: the continuous reflection events become clearly visible in the time slices. Despite the irregularities in the input data, the regularization result preserves both flat reflection events and steeply-dipping diffractions. Preserving diffractions is important for correct imaging of sharp edges in the subsurface structure (Biondi et al., 1998).
For simplicity, we assumed only a single local dip component in the data. This assumption degrades the result in the areas of multiple conflicting dips, such as the intersections of plane reflections and hyperbolic diffractions in Figure 19. One could improve the image by considering multiple local dips. Fomel (2002) describes an alternative offset-continuation approach, which uses a physical connection between neighboring offsets instead of assuming local continuity in the midpoint domain.

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

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

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

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

We suggest a constructive method for preconditioning multidimensional estimation problems using the helix transform. Applying inverse filtering operators constructed this way, we observe a significant (order of magnitude) speed-up in the optimization convergence. Since inverse recursive filtering takes almost the same time as forward convolution, the acceleration translates straightforwardly into computational time savings.

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

ACKNOWLEDGMENTS

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

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

REFERENCES


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


Exploring three-dimensional implicit wavefield extrapolation with the helix transform

Sergey Fomel and Jon F. Claerbout

ABSTRACT
Implicit extrapolation is an efficient and unconditionally stable method of wavefield continuation. Unfortunately, implicit wave extrapolation in three dimensions requires an expensive solution of a large system of linear equations. However, by mapping the computational domain into one dimension via the helix transform, we show that the matrix inversion problem can be recast in terms of an efficient recursive filtering. Apart from the boundary conditions, the solution is exact in the case of constant coefficients (that is, a laterally homogeneous velocity.) We illustrate this fact with an example of three-dimensional velocity continuation and discuss possible ways of attacking the problem of lateral variations.

INTRODUCTION
Implicit finite-difference wavefield extrapolation played an exceptionally important role in the early development of seismic migration methods. Using limited-degree approximations to the one-way wave equation, implicit schemes have provided efficient and unconditionally stable numerical wave extrapolation operators (Godfrey et al., 1979; Claerbout, 1985). Unfortunately, the advantages of implicit methods were lost with the development of three-dimensional seismic exploration. While the cost of 2-D implicit extrapolation is linearly proportional to the mesh size, the same approach, applied in the 3-D case, leads to a nonlinear computational complexity. Primarily for this reason, implicit extrapolators were replaced in practice by explicit ones, capable of maintaining linear complexity in all dimensions. A number of computational tricks (Hale, 1991b) allow the commonly used explicit schemes to behave stably in practical cases. However, their stability is not unconditional and may break in unusual situations (Etgen, 1994).

In this paper, we present an approach to three-dimensional extrapolation, based on the helix transform of multidimensional filters to one dimension (Claerbout, 1997b). The traditional approach involves an inversion of a banded matrix (tridiagonal in the 2-D case and blocked-tridiagonal in the 3-D case). With the help of the helix transform, we can recast this problem in terms of inverse recursive filtering. The coefficients of two-dimensional filters on a helix are obtained by one-dimensional spectral factorization methods. As a result, the complexity of three-dimensional implicit

1e-mail: sergey@sep.stanford.edu, jon@sep.stanford.edu
extrapolation is reduced to a linear function of the computational mesh size. This approach doesn’t provide an exact solution in the presence of lateral velocity variations. Nevertheless, it can be used for preconditioning iterative methods, such as those described by Nichols (1991). In this paper, we demonstrate the feasibility of 3-D implicit extrapolation on the example of laterally invariant velocity continuation and, in the final part, discuss possible strategies for solving the problem of lateral variations.

The main application of finite-difference wave extrapolation is post-stack depth migration. An application of similar methods for prestack common-shot migration is constrained by the limited aperture of commonly used seismic acquisition patterns. Recently developed acquisition methods, such as the vertical cable technique (Krail, 1993), open up new possibilities for 3-D wave extrapolation applications. An alternative approach is common-azimuth migration (Biondi and Palacharla, 1994; Biondi, 1996). Other interesting applications include finite-difference data extrapolation in offset (Fomel, 1995), migration velocity (Fomel, 1996a), and anisotropy (Alkhalifah and Fomel, 1997).

IMPLICIT VERSUS EXPLICIT EXTRAPOLATION

The difference between implicit and explicit extrapolation is best understood through an example. Following Claerbout (1985), let us consider, for instance, the diffusion (heat conduction) equation of the form

$$\frac{\partial T}{\partial t} = a(x) \frac{\partial^2 T}{\partial x^2}.$$  \(1\)

Here $t$ denotes time, $x$ is the space coordinate, $T(x, t)$ is the temperature, and $a$ is the heat conductivity coefficient. Equation (1) forms a well-posed boundary-value problem if supplied with the initial condition

$$T|_{t=0} = T_0(x)$$  \(2\)

and the appropriate boundary conditions. Our task is to build a digital filter, which transforms a gridded temperature $T$ from one time level to another.

It helps to note that when the conductivity coefficient $a$ is constant and the space domain of the problem is infinite (or periodic) in $x$, the problem can be solved in the wavenumber domain. Indeed, after the Fourier transform over the variable $x$, equation (1) transforms to the ordinary differential equation

$$\frac{d \hat{T}}{dt} = -a k^2 \hat{T} ,$$  \(3\)

which has the explicit analytical solution

$$\hat{T}(k, t) = \hat{T}_0(k) e^{-a k^2 t} ,$$  \(4\)
where \( \hat{T} \) denotes the Fourier transform of \( T \), and \( k \) stands for the wavenumber. Therefore, the desired filter in the wavenumber domain has the form

\[
H(k) = e^{-ak^2},
\]  
(5)

where for simplicity the coefficient \( a \) is normalized for the time step \( \Delta t \) equal to 1.

Returning now to the time-and-space domain, we can approach the filter construction problem by approximating the space-domain response of filter (5) in terms of the differential operators \( \frac{\partial^2}{\partial x^2} = -k^2 \), which can be approximated by finite differences. An explicit approach would amount to constructing a series expansion of the form

\[
H_{\text{ex}}(k) \approx a_0 + a_1k^2 + a_2k^4 + \ldots,
\]

and selecting the coefficients \( a_j \) to approximate equation (5). For example, the three-term Taylor series expansion around the zero wavenumber yields

\[
H_{\text{ex}}(k) = 1 - ak^2 + \frac{a^2 k^4}{2}.
\]

(7)

The error of approximation (7) as a function of \( k \) for two different values of \( a \) is shown in the left plot of Figure 1.

![Figure 1: Errors of second-order explicit (a) and implicit (b) approximations for the heat extrapolation.](image.png)

An implicit approach also approximates the ideal filter (5), but with a rational approximation of the form

\[
H_{\text{im}}(k) \approx \frac{b_0 + b_1k^2 + b_2k^4 + \ldots}{1 + c_1k^2 + c_2k^4 + \ldots}.
\]

(8)

One way of selecting the coefficients \( b_i \) and \( c_i \) is to apply an appropriate Padé approximation (Baker and Graves-Morris 1981). For example the \([2/2]\) Padé approximation

\[\text{[Notes: Additional comments or equations if needed.]}\]
is

\[ H_{im}(k) = \frac{1 - \frac{a}{2} k^2}{1 + \frac{a}{2} k^2}. \]  

(9)

This approximation corresponds to the famous Crank-Nicolson implicit method \cite{Crank1947}. The error of approximation (9) as a function of \( k \) for different values of \( a \) is shown in the right plot of Figure 1. Not only is it significantly smaller than the error of the same-order explicit approximation, but it also has a negative sign. It means that the high-frequency numerical noise gets suppressed rather than amplified. In practice, this property translates into a stable numerical extrapolation.

The second derivative operator \(-k^2\) can be approximated in practice by a digital filter. The most commonly used filter has the \( Z \)-transform

\[ D_2(Z) = -Z^{-1} + 2 - Z, \]

and the Fourier transform

\[ D_2(k) = e^{-ik} - 2 + e^{-ik} = 2(\cos k - 1) = -4 \sin^2 \frac{k}{2}. \]

(10)

Formula (10) approximates \(-k^2\) well only for small wavenumbers \( k \). As shown in Appendix A, the implicit scheme allows the accuracy of the second-derivative filter to be significantly improved by a variation of the “1/6-th trick” \cite{Claerbout1985}. The final form of the implicit extrapolation filter is

\[ H_{im}(k) = \frac{1 + \frac{a + \beta}{2} D_2(k)}{1 - \frac{a - \beta}{2} D_2(k)}, \]

(11)

where \( \beta \) is a numerical constant, found in Appendix A.

![Figure 2: Heat extrapolation with explicit and implicit finite-different schemes. Explicit extrapolation appears stable for \( a = 2/3 \) (left plot) and unstable for \( a = 4/3 \) (middle plot). Implicit interpolation is stable even for larger values of \( a \) (right plot).](findif/heat/heat)

A numerical 1-D example is shown in Figure 2. The initial temperature distribution is given by a step function. The discontinuity at the step gets smoothed with
time by the heat diffusion. The left plot shows the result of an explicit extrapolation with \( a = 2/3 \), which appears stable. The middle plot is an explicit extrapolation with \( a = 4/3 \), which shows a terribly unstable behavior: the high-frequency numerical noise is amplified and dominates the solution. The right plot shows a stable (though not perfectly accurate) extrapolation with the implicit scheme for the larger value of \( a = 2 \).

The difference in stability between explicit and implicit schemes is even more pronounced in the case of wave extrapolation. For example, let us consider the ideal depth extrapolation filter in the form of the phase-shift operator (Gazdag [1978], Claerbout [1985])

\[
W(k) = e^{i\sqrt{a^2-k^2}} ,
\]

where \( a = \omega/v \), \( \omega \) is the time frequency, and \( v \) is the seismic velocity (which may vary spatially); we assume for simplicity that both the depth step \( \Delta z \) and the space sampling \( \Delta x \) are normalized to 1. A simple implicit approximation to filter (12) is

\[
W_{im}(k) = e^{ia} \frac{1 - 4a^2 + iak^2}{1 - 4a^2 - iak^2} = e^{i\phi} ,
\]

where \( \phi = a - 2 \arctan \frac{ak^2}{4a^2-1} \). We can see that approximation (13) is again a pure phase shift operator, only with a slightly different phase. For that reason, the operator is unconditionally stable for all values of \( a \): the total wave energy from one depth level to another is preserved. Operator (12) corresponds to the Crank-Nicolson scheme for the 45-degree one-way wave equation (Claerbout [1985]). Its phase error as a function of the dip angle \( \theta = \arcsin \frac{k}{a} \) for different values of \( a \) is shown in Figure 1.

Figure 3: The phase error of the implicit depth extrapolation with the Crank-Nicolson method.

The unconditional stability property is not achievable with the explicit approach, though it is possible to increase the stability of explicit operators by using relatively long filters (Holberg [1988], Hale [1991b]).

**SPECTRAL FACTORIZATION AND THREE-DIMENSIONAL EXTRAPOLATION**

In this section, we continue our review of extrapolation methods to reveal the principal difficulties of three-dimensional extrapolation. We then describe a new, helix-
transform approach to this old and fascinating problem.

**Inverse filter factorization**

The conventional way of applying implicit finite-difference schemes reduces to solving a system of linear equations with a sparse matrix. For example, to apply the scheme of equation (11), we can put the filter denominator on the other side of the extrapolation equation, writing it as

\[
\begin{pmatrix}
I - \frac{a - \beta}{2} D_2
\end{pmatrix} T_{t+1} = \begin{pmatrix}
I + \frac{a + \beta}{2} D_2
\end{pmatrix} T_t ,
\]

(14)

where \( I \) is the identity matrix, \( D \) is the convolution matrix for filter (10), and \( T_t \) is the vector of temperature distribution at time level \( t \). In the case of two-dimensional extrapolation, the matrix on the left side of equation (14) takes the tridiagonal form

\[
A = (I - c D_2) = \\
\begin{bmatrix}
1 + 2c_1 & -c_1 & 0 & \cdots & 0 \\
-c_2 & 1 + 2c_2 & -c_2 & 0 & \cdots \\
0 & -c_3 & \cdots & \cdots & \cdots \\
\cdots & 0 & \cdots & \cdots & -c_{n-1} \\
0 & \cdots & \cdots & -c_n & 1 + 2c_n
\end{bmatrix},
\]

(15)

where \( c = \frac{a - \beta}{2} \), and where, for simplicity, we assume zero-slope boundary conditions. Like any positive-definite tridiagonal matrix, matrix \( A \) can be inverted recursively by an LU decomposition into two bidiagonal matrices. The cost of inversion is directly proportional to the number of vector components. The same conclusion holds for the case of depth extrapolation [equation (13)] with the substitution \( c = \frac{\beta + ia}{1 - 4a} \).

In the case of a laterally constant coefficient \( a \), we can take a different point of view on the tridiagonal matrix inversion. In this case, the matrix \( A_2 \) represents a convolution with a symmetric three-point filter \( 1 - c D_2(k) \). The LU decomposition of such a matrix is precisely equivalent to filter factorization into the product of a causal minimum-phase filter with its adjoint. This conclusion can be confirmed by the easily verified equality

\[
1 + c(Z^{-1} - 2 + Z) = \frac{(1 + b)^2}{4} \left(1 + \frac{1-b}{1+b} Z\right) \left(1 + \frac{1-b}{1+b} Z^{-1}\right),
\]

(16)

where \( b = \sqrt{1 + 4c} \). The inverse of the causal minimum-phase filter \( 1 + \frac{1-b}{1+b} Z \) is a recursive inverse filter. Correspondingly, the inverse of its adjoint pair, \( 1 + \frac{1-b}{1+b} Z^{-1} \), is the same inverse filtering, performed in the adjoint mode (backwards in space). In the next subsection, we show how this approach can be carried into three dimensions by applying the helix transform.
Helix and multidimensional deconvolution

The major obstacle of applying an implicit extrapolation in three dimensions is that the inverted matrix is no longer tridiagonal. If we approximate the second derivative (Laplacian) on the 2-D plane with the commonly used five-point filter $Z_{x}^{-1} + Z_{y}^{-1} - 4 + Z_{x} + Z_{y}$, then the matrix on the left side of equation (14), under the usual mapping of vectors from a two-dimensional mesh to one dimension, takes the infamous blocked-tridiagonal form (Birkhoff, 1971)

$$\tilde{A} = (I - c D_{2}) = \begin{bmatrix} A_{1} & -c_{1} I & 0 & \cdots & 0 \\ -c_{2} I & A_{2} & -c_{2} I & 0 & \cdots \\ 0 & -c_{3} I & \cdots & \cdots & \cdots \\ \cdots & 0 & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & -c_{n-1} I & A_{n} \end{bmatrix}. \quad (17)$$

Inspecting this form more closely, we see that the main diagonal of $\tilde{A}$, as well as the two offset diagonals formed by the scaled identity matrices, remains continuous, while the second top and bottom diagonals are broken. Therefore, even for constant $c$, the inverted matrix does not have a simple convolutional structure, and the cost of its inversion grows nonlinearly with the number of grid points.

A helix transform, recently proposed by one of the authors (Claerbout, 1997a), sheds new light on this old problem. Let us assume that the extrapolation filter is applied by sliding it along the $x$ direction in the $\{x, y\}$ plane. The diagonal discontinuities in matrix $\tilde{A}$ occur exactly in the places where the forward leg of the filter slides outside the computational domain. Let’s imagine a situation, where the leg of the filter that went to the end of the $x$ column, would immediately appear at the beginning of the next column. This situation defines a different mapping from two computational dimensions to the one dimension of linear algebra. The mapping can be understood as the helix transform, illustrated in Figure ?? and explained in detail by Claerbout (1997a). According to this transform, we replace the original two-dimensional filter with a long one-dimensional filter. The new filter is partially filled with zero values (corresponding to the back side of the helix), which can be safely ignored in the convolutional computation.

This is exactly the helix transform that is required to make all the diagonals of matrix $\tilde{A}$ continuous. In the case of laterally invariant coefficients, the matrix becomes strictly Toeplitz (having constant values along the diagonals) and represents a one-dimensional convolution on the helix surface. Moreover, this simplified matrix structure applies equally well to larger second-derivative filters (such as those described in Appendix B), with the obvious increase of the number of Toeplitz diagonals. Inverting matrix $\tilde{A}$ becomes once again a simple inverse filtering problem. To decompose the 2-D filter into a pair consisting of a causal minimum-phase filter and its adjoint, we can apply spectral factorization methods from the 1-D filtering theory (Claerbout 1976, 1992), for example, Kolmogorov’s highly efficient method.
Figure 4: 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.

(Kolmogorov, 1939). Thus, in the case of a laterally invariant implicit extrapolation, matrix inversion reduces to a simple and efficient recursive filtering, which we need to run twice: first in the forward mode, and second in the adjoint mode.

Figure 5 shows the result of applying the helix transform to an implicit heat extrapolation of a two-dimensional temperature distribution. The unconditional stability properties are nicely preserved, which can be verified by examining the plot of changes in the average temperature (Figure 6).

In principle, we could also treat the case of a laterally invariant coefficient with the help of the Fourier transform. Under what circumstances does the helix approach have an advantage over Fourier methods? One possible situation corresponds to a very large input data size with a relatively small extrapolation filter. In this case, the $O(N\log N)$ cost of the fast Fourier transform is comparable with the $O(N_fN)$ cost of the space-domain deconvolution (where $N$ corresponds to the data size, and $N_f$ is the filter size). Another situation is when the boundary conditions of the problem have an essential lateral variation. The latter case may occur in applications of velocity continuation, which we discuss in the next section. Later in this paper, we return to the discussion of problems associated with lateral variations.
Figure 5: Heat extrapolation in two dimensions, computed by an implicit scheme with helix recursive filtering. The left plot shows the input temperature distributions; the two other plots, the extrapolation result at different time steps. The coefficient $a$ is 2. findif/heat/heat3d

Figure 6: Demonstration of the stability of implicit extrapolation. The solid curve shows the normalized mean temperature, which remains nearly constant throughout the extrapolation time. The dashed curve shows the normalized maximum value, which exhibits the expected Gaussian shape. findif/heat/heat-mean
THREE-DIMENSIONAL IMPLICIT VELOCITY CONTINUATION

Velocity continuation is a process of navigating in the migration velocity space, applicable for time migration, residual migration, and migration velocity analysis (Fomel, 1996a). In the zero-offset (post-stack) case, the velocity continuation process is described by the simple partial differential equation (Claerbout, 1986; Fomel, 1994)

\[
\frac{\partial^2 P}{\partial v \partial t} + vt \left( \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right) = 0 ,
\]

where \( t \) is the vertical time coordinate of the migrated image, \( x \) and \( y \) are spatial (midpoint) coordinates, and \( v \) is the migration velocity. Slightly different versions of two-dimensional implicit extrapolation with equation (18) have been described by Li (1986) and (Fomel, 1996a).

Figure 7: Impulse responses of the velocity continuation operator, computed by an implicit, unconditionally stable extrapolation via the helix transform. The left plot corresponds to continuation towards higher velocities (migration mode); the right plot, smaller velocities (modeling mode).

The helix approach has allowed us to modify the old code for three dimensions. Figure 7 shows impulse responses of an implicit helix-based three-dimensional velocity continuation.

Figure 9 illustrates the velocity continuation process on the Qdome synthetic model (Claerbout, 1997b), shown in Figure 8. Continuation backward in velocity corresponds to the “modeling” mode, while forward continuation corresponds to the “migration” mode. It is possible to balance the amplitudes of the two processes so that the finite-difference velocity continuation behaves as a unitary operator (Fomel, 1996a,b).
Figure 8: Qdome synthetic model, used for testing the 3-D velocity continuation program.

Figure 9: Modeling (left) and migration (right) with the Qdome synthetic model, obtained by running the 3-D velocity continuation backward and forward in velocity.
DEPTH EXTRAPOLATION AND THE V(X) CHALLENGE

Can the constant-velocity result help us achieve the challenging goal of a stable implicit depth extrapolation through media with lateral velocity variations?

The first idea that comes to mind is to replace the space-invariant helix filters with a precomputed set of spatially varying filters, which reflect local changes in the velocity fields. This approach would merely reproduce the conventional practice of explicit depth extrapolators, popularized by Holberg (1988) and Hale (1991b). However, it hides the danger of losing the property of unconditional stability, which is obviously the major asset of implicit extrapolators.

Another route, partially explored by Nichols (1991), is to implement the matrix inversion in the three-dimensional implicit scheme by an iterative method. In this case, the helix inversion may serve as a powerful preconditioner, providing an immediate answer in constant velocity layers and speeding up the convergence in the case of velocity variations. To see why this might be true, one can write the variable-coefficient matrix $\tilde{A}$ in the form

$$\tilde{A} = B + D,$$

where matrix $B$ corresponds to some constant average velocity, and $D$ is the matrix of velocity perturbations. The system of linear equations that we need to solve is then

$$(B + D) m = d,$$

where $m$ is the vector of extrapolated wavefield, and $d$ is an appropriate righthand side. The helix transform provides us with the operator $B^{-1}$, which we can use to precondition system (20). Introducing the change of variables

$$m = B^{-1}x,$$

we can transform the original system (20) to the form

$$d = (B + D) B^{-1} x = x + D B^{-1} x.$$

When the velocity perturbation is small, even the simple iteration

$$x_0 = d;$$

$$x_k = d - D B^{-1} x_{k+1}$$

will converge rapidly to the desired solution. This interesting possibility needs thorough testing.

The third untested possibility (Papanicolaou, personal communication) is to implement a clever patching in the velocity domain, applying a constant-velocity filter locally inside each patch. Recently developed fast wavelet transform techniques (Vetterli and Kovacevic, 1995), in particular the local cosine transform, provide a formal framework for that approach.

---

3In the linear-algebraic sense, this means that the spectral radius of operator $D B^{-1}$ is strictly less than one.
CONCLUSIONS

The feasibility of multidimensional deconvolution, proven by the helix transform, allows us to revisit the problem of implicit wavefield extrapolation in three dimensions. The attraction of implicit finite-difference methods lies in their unconditional stability, a property invaluable for practical applications.

We have shown that at least in the constant coefficient case (that is, laterally invariant velocity), it is possible to implement an extremely efficient implicit extrapolation by a recursive inverse filtering in the helix-transformed computational model. Unfortunately, the case of lateral velocity variations still presents a difficult problem that may not have an exact solution. We are currently exploring different roads to that goal.

ACKNOWLEDGMENTS

We thank Biondo Biondi for useful discussions and for sharing his three-dimensional expertise.

REFERENCES

Birkhoff, G., 1971, The numerical solution of elliptic equations: SIAM.
———, 1985, Imaging the Earth’s Interior: Blackwell Scientific Publications.
———, 1997b, Geophysical exploration mapping: Environmental soundings image enhancement: Stanford Exploration Project.


——–, 1996a, Migration and velocity analysis by velocity continuation, in SEP-92: Stanford Exploration Project, 159–188.


APPENDIX A

THE 1/6-TH TRICK

Given the filter $D_2(k)$, defined in formula (10), we can construct an accurate approximation to the second derivative operator $-k^2$ by considering a filter ratio (another
Padé-type approximation) of the form

\[-k^2 \approx \frac{D_2(k)}{1 + \beta D_2(k)}, \quad (A-1)\]

where \(\beta\) is an adjustable constant \(^{(1)}\)\textsuperscript{Claerbout, 1985}. The actual Padé coefficient is \(\beta = 1/12\). As pointed out by Francis Muir, the value of \(\beta = 1/4 - 1/\pi^2 \approx 1/6.726\) gives an exact fit at the Nyquist frequency \(k = \pi\). Fitting the derivative operator in the \(L_1\) norm yields the value of \(\beta \approx 1/8.13\). All these approximations are shown in Figure [A-1](#).

Figure A-1: The second-derivative operator in the wavenumber domain and its approximations.

**APPENDIX B**

**CONSTRUCTING AN “ISOTROPIC” LAPLACIAN OPERATOR**

The problem of approximating the Laplacian operator in two dimensions not only inherits the inaccuracies of the one-dimensional finite-difference approximations, but also raises the issue of azimuthal asymmetry. For example, the usual five-point filter

\[F_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (B-1)\]

exhibits a clear difference between the grid directions and the directions at a 45-degree angle to the grid. To overcome this unpleasant anisotropy, we can consider a slightly larger filter of the form

\[F_9 = \begin{bmatrix} \alpha & \gamma & \alpha \\ \gamma & -4(\alpha + \gamma) & \gamma \\ \alpha & \gamma & \alpha \end{bmatrix} \quad (B-2)\]

where the constants \(\alpha\) and \(\gamma\) are to be defined. The Fourier-domain representation of filter (B-2) is

\[F_9(k_x, k_y) = 4 \alpha [\cos k_x \cos k_y - 1] + 2 \gamma [\cos k_x + \cos k_y - 2], \quad (B-3)\]
and the isotropic filter that we can try to approximate is defined analogously to its one-dimensional equivalent, as follows:

\[ F(k_x, k_y) = 2(\cos k - 1) = 2(\cos \sqrt{k_x^2 + k_y^2} - 1) . \]  \hspace{1cm} (B-4)

Comparing equations (B-3) and (B-4), we notice that they match exactly, when either of the wavenumbers \( k_x \) or \( k_y \) is equal to zero, provided that

\[ \alpha = \frac{1 - \gamma}{2} . \]  \hspace{1cm} (B-5)

Therefore, we can reduce the problem to estimating a single coefficient \( \gamma \). Another way of expressing this conclusion is to represent filter \( F_9 \) in equation (B-3) as a linear combination of filter \( F_5 \) from equation (B-3) and its rotated version (Cole, 1994), as follows:

\[
F_9 = \gamma \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} + (1 - \gamma) \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & -2 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix} \]  \hspace{1cm} (B-6)

With the value of \( \gamma = 0.5 \), filter \( F_9 \) takes the value

\[
F_9 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & -3 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{pmatrix} \]  \hspace{1cm} (B-7)

and corresponds precisely to the nine-point McClellan filter (McClellan, 1973; Hale, 1991a). On the other hand, the value of \( \gamma = 2/3 \) gives the least error in the vicinity of the zero wavenumber \( k \). In this case, the filter is

\[
F_9 = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 2/3 & -10/3 & 2/3 \\ 1/6 & 2/3 & 1/6 \end{pmatrix} \]  \hspace{1cm} (B-8)

Errors of different approximations are plotted in Figure B-1

Under the helix transform, a filter of the general form (B-2) becomes equivalent to a one-dimensional filter with the \( Z \) transform

\[
F_9(Z) = \alpha Z^{-N_x-1} + \gamma Z^{-N_x} + \alpha Z^{-N_x+1} + \gamma Z^{-1} - 4 (\alpha + \gamma) + \gamma Z + \alpha Z^{N_x-1} + \gamma Z^{N_x} + \alpha Z^{N_x+1} ,
\]  \hspace{1cm} (B-9)

where \( N_x \) is the helix period (the number of grid points in the \( x \) dimension). To find the inverse of a convolution with filter (B-9), we factorize the filter into the causal minimum-phase component and its adjoint:

\[
F_9(Z) = P(Z)P(1/Z) .
\]  \hspace{1cm} (B-10)

\(^{\text{4}}\)Another way of constructing circular-symmetric filters is suggested by the rotated McClellan transform (Biondi and Palacharla, 1993).
Figure B-1: The numerical anisotropy error of different Laplacian approximations. Both the five-point Laplacian (plot a) and its rotated version (plot b) are accurate along the axes, but exhibit significant anisotropy in between at large wavenumbers. The nine-point McClellan filter (plot c) has a reduced error, while the filter with $\gamma = 2/3$ (plot d) has the flattest error around the origin.

To find the coefficients of the filter $P$, any one-dimensional spectral factorization method can be applied. It is important to point out that the result of factorization (neglecting the numerical errors) does not depend on $N_x$. Another approach is to define a residual error vector for the coefficients of $Z$ in equation (B-10) and minimize it for some particular norm. For example, minimizing the $L_1$ norm when $F_9$ is the McClellan filter (B-7), we discover that the filter $P$, after transforming back to two dimensions, takes the form

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01428</td>
<td>0.033513</td>
<td>0.0808</td>
<td>0.2543</td>
<td>0.3521</td>
</tr>
</tbody>
</table>

(B-11)

The results of applying a recursive deconvolution with filter (B-11) are shown in Figure B-2. An essentially similar procedure, only with a different set of filters, works for implicit wavefield extrapolation.
Figure B-2: Inverting the Laplacian operator by a helix deconvolution. The top left plot shows the input, which contains a single spike and the causal minimum-phase filter $P$. The top right plot is the result of inverse filtering. As expected, the filter is deconvolved into a spike, and the spike turns into a smooth one-sided impulse. After the second run, in the backward (adjoint) direction, we obtain a numerical solution of Laplace’s equation! In the two bottom plots, the solution is shown with grayscale and contours. [findif/laplace/ inv-laplace]
The Wilson-Burg method of spectral factorization with application to helical filtering

Sergey Fomel*, Paul Sava†, James Rickett‡, and Jon F. Claerbout†∗

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 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 behavior in unconstrained regions is controlled by the tension parameter. We show that such surfaces can be efficiently constructed with recursive filter preconditioning and 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 multi-dimensional problems. Specifically, spectral factorization algorithms provide the key to rapid multi-dimensional 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 et al., 1998; Rickett, 2000; Zhang et al., 2000; Zhang and Shan, 2001), and 3-D noise attenuation (Ozdemir et al., 1999a,b; Rickett et al., 2001).

* Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas at Austin, University Station, Box X, Austin, Texas 78713-8972, USA
† Stanford Exploration Project, Department of Geophysics, Stanford University, Stanford, California 94305, USA
‡ ChevronTexaco Exploration and Production Technology Company, 6001 Bollinger Canyon Road, San Ramon, California 94583-2324
∗e-mail: sergey@sep.stanford.edu, jon@sep.stanford.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 multi-dimensional 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 Kailath, 2001) discuss 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 puts 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, reduces the computational effort from $O(N^3)$ operations to $O(N^2)$ operations per iteration. A different way to accelerate 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 two-dimensional 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.

**METHOD DESCRIPTION**

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

$$ S(Z) = A(Z)\hat{A}(1/Z) $$

(1)
for a given spectrum \( S(Z) \) and unknown minimum-phase signal \( A(Z) \) with an iterative linearization

\[
S(Z) = A_t(Z) \tilde{A}_t(1/Z) + A_t(Z) [\tilde{A}_{t+1}(1/Z) - \tilde{A}_t(1/Z)] + \tilde{A}_t(1/Z) [A_{t+1}(Z) - A_t(Z)]
\]

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 system (2) for the updated signal \( A_{t+1}(Z) \). Wilson (1969) presents 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, personal communication) recognized that dividing both sides of equation (2) by \( \tilde{A}_t(1/Z)A_t(Z) \) 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:

\[
1 + \frac{S(Z)}{\tilde{A}_t(1/Z) A_t(Z)} = \frac{A_{t+1}(Z)}{A_t(Z)} + \frac{\tilde{A}_{t+1}(1/Z)}{A_t(1/Z)}
\]

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 equation (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 1-D 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.
Comparison of Wilson-Burg and Kolmogoroff methods

The Kolmogoroff (cepstral or Hilbert transform) spectral factorization algorithm\cite{Kolmogoroff1939, Claerbout1976, Oppenheim1989} 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\cite{Rickett1999}, 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 \[\text{[number of iterations]} \times \text{[filter length]} \times N\]. If we keep the filter small and limit the number of iterations, the Wilson-Burg method can be cheaper (linear in $N$). In comparison, the cost of the original Wilson’s method is the \[\text{[number of iterations]} \times O(N^3)\].

- 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, $N$. 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.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
iter & $a_0$ & $a_1$ & $a_2$ & $a_3$ \\
\hline
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 \\
\hline
\end{tabular}
\caption{Example convergence of the Wilson-Burg iteration}
\end{table}
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 autocorrelation function has roots close to the unit circle, truncating filter coefficients may easily lead to non-minimum-phase filters. The Wilson-Burg 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 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. 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 Figure 1.)

A practical example is depicted in Figure 2. The symmetric Laplacian operator is often used in practice for regularizing smooth data. In order to construct a corresponding recursive preconditioner, we factor 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 \( C = PP^T \) operator, where \( P \) is deconvolution with the minimum-phase Laplacian. The application of \( C \) 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 behavior 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 two-dimensional case, a surface can be interpolated with biharmonic
Figure 1: Example of 2-D Wilson-Burg factorization. Top left: the input filter. Top right: its auto-correlation. Bottom left: the factor obtained by the Wilson-Burg method. Bottom right: the result of deconvolution. [burg/helix/ autowaves]
Figure 2: Creating a minimum-phase Laplacian filter. Top left: Laplacian filter. Top right: its auto-correlation (bi-harmonic filter). Bottom left: factor obtained by the Wilson-Burg method (minimum-phase Laplacian). Bottom right: the result of deconvolution.

Figure 3: 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).
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 of the 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 2-D 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 2-D 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 [2002]).

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:

\[
\iint_D \left| \nabla^2 f(x, y) \right|^2 \, dx \, dy ,
\]

where \( \nabla^2 \) 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
\[(\nabla^2)^2 f(x, y) = 0.\] \hfill (5)

Fung (1965) and Briggs (1974) derive equation (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-Krieger, 1968). Taking tension into account modifies both the energy formula (4) and the corresponding equation (5). Smith and Wessel (1990) suggest the following form of the modified equation:
\[
[(1 - \lambda)(\nabla^2)^2 - \lambda(\nabla^2)] f(x, y) = 0,
\] \hfill (6)

where the tension parameter \( \lambda \) ranges from 0 to 1. The corresponding energy functional is
\[
\int_D \left[ (1 - \lambda) |\nabla^2 f(x, y)|^2 + \lambda |\nabla f(x, y)|^2 \right] \, dx \, dy.
\] \hfill (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.

Norman Sleep (2000, personal communication) points 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 \( \lambda \) to a local function depending on \( x \) and \( y \). In a more general form, \( \lambda \) 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 equation (6) under the constraint
\[
f(x_k, y_k) = f_k.
\] \hfill (8)

We can accelerate the solution by recursive filter preconditioning. If \( A \) is the discrete filter representation of the differential operator in equation (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, 2002)
\[
K D^{-1} p \approx f_k,
\] \hfill (9)

where \( f_k \) represents the vector of known data, \( K \) is the operator of selecting the known data locations, and \( p \) is the preconditioned variable: \( p = D f \). After obtaining
an iterative solution of system (9), we reconstruct the model $f$ by inverse recursive filtering: $f = D^{-1} p$. Formulating the problem in helical 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 one-dimensional case, one finite-difference representation of the squared Laplacian is as a centered 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 equation (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 one-dimensional minimum-phase filters for different values of the parameter $\lambda$. Figure 5 demonstrates the interpolation results obtained with these filters on a simple one-dimensional 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 $\lambda$ allow us to achieve a compromise: a smooth surface with constrained oscillations.

![Figure 4: One-dimensional minimum-phase filters for different values of the tension parameter $\lambda$. The filters range from the second derivative for $\lambda = 0$ to the first derivative for $\lambda = 1$.](burg/tension/otens)

To design the corresponding filters in two dimensions, we define the finite-difference representation of operator (6) on a 5-by-5 stencil. 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:
Figure 5: Interpolating a simple one-dimensional synthetic with recursive filter preconditioning for different values of the tension parameter $\lambda$. The input data are shown on the top. The interpolation results range from a natural cubic spline interpolation for $\lambda = 0$ to linear interpolation for $\lambda = 1$. 

[Diagram showing different tension values]
The Laplacian representation with the same order of accuracy has the coefficients

\[
\begin{array}{cccc}
-1/360 & 2/45 & 0 & 2/45 & -1/360 \\
2/45 & -14/45 & -4/5 & -14/45 & 2/45 \\
0 & -4/5 & 41/10 & -4/5 & 0 \\
2/45 & -14/45 & -4/5 & -14/45 & 2/45 \\
-1/360 & 2/45 & 0 & 2/45 & -1/360 \\
\end{array}
\]

\[= \frac{1}{360}\]

For the sake of simplicity, we assumed equal spacing in the \(x\) and \(y\) direction. 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 small frequencies.

Figure 6: Spectra of the finite-difference splines-in-tension schemes for different values of the tension parameter (contour plots).
Regarding the finite-difference operators as two-dimensional auto-correlations and applying the Wilson-Burg method of spectral factorization, we obtain two-dimensional 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 $\lambda$. Keeping only the coefficients that are 200 times smaller that the leading coefficient, we obtain 25- and 16-point filters for respectively $\lambda = 0$ and $\lambda = 1$. The restricted filters do not factor 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 two-dimensional filters for different values of $\lambda$ and illustrates inverse recursive filtering, which is the essence of the helix method (Claerbout, 1998). The case of $\lambda = 1$ leads to the filter known as helix derivative (Claerbout, 2002). 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 Figure 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 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 turns spikes into inverse filter responses (middle plots). Adjoint filtering creates smooth isotropic shapes (right plots). The tension parameter takes on the values 0.3, 0.7, and 1 (from top to bottom). The case of zero tension corresponds to Figure 3.
of \( \lambda \) exhibit intermediate properties. Theoretically, the inverse impulse response of the filter corresponds to the Green’s function of equation (6). The theoretical Green’s function for the case of \( \lambda = 1 \) is

\[
G = \frac{1}{2\pi} \ln r ,
\]

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:

\[
G = \frac{1}{8\pi} r^2 \ln r .
\]

The theoretical Green’s function expression for an arbitrary value of \( \lambda \) is unknown\( ^\dagger \), 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 two-dimensional interpolation problem.

**Regularization example**

We chose an environmental dataset \( ^{[Claerbout, 2002]} \) 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-neighbor 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 \( \lambda \)) are shown in Figures 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 one-dimensional example (Figure 5). It is clearly seen in the cross-section plots in Figure 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 artifacts, but it also produces an undesirable non-smooth behavior in the middle part of the image. As in the one-dimensional example, intermediate tension allows us to achieve a compromise: smooth interpolation in the middle and constrained behavior at the sides of the sea bottom.

**CONCLUSIONS**

The Wilson-Burg spectral factorization method, presented in this paper, allows one to construct stable recursive filters. The method appears to have attractive compu-

\( ^\dagger \text{Mitášová and Mitás (1993) derive an analytical Green’s function for a different model of tension splines using special functions.} \)
tional 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 two-dimensional 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.

ACKNOWLEDGMENTS

This paper owes a great deal to John Burg. We would like to thank him and Francis Muir for many useful and stimulating discussions. The first author 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
Figure 10: The Sea of Galilee dataset 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.
Figure 11: Cross-sections of the Sea of Galilee dataset after missing-data interpolation with helical preconditioning. Different plots correspond to different values of the tension parameter.
Exploration Project.

REFERENCES

Bauer, F. L., 1955, Ein direktes iterations verfahren zur hurwitz-zerlegung eines poly-

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

Briggs, I. C., 1974, Machine contouring using minimum curvature: Geophysics, 39,

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

——–, 2002, Image estimation by example: Geophysical soundings image construc-
tion: Stanford Exploration Project.


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

——–, 2003, Multidimensional recursive filter preconditioning in geophysical estima-


factorization of Laurent polynomials: Advances in Computational Mathematics, 7,

Kolmogoroff, A. N., 1939, Sur l’interpolation et extrapolation des suites stationnaires:

Laurie, D. P., 1980, Efficient implementation of wilson’s algorithm for factorizing a

sequences as one-dimensional sequences: IEEE Trans. on Acoustics, Speech, and

Mitášová, H., and L. Mitáš, 1993, Interpolation by regularized splines with tension:

Hall.

Ozdemir, A., A. Ozbek, R. Ferber, and K. Zerouk, 1999a, f-xy projection filtering
1231–1234.

Ozdemir, A. K., A. Ozbek, R. Ferber, and K. Zerouk, 1999b, F-Xy projection via


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


Spectral factorization revisited

Paul Sava and Sergey Fomel

ABSTRACT
In this paper, we review some of the iterative methods for the square root, showing that all these methods belong to the same family, for which we find a general formula. We then explain how those iterative methods for real numbers can be extended to spectral factorization of auto-correlations. The iteration based on the Newton-Raphson method is optimal from the convergence stand point, though it is not optimal as far as stability is concerned. Finally, we show that other members of the iteration family are more stable, though slightly more expensive and slower to converge.

INTRODUCTION

Spectral factorization has been recently revived by the advent of the helical coordinate system. Several methods are reported in the literature, ranging from Fourier domain methods, such as Kolmogoroff’s (Claerbout 1992, Kolmogoroff 1939), to iterative methods, such as the Wilson-Burg method (Claerbout 1999, Wilson 1969, Sava et al. 1998).

In this paper, after reviewing the general theory of root estimation by iterative methods, we derive a general square root relationship applicable to both real numbers and to auto-correlation functions. We introduce a new spectral factorization relation and show its relation to the Wilson-Burg method.

THE SQUARE ROOT OF REAL NUMBERS

This section briefly reviews some well known square root iterative algorithms, and derives the Newton-Raphson and Secant methods. It also shows that Muir’s iteration for the square root (Claerbout 1995) belongs to the same family of iterative methods, if we make an appropriate choice of the generating function.

*e-mail: paul@sep.stanford.edu, sergey@sep.stanford.edu

239
Root-finding recursions

Given a function $f(x)$ and an approximation for one of its roots $x_n$, we can find a better approximation for the root by linearizing the function around $x_n$

$$f(x) \approx f(x_n) + (x_{n+1} - x_n)f'(x_n)$$

and by setting $f(x)$ to be zero for $x = x_{n+1}$. We find that

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (1)$$

1. Newton-Raphson’s method for the square root

A common choice of the function $f$ is $f(x) = x^2 - s$. This function has the advantage that it is easily differentiable, with $f'(x) = 2x$. The recursion relation thus becomes

$$x_{n+1} = x_n - \frac{x_n^2 - s}{2x_n} = \frac{x_n}{2} + \frac{s}{2x_n}$$

or

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{s}{x_n} \right)$$

or, after rearrangement,

$$x_{n+1} = \frac{s + x_n^2}{2x_n} \quad (2)$$

The recursion (2) converges to $\pm \sqrt{s}$ depending on the sign of the starting guess $x_0 \neq 0$.

2. Secant method for the square root

A variation of the Newton-Raphson method is to use a finite approximation of the derivative instead of the differential form. In this case, the approximate value of the derivative at step $n$ is

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

For the same choice of the function $f$, $f(x) = x^2 - s$, we obtain

$$x_{n+1} = x_n - \frac{x_n^2 - s}{x_n + x_{n-1}}$$

and

$$x_{n+1} = \frac{s + x_n x_{n-1}}{x_n + x_{n-1}} \quad (3)$$
In this case, recursion (3) also converges to $\pm \sqrt{s}$ depending on the sign of the starting guesses $x_0$ and $x_1$.

3. Muir’s method for the square root

Another possible iterative relation for the square root is Francis Muir’s, described by Claerbout (1995):

$$x_{n+1} = \frac{s + x_n}{x_n + 1}$$

This relation belongs to the same family of iterative schemes as Newton and Secant, if we make the following special choice of the function $f(x)$ in (1):

$$f(x) = |x + \sqrt{s}| \frac{\sqrt{2} - 1}{\sqrt{2}} |x - \sqrt{s}| \frac{\sqrt{2} + 1}{\sqrt{2}}$$

Figure 1 is a graphical representation of the function $f(x)$.

4. A general formula for the square root

From the analysis of equations (21), (3), and (4), we can derive the following general form for the square root iteration:

$$x_{n+1} = \frac{s + x_n \gamma}{x_n + \gamma}$$

where $\gamma$ can be either a fixed parameter, or the value of the iteration at the preceding step, as shown in Table 1. The parameter $\gamma$ is the estimate of the square root at the given step (Newton), the estimate of the square root at the preceding step (Secant), or a constant value (Muir). Ideally, this value should be as close as possible to $\sqrt{s}$. 
Table 1: Recursions for the square root

<table>
<thead>
<tr>
<th></th>
<th>( \gamma )</th>
<th>Recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muir</td>
<td>1</td>
<td>( x_{n+1} = \frac{s + x_n}{x_n + 1} )</td>
</tr>
<tr>
<td>Secant</td>
<td>( x_{n-1} )</td>
<td>( x_{n+1} = \frac{s + x_n x_{n-1}}{x_n + x_{n-1}} )</td>
</tr>
<tr>
<td>Newton</td>
<td>( x_n )</td>
<td>( x_{n+1} = \frac{s + x_n^2}{2x_n} )</td>
</tr>
<tr>
<td>Ideal</td>
<td>( \sqrt{s} )</td>
<td>( x_{n+1} = \frac{s + x_n \sqrt{s}}{x_n + \sqrt{s}} )</td>
</tr>
</tbody>
</table>

The convergence rate

We can now analyze which of the particular choices of \( \gamma \) is more appropriate as far as the convergence rate is concerned.

If we consider the general form of the square root iteration

\[
x_{n+1} = \frac{s + x_n \gamma}{x_n + \gamma}
\]

we can estimate the convergence rate by the difference between the actual estimation at step \((n + 1)\) and the analytical value \( \sqrt{s} \). For the general case, we obtain

\[
x_{n+1} - \sqrt{s} = \frac{s + \gamma x_n - x_n \sqrt{s} - \gamma \sqrt{s}}{x_n + \gamma}
\]

or

\[
x_{n+1} - \sqrt{s} = \frac{(x_n - \sqrt{s})(\gamma - \sqrt{s})}{x_n + \gamma}
\]  \( (7) \)

The possible selections for \( \gamma \) from Table 1 clearly show that the recursions described in Figure 2: Convergence plots for different recursive algorithms, shown in Table 1.
Table 2: Convergence rate

<table>
<thead>
<tr>
<th></th>
<th>γ</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muir</td>
<td>1</td>
<td>linear</td>
</tr>
<tr>
<td>Secant</td>
<td>$x_{n-1}$</td>
<td>quasi-quadratic</td>
</tr>
<tr>
<td>Newton</td>
<td>$x_n$</td>
<td>quadratic</td>
</tr>
</tbody>
</table>

in the preceding subsection generally have a linear convergence rate (that is, the error at step $n + 1$ is proportional to the error at step $n$), but can converge quadratically for an appropriate selection of the parameter $\gamma$, as shown in Table 2. Furthermore, the convergence is faster when $\gamma$ is closer to $\sqrt{s}$.

We therefore conclude that Newton’s iteration has the potential to achieve the fastest convergence rate. Ideally, however, we could use a fixed $\gamma$ which is a good approximation to the square root. The convergence would then be slightly faster than for the Newton-Raphson method, as shown in Figure 2.

SPECTRAL FACTORIZATION

We can now extend the equations derived for real numbers to polynomials of $Z$, with $Z = e^{i\omega t}$, and obtain spectral factorization algorithms similar to the Wilson-Burg method (Sava et al., 1998), as follows:

$$X_{n+1} = \frac{S + X_n \tilde{G}}{X_n + \tilde{G}}$$

(8)

If $L$ represents the limit of the series in (8),

$$L\tilde{L} + L\tilde{G} = S + L\tilde{G}$$

and so

$$L\tilde{L} = S$$

Therefore, $L$ represents the causal or anticausal part of the given spectrum $S = XX$.

Table 3 summarizes the spectral factorization relationships equivalent to those established for real numbers in Table 1.

The convergence properties are similar to those derived for real numbers. As shown above, the Newton-Raphson method should have the fastest convergence.
Table 3: Spectral factorization

<table>
<thead>
<tr>
<th>Method</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>(X_{n+1} = \frac{S + X_n \bar{X}_n}{X_n \bar{X}_n + G})</td>
</tr>
<tr>
<td>Muir</td>
<td>(X_{n+1} = \frac{S + X_n}{X_n \bar{X}_n + G})</td>
</tr>
<tr>
<td>Secant</td>
<td>(X_{n+1} = \frac{S + X_{n-1} X_n}{X_n + X_{n-1}})</td>
</tr>
<tr>
<td>Newton</td>
<td>(X_{n+1} = \frac{S + X_n}{2X_n})</td>
</tr>
<tr>
<td>Ideal</td>
<td>(X_{n+1} = \frac{S + X_n \sqrt{S}}{X_n + \sqrt{S}})</td>
</tr>
</tbody>
</table>

A COMPARISON WITH THE WILSON-BURG METHOD

For reasons of symmetry, we can take Newton’s relation from Table 3

\[ X_{n+1} = \frac{S + X_n \bar{X}_n}{2X_n} \]

and convert it to

\[ \frac{X_{n+1}}{2X_n} = \frac{S + X_n \bar{X}_n}{(2X_n)(2\bar{X}_n)}. \]

We can then consider a symmetrical relation where on the left side we insert the anticausal part of the spectrum, and obtain

\[ \frac{\bar{X}_{n+1}}{2\bar{X}_n} = \frac{S + X_n \bar{X}_n}{(2X_n)(2\bar{X}_n)}. \]

Finally, we can sum the preceding two equations and get

\[ \frac{X_{n+1}}{2X_n} + \frac{\bar{X}_{n+1}}{2\bar{X}_n} = \frac{2S + X_n \bar{X}_n + \bar{X}_n X_n}{(2X_n)(2\bar{X}_n)} \quad (9) \]

which can easily be shown to be equivalent to the Wilson-Burg relation

\[ \frac{X_{n+1}}{X_n} + \frac{\bar{X}_{n+1}}{\bar{X}_n} = 1 + \frac{S}{X_n \bar{X}_n} \quad (10) \]

In an analogous way, we can take the general relation from Table 3

\[ X_{n+1} = \frac{S + X_n G}{X_n + G} \]

and convert it to

\[ \frac{X_{n+1}}{X_n + G} = \frac{S + X_n \bar{G}}{(X_n + G)(\bar{X}_n + G)} \]
We can then consider a symmetrical relation where on the left side we insert the anticausal part of the spectrum, and obtain

\[
\frac{\hat{X}_{n+1}}{X_n + G} = \frac{S + \hat{X}_n G}{(X_n + G)(X_n + \hat{G})}
\]

Finally, we can sum the preceding two equations and get

\[
\frac{X_{n+1}}{X_n + G} + \frac{\hat{X}_{n+1}}{X_n + \hat{G}} = \frac{2S + X_n \hat{G} + \hat{X}_n G}{(X_n + G)(X_n + \hat{G})}
\]  \( (11) \)

Equation (11) represents our general formula for spectral factorization. If we consider the particular case when \( G \) is \( X_n \), we obtain equation (10), which we have shown to be equivalent to the Wilson-Burg formula.

From the computational standpoint, our equation is more expensive than the Wilson-Burg because it requires two more convolutions on the numerator of the right-hand side. However, our equation offers more flexibility in the convergence rate. If we try to achieve a quick convergence, we can take \( G \) to be \( X_n \) and get the Wilson-Burg equation. On the other hand, if we worry about the stability, especially when some of the roots of the auto-correlation function are close to the unit circle, and we fear losing the minimum-phase property of the factors, we can take \( G \) to be some damping function, more tolerant of numerical errors.

Moreover, by using the Equation (11), we can achieve fast convergence in cases when the auto-correlations we are factorizing have a very similar form, for example, in nonstationary filtering. In such cases, the solution at the preceding step can be used as the \( G \) function in the new factorization. Since \( G \) is already very close to the solution, the convergence is likely to occur quite fast.

**CONCLUSIONS**

The general iterative formula for the square root that we derived can be extended to the factorization of the auto-correlation functions. The Wilson-Burg algorithm is a special case of our more general formula. Using such a general formula provides flexibility in choosing between fast convergence and stability. We can achieve fast convergence when factorizing auto-spectra that have a very similar form. This improvement in convergence rate can have a useful application, for instance, in non-stationary preconditioning.

**ACKNOWLEDGMENTS**

We thank Jon Claerbout, who brought Muir’s iterative scheme to our attention, and suggested its application to spectral factorization.
REFERENCES

Plane wave prediction in 3-D

Sergey Fomel

INTRODUCTION

The theory of plane-wave prediction in three dimensions is described by Claerbout (1993, 1999). Predicting a local plane wave with T-X filters amounts to finding a pair of two-dimensional filters for two orthogonal planes in the 3-D space. Each of the filters predicts locally straight lines in the corresponding plane. The system of two 2-D filters is sufficient for predicting all but purely vertical plane waves. In the latter case, a third 2-D filter for the remaining orthogonal plane is needed. Schwab (1998) discusses this approach in more detail.

Using two prediction filters implies dealing with two filtering output volumes for each input volume. This situation becomes inconvenient when one uses the prediction output as a measure of coherency in the input volume (Claerbout, 1993; Schwab et al., 1996). Two outputs are obviously more difficult to interpret than one, and there is no natural way of combining them into one image. Another difficulty arises when plane-wave destructors are used for regularizing linear inverse problems (Clapp et al., 1997). We cannot apply an efficient recursive preconditioning (Claerbout, 1998a) unless the regularization operator is square, or, in other words, only one plane-wave destructor is involved.

Helical filtering (Claerbout, 1998b) brings us new tools for addressing this problem. In this paper, I show how to combine orthogonal 2-D plane predictors into a single three-dimensional filter with similar spectral properties. The 3-D filter can then work for coherency measurements or for preconditioning 3-D inverse problems. The construction employs the Wilson-Burg method of spectral factorization, adapted for multidimensional filtering with the help of the helix transform (Sava et al., 1998).

I use simple synthetic examples to demonstrate the applicability of plane-wave prediction to 3-D problems.

FACTORIZING PLANE WAVES

Let us denote the coordinates of a three-dimensional space by \( t, x, \) and \( y \). A theoretical plane wave is described by the equation

\[
U(t, x, y) = f(t - px x - py y), \quad (1)
\]

*e-mail: sergey@sep.stanford.edu*
where \( f \) is an arbitrary function, and \( p_x \) and \( p_y \) are the plane slopes in the corresponding direction. It is easy to verify that a plane wave of the form (1) satisfies the following system of partial differential equations:

\[
\begin{aligned}
\left( \frac{\partial}{\partial x} + p_x \frac{\partial}{\partial t} \right) U &= 0 \\
\left( \frac{\partial}{\partial y} + p_y \frac{\partial}{\partial t} \right) U &= 0
\end{aligned}
\]  

(2)

The first equation in (2) describes plane waves on the \( \{t,x\} \) slices. In its discrete form, it is represented as a convolution with the two-dimensional finite-difference filter \( A_x \). Similarly, the second equation transforms into a convolution with filter \( A_y \), which acts on the \( \{t,y\} \) slices. The discrete (finite-difference) form of equations (2) involves a blocked convolution operator:

\[
\begin{pmatrix}
A_x \\
A_y
\end{pmatrix} U = 0 .
\]

(3)

where \( U \) is the model vector corresponding to \( U(t,x,y) \).

In many applications, we are actually interested in the spectrum of the prediction filter, which approximates the inverse spectrum of the predicted data. In other words, we deal with the square operator

\[
\begin{pmatrix}
A_x^T & A_y^T
\end{pmatrix} \begin{pmatrix}
A_x \\
A_y
\end{pmatrix} = A_x^T A_x + A_y^T A_y .
\]

(4)

If we were able to transform this operator to the form \( A^T A \), where \( A \) is a three-dimensional minimum-phase convolution, we could use the three-dimensional filter \( A \) in place of the inconvenient pair of \( A_x \) and \( A_y \).

The problem of finding \( A \) from its spectrum is known as spectral factorization. It is well understood for 1-D signals (Claerbout, 1976), but until recently it was an open problem in the multidimensional case. Helix transform maps multidimensional filters to 1-D by applying special boundary conditions and allows us to use the full arsenal of 1-D methods, including spectral factorization, on multidimensional problems (Claerbout, 1998b). A problem, analogous to (4), has already occurred in the factorization of the discrete two-dimensional Laplacian operator:

\[
\Delta = \nabla^T \nabla = \begin{pmatrix}
D_x^T & D_y^T
\end{pmatrix} \begin{pmatrix}
D_x \\
D_y
\end{pmatrix} = H^T H ,
\]

(5)

where \( D_x \) and \( D_y \) represent the partial derivative operators along the \( x \) and \( y \) dimensions, respectively, and the two-dimensional filter \( H \) has been named helix derivative (Claerbout, 1999; Zhao, 1999).

If we represent the filter \( A_x \) with the help of a simple first-order upwind finite-difference scheme

\[
U_{x+1}^t - U_x^t + p_x (U_{x+1}^{t+1} - U_{x+1}^t) = 0 ,
\]

(6)
then, after the helical mapping to 1-D, it becomes a one-dimensional filter with the 
$Z$-transform

$$A_x(Z) = 1 - p_x Z^{N_t+1} + (p_x - 1) Z^{N_t},$$  \hspace{1cm} (7)

where $N_t$ is the number of samples on the $t$-axis. Similarly, the filter $A_y$ takes the 
form

$$A_y(Z) = 1 - p_y Z^{N_t N_x+1} + (p_y - 1) Z^{N_t N_x}.$$  \hspace{1cm} (8)

The problem is reduced to a 1-D spectral factorization of

$$A_x(1/Z)A_x(Z) + A_y(1/Z)A_y(Z) = -p_y Z^{N_t N_x} - \frac{1}{Z} \left[ \frac{1}{Z} \right] - p_x Z^{N_t} + \frac{1}{Z} \left[ \frac{1}{Z} \right] + [p_x (1 - p_x) + p_y (1 - p_y)] \frac{1}{Z} + 2 + p_x (p_x - 1) + p_y (p_y - 1) + [p_x (1 - p_x) + p_y (1 - p_y)] Z + \frac{1}{Z} \left[ \frac{1}{Z} \right] = (p_x - 1) Z^{N_t-1} - p_x Z^{N_t+1} + (p_y - 1) Z^{N_t N_x} - p_y Z^{N_t N_x+1}.$$  \hspace{1cm} (9)

After a minimum-phase factor of (9) has been found, we can use it for 3-D forward 
and inverse convolution.

All examples in this paper actually use a slightly more sophisticated formula for 
2-D plane-wave predictors:

$$A_x(Z) = 1 + \frac{p_x}{2} (1 - p_x) Z^{N_t-1} + (p_x^2 - 1) Z^{N_t} - \frac{p_x}{2} (1 + p_x) Z^{N_t+1}.$$  \hspace{1cm} (10)

Formula (10) corresponds to the Lax-Wendroff finite difference scheme (Clapp et al., 
1997). It provides a valid approximation of the plane-wave differential equation for 
$-1 \leq p_x \leq 1$.

Figure 1: 3-D plane wave prediction with a 402-point filter. Left: $p_x = 0.7$, $p_y = 0.5$. 
Right: $p_x = -0.7$, $p_y = 0.5$.  plane/eplane/ eplane

Figure shows examples of plane-wave construction. The two plots in the figure 
are outputs of a spike, divided recursively (on a helix) by $A^T A$, where $A$ is a 3- 
D minimum-phase filter, obtained by Wilson-Burg factorization. The factorization
Figure 2: Schematic filter shape for a 26-point 3-D plane prediction filter. The dark block represents the leading coefficient. There are 9 blocks in the first row and 17 blocks in the second row.

Figure 3: 3-D plane wave prediction with a 26-point filter. Left: $p_x = 0.7, p_y = 0.5$. Right: $p_x = -0.7, p_y = 0.5$.
was carried out in the assumption of $N_t = 20$ and $N_x = 20$; therefore, the filter had $N_t N_x + 2 = 402$ coefficients. Using such a long filter may be too expensive for practical purposes. Fortunately, the Wilson-Burg method allows us to specify the filter length and shape beforehand. By experimenting with different filter shapes, I found that a reasonable accuracy can be achieved with a 26-point filter, depicted in Figure 2. Plane-wave construction for a shortened filter is shown in Figure 3. The predicted plane wave is shorter and looks more like a slanted disk. It is advantageous to deal with short plane waves if the filter is applied for local prediction of non-stationary signals.

Clapp (2000) has proposed constructing 3-D plane-wave destruction (steering) filters by splitting. In Clapp’s method, the two orthogonal 2-D filters $A_x$ and $A_y$ are simply convolved with each other instead of forming the autocorrelation (4). While being a much more efficient approach, splitting suffers from induced anisotropy in the inverse impulse response. Figure 4 illustrates this effect in the 2-D plane by comparing the inverse impulse responses of plane-wave filters obtained by spectral factorization and splitting. The splitting response is evidently much less isotropic.

Figure 4: Two-dimensional inverse impulse responses for filters constructed with spectral factorization (left) and splitting (right). The splitting response is evidently much less isotropic. [plane/bob/ bob]

In the next sections, I address the problem of estimating plane-wave slopes and show some examples of applying local plane-wave prediction in 3-D problems.

**ESTIMATING PLANE WAVES**

It may seem difficult to estimate the plane slope $p_x$ for a Lax-Wendroff filter of the form (10) because $p_x$ appears non-linearly in the filter coefficients. However, using the analytical form of the filter, we can easily linearize it with respect to the plane slope and set up a simple iterative scheme:

$$p_x^{(k+1)} = p_x^{(k)} + \Delta p_x^{(k)},$$ (11)
where $k$ stands for the iteration count, and $\Delta p_x^{(k)}$ is found from the linearized equation

$$ (A'_x U) \Delta p_x = -A_x U , \tag{12} $$

where $A'_x$ is the derivative of $A_x$ with respect to $p_x$. To avoid unstable division by zero when solving equation (12) for $\Delta p_x$, Adding a regularization equation

$$ \epsilon \nabla \Delta p_x \approx 0 , \tag{13} $$

where $\epsilon$ is a small scalar regularization parameter, I solve system (12,13) in the least-square sense to obtain a smooth slope variation $\Delta p_x$ at each iteration. In practice, iteration process (11) quickly converges to a stable estimate of $p_x$.

**EXAMPLES**

Two simple examples in this section demonstrate an application of 3-D local plane-wave prediction to the problems of discontinuity enhancement and missing data interpolation.

**3-D discontinuity enhancement**

Figure 5: A synthetic model, showing a fault between two plane waves of different slopes.

Figure 5 shows a simple synthetic model of two plane waves, separated by a plane fault. The slope estimates for the two orthogonal directions are shown in Figure 6. We can see that the estimation procedure correctly identified the regions of constant slope. Finally, estimating a local 3-D plane-wave predictor by spectral factorization and convolving the resultant non-stationary filter with the input model, we obtain the prediction residual, shown in Figure 7. In the residual, both plane waves are effectively destroyed, and we observe a sharp image of the fault plane. This result compares favorably with results of alternative methods, collected by Schwab (1998).
Figure 6: Plane wave slope estimates in the $x$ and $y$ directions (left and right plots, respectively) from the synthetic two-plane model.

Figure 7: Magnitude of the residual after convolving the synthetic two-plane model with a local 3-D plane wave filter.
3-D missing data interpolation

Figure 8: Claerbout’s “qdome” synthetic model.

Figure 9: Plane wave slope estimates in the $x$ and $y$ directions (left and right plots, respectively) from the “qdome” model.

Figures 8 and 9 show Claerbout’s “qdome” synthetic model (Claerbout, 1993, 1999) and its corresponding slope estimates. In a missing data interpolation experiment, I remove 75% of the traces in the original model, arriving at the missing data model, shown in the left plot of Figure 10. The missing data interpolation result is shown in the right plot of Figure 10. Most of the original signal, except for some high-curvature areas, has been restored. Local 3-D plane-wave predictors allow us to use the efficient interpolation technique of Fomel et al. (1997), based on recursive filter preconditioning.

CONCLUSIONS

I have shown that a 3-D plane-wave prediction filter can be constructed from a pair of two-dimensional filters by using helix transform and a one-dimensional spectral
factorization algorithm.

In all the examples, I used analytical finite-difference filters instead of more general prediction-error filters. A similar factorization idea could be applied to 3-D prediction-error filters. However, treating non-stationarity in this case is less straightforward and requires additional care (Crawley et al., 1998; Clapp et al., 1999).

3-D plane-wave prediction filters can find many interesting applications in data processing and inversion. An especially promising application is solution steering in tomography-type problems (Clapp et al., 1997; Clapp and Biondi, 1998).

ACKNOWLEDGMENTS

Jon Claerbout suggested the problem of 3-D plane wave prediction and the idea of its solution. Unfortunately, he is currently not able to share the authorship.

REFERENCES

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


Solution steering with space-variant filters

Robert G. Clapp, Sergey Fomel, and Jon Claerbout

ABSTRACT

Most geophysical problems require some type of regularization. Unfortunately, most regularization schemes produce “smeared” results that are often undesirable when applying other criteria (such as geologic feasibility). By forming regularization operators in terms of recursive steering filters, built from a priori information sources, we can efficiently guide the solution towards a more appealing form. The steering methodology proves effective in interpolating low frequency functions, such as velocity, but performs poorly when encountering multiple dips and high frequency data. Preliminary results using steering filters for regularization in tomography problems are encouraging.

INTRODUCTION

When attempting to do inversion we are constantly confronted with the problem of slow convergence. Claerbout and Nichols [1994] suggested using a preconditioner to speed up convergence. Unfortunately, it is often difficult to find an appropriate preconditioner and/or the preconditioner is so computationally expensive that it negates the savings gained by reducing the number of iterations [Claerbout, 1994]. Claerbout [1997] proposed designing helicon-style operators to provide a method to find stable inverses, and potentially, appropriate preconditioners [Fomel et al., 1997; Fomel, 1997].

In addition, geophysical problems are often under-determined, requiring some type of regularization. Unfortunately, the simplest, and most common, regularization techniques tend to create isotropic features when we would often prefer solutions that follow trends. This problem is especially prevalent in velocity estimation. The result obtained through many inversion schemes produce a velocity structure that geologists (whose insights are hard to encode into the regression equations) find unreasonable [Etgen, 1997]. Fortunately, there are often other sources of information that can be encoded into the regularization operator that allow the inversion to be guided towards a more appealing result. For example, in the case of velocity estimation, reflector dips might be appropriate.

We create small, space-variant, steering filters from dip or other a priori information. We use the inverse of these filters to form a preconditioner which acts as our
regularization operator. We show this methodology applied to three different types of problems. In the first set of examples we interpolate well-log information using reflector dip as the basis for our steering filters. For the second set of examples we do a more traditional seismic data interpolation problem. Starting from a shot gather with a portion of the data missing. We use a velocity function to create hyperbolic paths, which in turn are used to construct steering filters. In the final example we show some preliminary results of using steering filters in conjunction with a tomographic operator to create velocity models which both satisfy the data and are geologically reasonable.

**THEORY/MOTIVATION**

**Regularization**

In general, geophysical problems are ill-posed. To obtain pleasing results we impose some type of regularization criteria such as diagonal scaling, limiting solutions to large singular values (Clapp and Biondi, 1995), or minimizing different solution norms (Nichols, 1994). The typical SEP approach is to minimize the power out of a regularization operator \(A\) applied to the model \(m\), described by the fitting goal

\[
0 \approx Am. \quad (1)
\]

Where \(A\)'s spectrum will be the inverse of \(m\), so to produce a smooth \(m\), we need a rough \(A\) (Claerbout, 1994). The regularization operator can take many forms, in order of increasing complexity:

**Laplacian operator** \((\nabla^2)\) The symmetric nature of the Laplacian leads to isotropic smoothing of the image.

**Steering filters** Simple plane wave annihilation filters which tend to orient the data in some preferential direction, chosen a priori. These filters can be simple two point filters, Figure 1, to larger filters that sacrifice compactness for more precise dip annihilation.

![Figure 1: An example of steering filter. In this case preference is given to slopes at 45 degrees.](1 -0.5 -0.5)

**Prediction Error Filters (PEF)** Like steering filters apply a preferential smoothing direction, but are not limited to a single dip and determine their smoothing directions from the known data (Schwab, 1997).
Preconditioning

Another important consideration is the speed of convergence of the problem. The size of most geophysical problems make direct matrix inversion methods impractical. An appealing alternative for linear problems is the family of conjugate gradient methods. Unfortunately, the operators used in seismic reflection problems are often computationally expensive. As a result it is important to minimize the number of steps it takes to get to a reasonable solution. One way that can reduce the number of iterations is by reformulating the problem in terms of some new variable \(x\) with a preconditioning operator \(B\). Changing a tradition inversion problem where the operator \(C\) maps the model \(m\) to the data \(d\),

\[
d \approx Cm
\]

we can rewrite

\[
d \approx CBx
\]

where

\[
m = Bx.
\]

Helix transform

The next question is how to choose \(B\)? We have three general requirements:

- it produces relatively smooth (by some criteria) results;
- it spreads information quickly;
- and it is computationally inexpensive.

By defining our operators via the helix method (Claerbout, 1997) we can meet all of these requirements. The helix concept is to transform \(N\)-Dimensional operators into 1-D operators to take advantage of the well developed 1-D theory. In this case we utilize our ability to construct stable inverses from simple, causal filters. We can set \(B\), from equation (4) to

\[
B = A^{-1},
\]

where \(A\) is the roughening operator from fitting goal (1), and \(B\) is simulated using polynomial division. If \(A\) is a small roughening operator, \(B\) is a large smoothing operator without the heavy costs usually associated with larger operators.

Steering Filters

At this point a discussion of steering filters is appropriate. Plane waves with a given slope on a discrete grid can be predicted (destroyed) with compact filters (Schwab
Inverting such a filter by the helix method, we can create a signal with a given arbitrary slope extremely quickly. If this slope is expected in the model, the described procedure gives us a very efficient method of preconditioning the model estimation problem, fitting goal (2).

How can a plane prediction (steering) filter be created? On the helix surface, the plane wave \( P(t, x) = f(t - px) \) translates naturally into a periodic signal with the period of \( T = N_t + \sigma \), where \( N_t \) is the number of points on the \( t \) trace, and \( \sigma = \frac{\Delta x}{\Delta t} \), where \( \sigma \) is the plane slope, \( \Delta x \) and \( \Delta t \) correspond to the mesh size. If we design a filter that is two columns long (assuming the columns go in the \( t \) direction), then the plane prediction problem is simply connected with the interpolation problem: to destroy a plane wave, shift the signal by \( T \), interpolate it, and subtract the result from the original signal. Therefore, we can formally write

\[
P = I - S(\sigma),
\]

where \( P \) denotes the steering filter, \( S \) is the shift-and-interpolation operator, and \( I \) is the identity operator.

Different choices for the operator \( S \) in (6) produce filters with different length and prediction power. A shifting operation corresponds to the filter with the \( Z \)-transform \( \Sigma(Z) = Z^T \), while the operator \( S \) corresponds to an approximation of \( \Sigma(Z) \) with integer powers of \( Z \). One possible approach is to expand \( \Sigma(Z)Z^{-N_t} \) using the Taylor series around the zero frequency (\( Z = 1 \)). For example, the first-order approximation is

\[
S_1(Z) = Z^{N_t}(1 + \sigma(Z - 1)) = (1 - \sigma)Z^T + \sigma Z^{T+1},
\]

which corresponds to linear interpolation and leads in the two-dimensional space to the steering filter \( P \) of the form

\[
\begin{array}{cc}
1 & \\
\sigma - 1 & -\sigma
\end{array}
\]

Filter (7) is equivalent to the explicit first-order upwind finite-difference scheme on the plane wave equation

\[
\frac{\partial P}{\partial x} + p \frac{\partial P}{\partial t} = 0.
\]

An important property of filter (7) is that it produces an exact answer for \( \sigma = 0 \) and \( \sigma = 1 \). The values of \( \sigma > 1 \) lead to unstable inversion. For negative \( \sigma \), the filter is reflected:

\[
P_1 = \begin{array}{cc}
1 & \\
\sigma & -\sigma - 1
\end{array}
\]

The top panel in Figure 2 shows a plane wave, created by applying the helix inverse of filter (7) on a single spike (unit impulse) for the value of \( \sigma = 0.7 \). We see a noticeable frequency dispersion, caused by the low order of the approximation.

\[\text{In computational physics, the dimensionless number } \sigma \text{ is sometimes referred to as the CFL (Courant, Friedrichs, and Lewy) number (Sod, 1985).}\]
Figure 2: Steering filters with Lagrange interpolation. The left and middle plots show the impulse responses of steering filters: the top panel corresponds to linear interpolation (two-point Lagrange, upwind finite-difference); the second top plot, the three-point Lagrange filter (Lax-Wendroff scheme); the two bottom plots, the 8-point and 13-point Lagrange filters. The right plots in each panel show the corresponding average spectrum. The spectrum flattens and the prediction get more accurate with an increase of the filter size. 

steer/steering/steer-lagrange
The second-order Taylor approximation yields

\[ S_2(Z) = Z^{N_t - 1} \left( 1 + \sigma (Z - 1) \frac{(\sigma - 1) \sigma (Z - 1)^2}{2} \right) = \frac{\sigma (\sigma - 1)}{2} Z^{T-1} + (1 - \sigma^2) Z^T + \frac{\sigma (\sigma + 1)}{2} Z^{T+1}, \]

which corresponds to the 2-D filter

\[ P_2 = \begin{bmatrix} 1 & \frac{1}{\sigma(1-\sigma)} & -\frac{\sigma(\sigma+1)}{2} \\ \frac{\sigma(1-\sigma)}{2(\sigma^2-1)} & 1-\sigma & 1+\sigma \\ -\frac{\sigma(\sigma+1)}{2} & 1+\sigma & -1 \end{bmatrix} \]

and is equivalent to the Lax-Wendroff finite-difference scheme of equation (6). The interpolation, implied by filter (10) is a local three-point polynomial (Lagrange) interpolation. The correspondence of the Taylor series method, described above, and the Lagrange interpolation can be proved by induction. In general, the filter coefficients for the second row of the \( N \)-th order Lagrangian filter are given by the explicit formula

\[ a_k = \prod_{i \neq k} \frac{(\sigma - \lceil N/2 \rceil - i)}{(k - i)}, \]

where the \( k \) and \( i \) range from 0 to \( N \). Such a filter has a stable inverse for \( -N/2 \leq \sigma \leq N/2 \) and additionally produces an exact answer for all integer \( \sigma \)'s in that range. We would have arrived at the same conclusion if instead of expanding the \( Z \)-transform of the filter \( S \) around \( Z = 1 \), expanded its Fourier transform around the zero frequency. The latter case corresponds to the “self-similar” construction of [Karrenbach (1995)]. The impulse responses for the helix inverses of different-order Lagrangian filters are shown in Figure 2.

If instead of Taylor series in \( Z \), we use a rational (Padè) approximation, the filter will get more than one coefficient in the first row, which corresponds to an implicit finite-difference scheme. For example, the \([1/1]\) Padè approximation is

\[ S_1^1(Z) = \frac{1 + \frac{1+\sigma}{2} (Z - 1)}{1 + \frac{1-\sigma}{2} (Z - 1)} = \frac{1 - \sigma + (1 + \sigma)Z}{1 + \sigma + (1 - \sigma)Z} \]

which leads to the filter

\[ P_1^1 = \begin{bmatrix} 1 & \frac{1-\sigma}{\sigma-1} & \frac{1+\sigma}{1-\sigma} \\ \frac{1-\sigma}{\sigma-1} & 1-\sigma & 1+\sigma \\ \frac{1+\sigma}{1-\sigma} & 1+\sigma & -1 \end{bmatrix} \]

and corresponds to the Crank-Nicolson implicit scheme.

It is interesting to note that a space-variant convolution with inverse plane filters can create signals with different shape, which remains planar only locally. This situation corresponds to a variable slowness \( p \) in the one-way wave equation (6). Figure 3 shows an example: predicting hyperbolas with a 7-point Lagrangian filter.
Space variable filters

Steering filters are effective in spreading information along a given direction, but are limited to a single dip. If it is inappropriate to apply a single smoothing direction to the entire model there are two general courses of action:

Patching (Claerbout 1992b; Schwab and Claerbout 1995) Redefine our problem into a series of problems, each on a small subset of the data where the stationarity assumption is valid, then recombine the data. This approach leads to problems in determining subsets where the stationarity condition is satisfied and how to effectively remove patching boundaries from the final output.

Space varying filters Filters that vary with location but are spatially smooth. In many ways this is the more appealing approach. In the past, space varying filters have not been used because they impose significant memory issues (a filter at every location) and must be spatially smooth. By choosing steering filters for our regularization operator and using helix enabled polynomial division, these weaknesses are significantly diminished. We can construct and store relatively small filters which are much easier to smooth (smoothing the preferential dip direction is sufficient). In addition the polynomial division produced inverse filters will have an even higher level of smoothness because each filter spreads information over large, overlapping regions at each iteration.

WELL LOG/DIP INTERPOLATION

To illustrate the effectiveness of this method imagine a simple interpolation problem. Following the methodology of (Fomel et al. 1997) we first bin the data, producing a model $m$, composed of known data $m_k$ and unknown data $m_u$. We have an operator
which is simply a diagonal masking operator with zeros at known data locations and ones at unknown locations. We can write \( m_k \) and \( m_u \) in terms of \( m \) and \( J \),

\[
\begin{align*}
  m_k & \approx (I-J)m \\
  m_u & \approx Jm
\end{align*}
\]

where \( I \) is the identity matrix. We have the preconditioning operator \( B \), which applies polynomial division using the helix methodology. In this case we have a single equation in our estimation problem,

\[
m_k \approx (I - J)Bx.
\]

So the only question that remains is what to use for \( B \), or more specifically \( B^{-1} \), \( A \).

For this experiment we create a series of well logs by subsampling a 2-D velocity field. We use as our a priori information source, reflector dips, to build our steering filters, and thus our operator \( A \). For this test we pick our dips from our “goal”, left portion of Figure 4. We define areas in which we believe each of these dips to be approximately correct, and smooth the overall field (right portion of Figure ??).

![Figure 4: Left, a synthetic seismic section with four picked reflectors indicated by ‘*’, right; the dip field constructed from the picked reflectors.](image)

For the first test, we simulate nine well logs along the survey (Figure ??). We use equation (18) as our fitting goal and a conjugate gradient solver to estimate \( x \). Within 12 iterations we have a satisfactory solution(Figure ??). If you look closely, especially near the bottom of the section you can still see the well locations, but in
general the solution converges quickly to something fairly close to the correct velocity field (Figure 5).

For a more difficult test, we decreased the number of wells, and give them varying lengths. In Figure 5 you see that in a few iterations we achieve a result quite similar to our goal. In addition, in areas far away from known data the method still followed the general dip direction simply at a lower frequency level.

**SHOT-GATHER BASED INTERPOLATION**

Another possible application for using recursive steering filters is to interpolate seismic data. As an initial test we chose to interpolate a shot gather. We used a $v(z)$ velocity function to construct hyperbolic trajectories, which in turn were used to construct our dip field (similar to the seismic dips used in the previous section).

For a first test we created a synthetic shot gather using a $v(z) = a + bz$ model as input to a finite difference code. We then cut a hole in this shot gather and attempted to recover the removed values. As Figure 7 shows we did a good job recovering the amplitude within a few iterations.

Left, synthetic shot gather; center, holes cut out of shot gather; right, inversion result after 15 iterations.

To see how the method reacted when it was given data that did not fit its model
Figure 6: Left model (our goal), middle well logs, and right estimated model after 12 iterations.  

steer/qdome/ combo4

Figure 7:  

steer/shot/ combo
(in this case hyperbolic moveout) we used a dataset with significant noise problems (ground roll, bad traces, etc.). Using the same technique as in Figure 7 we ended up with a result which did a fairly decent job fitting portions of the data where noise content was low, but a poor job elsewhere (Figure 8). Even where the method did the best job of reconstructing the data, it still left a visible footprint. A more esthetically pleasing result can be achieved by using the above method followed a more traditional interpolation problem using the operator $A$ and the fitting goal

$$\text{Am} \approx 0,$$  \hspace{1cm} (19)

where $m$ is initialized with the result of our previous inversion problem and not allowed to change at locations where we have data. The bottom right panel in Figure 8 shows the result of applying a few iterations of fitting goal (19) to the bottom left result in Figure 8. By using both methodologies the interpolated data does a much better job blending into its surroundings but still is a poor interpolation result.

**FUTURE WORK AND CONCLUSIONS**

We show that by using helicon enabled inverse operators built from small steering filters we can quickly obtain esthetically pleasing models. Tests on smooth models, with a single dip at each location proved successful. The methodology does not adequately handle models with multiple dips at each location and presupposes some knowledge of the desired final model. A different approach would be to estimate the steering filters ($S$) from the experimental data ($m$). Generally, this leads to a system of non-linear equations

$$P(\sigma)m = (I - S(\sigma))m = 0,$$  \hspace{1cm} (20)

which need to be solved with respect to $\sigma$. One way of solving system (20) is to apply the general Newton’s method, which leads to the iteration

$$\sigma_k = \sigma_{k-1} + \frac{P(\sigma_{k-1})m}{S'(\sigma_{k-1})m},$$  \hspace{1cm} (21)

where the derivative $S'(\sigma)$ can be computed analytically. It is interesting to note that if we start with $\sigma = 0$ and apply the first-order filter (79), then the first iteration of scheme (21) will be exactly equivalent to the slope-estimation method of Claerbout (1992a), used by Bednar (1997) for calculating coherency attributes. Finally, the steering filter regularization methodology needs to be tried in conjunction with a variety of operators and applied to real data problems.

**REFERENCES**

Figure 8: Top left, original shot gather; top right, gather with holes (input); bottom left, result applying equation 18; bottom right, result after applying equation (18) followed by (19).
INTRODUCTION

Locally, seismic data is a superposition of plane waves. The statistical properties of such superpositions are relevant to geophysical estimation and they are not entirely obvious.

Clearly, a planar wave can be constructed from a planar distribution of point sources. Contrariwise, a point source can be constructed from a superposition of plane waves going in all directions. We can represent a random wave source either as a superposition of points or as a superposition of plane waves. Here is the question:

Given a superposition of infinitely many impulsive plane waves of random amplitudes and orientations, what is their spectrum?

If you said the spectrum is white, you guessed wrong. Figure 1 shows that it does not even look white. It is lower frequency than white for good reason. Mathematically independent variables are not necessarily statistically independent variables.

RESOLUTION OF THE PARADOX

If we throw impulse functions randomly onto a plane, the power spectrum of the plane is the power spectrum of impulse functions, namely white.

Think of a 2-D Gaussian whose contour of half-amplitude describes an ellipse of great eccentricity. In the limit of large eccentricity, this Gaussian could be one of the lines that we sprinkle on the plane with random amplitudes and orientations. The spatial spectrum of such an eccentric Gaussian must be lower than that of a symmetrical point Gaussian because the spectrum along the long axis of the ellipsoid is concentrated at very low frequency.

Consider a single delta function along a line with an arbitrary slope and location in a plane. The autocorrelation of this dipping line is another dipping line with the same slope, but passing through the origin at zero lag. The polarity of the impulse function is lost in the autocorrelation; in the autocorrelation space, the amplitude of the dipping line is positive.

Now consider a superposition of many dipping lines on the plane. Its autocorrelation is the sum of the autocorrelations of individual lines. The autocorrelation of any individual line is a line of the same slope that is translated to pass through the origin. [The 2-D autocorrelation is not shown in the graphics here. You’ll need to understand it from the words here. Sorry.] The autocorrelation is a superposition of lines of various slopes all passing through the origin, all having positive amplitude. This function would resemble a positive impulse function at the origin (and hence suggest a white spectrum). The function is actually not an impulse function, but, as we’ll see, it is the pole 1/r.
Figure 1: Top shows a superposition of 100 randomly positioned lines. Middle shows a superposition of 10,000 such lines. Bottom shows a superposition of 10,000 random point values. The bottom panel shows the most high frequency. Its spectrum is theoretically white. This paper claims that the middle panel is more representative of natural noises.
Consider an integral on a circular path around the origin. The circle crosses each line exactly twice. Thus the integral on this circular path is independent of the radius of the circle. Hence the average amplitude on the circumference is inverse with the circumference to keep the integral constant. Thus the autocorrelation function is the pole $1/r$.

**FOURIER TRANSFORM OF 1/R**

We would like to know the 2-D Fourier transform of $1/r$. Everywhere I found tables of 1-D Fourier transforms but only one place did I find a table that included this 2-D Fourier transform. It was at http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Statisti.html

Sergey Fomel showed me how to work it out: Express the FT in radial coordinates:

\[
\text{FT} \left( \frac{1}{r} \right) = \int \int \exp[i k_x r \cos \theta + i k_y r \sin \theta] \frac{1}{r} r \, dr \, d\theta
\]

(1)

\[
\text{FT} \left( \frac{1}{r} \right) = \int \delta[k_x \cos \theta + k_y \sin \theta] \, d\theta
\]

(2)

To evaluate the integral, we use the fact that $\int \delta(f(x)) \, dx = 1/|f'(x_0)|$ where $x_0$ is defined by $f(x) = 0$ and the definition $\theta_0 = \arctan(-k_x/k_y)$.

\[
\text{FT} \left( \frac{1}{r} \right) = \frac{1}{|− k_x \sin \theta_0 + k_y \cos \theta_0|} \quad (3)
\]

\[
\text{FT} \left( \frac{1}{r} \right) = \frac{1}{\sqrt{k_x^2 + k_y^2}} = \frac{1}{k_r} \quad (4)
\]

**UTILITY OF THIS RESULT**

At present we are accustomed to estimating statistical properties of seismic data by computing a 2-D prediction-error filter. This filter is needed to interpolate and extrapolate missing values.

Knowing that the prior spectral estimate is not a constant but instead is $1/k_r$, suggests a procedure that is more efficient statistically: By more efficient, I mean that a simpler model should fit the data, a model with fewer adjustable parameters.

**INTERPOLATION**

We’ll need to know a wavelet in the time and space domain whose amplitude spectrum is $\sqrt{k_r}$ (so its power spectrum is $k_r$). Do not mistake this for the the helix derivative
Claerbout (1998) whose power spectrum is $k_r^2$. What we need to use here is the square root of the helix derivative. Let the (unknown) wavelet with amplitude spectrum $\sqrt{k_r}$ be known as $G$.

1. Apply $G$ to the data. The prior spectrum of the modified data is now white.

2. Estimate the PEF of the modified data.

3. The interpolation filter for the original data is now $G$ times the PEF of the modified data.

Why is this more efficient? The important point is that the PEF should estimate the minimal practical number of freely adjustable parameters. If $G$ is a function that is lengthy in time or space, then the PEF does not need to be.

How important is this extra statistical efficiency? I don’t know.

**WHAT ARE THE NEXT STEPS?**

- Compute $\sqrt{k_r}$ in physical space and look at it. How best to do this? How best to package the software?

- Invent a synthetic data test.

- Think about how it might impact Sean (or Sergey). Which example of Sean’s would be worth redoing?

- Extend this idea to 3-D (lettuce versus noodles).

- Matt Schwab and I were frequently disappointed in the performance of local PEFs for the task of visualizing data. This might explain it. Which example of his might be worth redoing?

**REFERENCES**

Texture synthesis and prediction error filtering

Morgan Brown

ABSTRACT
The spectrum of a prediction-error filter (PEF) tends toward the inverse spectrum of the data from which it is estimated. I compute 2-D PEF’s from known “training images” and use them to synthesize similar-looking textures from random numbers via helix deconvolution. Compared to a similar technique employing Fourier transforms, the PEF-based method is generally more flexible, due to its ability to handle missing data, a fact which I illustrate with an example. Applying PEF-based texture synthesis to a stacked 2-D seismic section, I note that the residual error in the PEF estimation forms the basis for “coherency” analysis by highlighting discontinuities in the data, and may also serve as a measure of the quality of a given migration velocity model. Last, I relate the notion of texture synthesis to missing data interpolation and show an example.

INTRODUCTION
In terms of digital images, the word texture might be defined as, “an attribute representing the spatial arrangement of gray levels of the pixels in a region,” (IEEE [1990]). In the same context, I define texture synthesis as the process of first estimating the spatial statistical properties of a known image and then imparting these statistics onto a second (random) image. Figure 1 illustrates the general approach taken here: an uncorrelated image is transformed into one with the same statistical qualities as a known “training image” (TI), through an as-yet undefined filtering operation.

Texture synthesis is an active area of research in the computer graphics community, owing to the need for realistic, quickly generated surface textures (Simoncelli and Portilla 1998; Heeger and Bergen 1995; Brown and Mao 1998), but the same notion of texture applies to the earth sciences as well. Physically measurable quantities, be they geology, gravity, or topography, behave in certain repeatable ways as a function of space, i.e., these quantities have a given texture. Inversion problems are often underdetermined, hampered by a lack of “hard” measurements, causing a nullspace of high dimension. A priori “soft” constraints on functional form of the unknown model help in suppressing the nullspace of modeling operators. These a priori constraints can be conceptualized as textures. For instance, in velocity analysis and tomography, the earth’s velocity field is sometimes assumed to have a “blocky” texture (Clapp

*e-mail: morgan@sep.stanford.edu
Figure 1: The generalized texture synthesis algorithm. From the training image (TI), statistics are extracted and encoded into a filtering operation, which forces an uncorrelated image to have the same statistical qualities, or texture, as the TI.

et al. [1998]. Underdetermined inverse interpolation problems are often regularized by assuming “smooth” model texture (Claerbout 1998a).

The prediction-error filter (PEF) is an autoregressive filter which has the distinction of capturing the inverse spectrum of the data it is regressed upon. Because it captures this essential statistical property of the data, the PEF is a candidate for the generic "filter" operation shown in Figure 1.

This paper is intended as a follow-up to the earlier work by Claerbout and Brown (1999), which presented a texture synthesis technique utilizing 2-D PEF’s and 2-D deconvolution via the helix transform (Claerbout 1998b). First I motivate the texture synthesis problem by applying a Fourier transform-based technique to create synthetic textures of everyday objects, then introduce and apply a PEF-based technique to synthesize the same images. I compare the results of the two methods and conclude that the PEF-based method is the better choice because it more naturally handles missing data. Next I apply the PEF-based method to a 2-D stacked seismic section. The nature of the residual error in the PEF estimation of this example suggests application to seismic discontinuity detection and migration velocity analysis. Last, I solve a simple missing data problem to illustrate how regularization with a PEF imparts a reasonable “texture” onto the nullspace.

FOURIER TRANSFORM METHOD

The texture synthesis methodology of this paper really boils down to one of spectral estimation. An image’s amplitude spectrum contains the relative weights between frequency components, while the phase spectrum localizes these frequency components in space (Castleman 1996). Therefore, it stands to reason that the texture of stationary, loosely correlated images is adequately modeled using the amplitude spectrum alone. This idea is the basis for the Fourier Transform method of texture synthesis: all “realizations” of texture synthesis are forced to have the same amplitude spectrum,
differing only in phase. The following is an outline of the method.

1. Given a training image, \( t(x, y) \), compute its amplitude spectrum:

\[
R(k_x, k_y) = T^*(k_x, k_y)T(k_x, k_y)
\]  

(1)

2. Create random phase function: \( \phi_r(k_x, k_y) = \text{random numbers} \).

3. Reconstruct by substituting random phase:

\[
t_{\text{recon}}(x, y) = \mathcal{F}^{-1}\left\{ \sqrt{|R(k_x, k_y)|} \ e^{i\phi_r(k_x, k_y)} \right\}
\]

(2)

Figures 2 and 3 illustrate the Fourier transform method of texture synthesis. Clockwise from top-left: the training image, the synthesized image, the TI’s amplitude spectrum, and the TI’s phase spectrum.

Figure 2: Smoothed random image and Fourier transform synthesis. The TI is stationary, so the synthesis result is convincing. Notice that the true phase, in the regions where the modulating amplitude spectrum is nonzero, is quite random in appearance.

PEF-BASED METHOD

Theoretically, the convolution of data \( (N_d \text{ points}) \) and a PEF \( (N_a \text{ coefficients}) \) estimated from the data is approximately uncorrelated in the limit \( N_a \to N_d \to \infty \): a spike at zero lag plus Gaussian, independent identically distributed (iid) noise elsewhere. Thus the spectrum of this residual error is approximately white. The frequency response of the “inverse PEF”, as computed by deconvolution, is an \( N_a \)-point parameterization of the \( N_d \)-point inverse amplitude spectrum, as illustrated in Figure 4. As the size of the filter increases, the parameterization becomes more accurate, as expected from theory (Claerbout 1976). The notion of PEF as “decorrelator” is quite
Figure 3: "Ridges" image and Fourier transform synthesis. The correlation is both long-range and extremely complicated - quite like a meandering network of fluvial channels. Though the synthesized image has the same general character as the TI, not all of the structures are modeled, proving the inadequacy of the amplitude spectrum for modeling nonstationary, highly correlated images. The TI phase spectrum shows some ordering, so the random phase substitution was ill-advised.

Figure 4: Frequency response of “inverse PEF” (deconvolution) as a function of filter size. As expected, as the filter length increases, the approximation improves.
akin to decomposition by principal components (Castleman, 1996), where the number of principal components used in computation determines the degree of decorrelation.

The following is an outline of the PEF-based texture synthesis method.

1. Given training image \( t(x, y) \), estimate unknown PEF \( a(x, y) \) via least squares minimization:

   \[
   \min \| t \ast a \|^2
   \]

2. The residual \( r = t \ast a \) is approximately uncorrelated, with the same dimension as the TI, since we use an "internal" convolution algorithm (Claerbout, 1998a). It can be proved that \( a \) is a minimum phase filter, (Claerbout, 1976) so deconvolution (polynomial division) robustly and stably reconstructs \( t \) given \( r \).

   Generate a random residual \( r' \) with the same dimension as \( r \). To create the synthetic texture, simply deconvolve \( r' \) by \( a \):

   \[
   t_{\text{syn}} = r' / a
   \]

   where the " / " refers to polynomial division, our preferred method of deconvolution.

   Though the residual is uncorrelated, it does contain "phase" information. Deconvolution of a random image blindly spreads scaled copies of the impulse response of the inverse PEF across the output space. If the residual \( r \) is not sufficiently whitened, then the replacement of \( r \) with \( r' \) will lead to an ineffective representation of \( t \) by \( t_{\text{syn}} \).

   Figures 5 through 7 illustrate the PEF-based texture synthesis process. The left-hand panel shows the training image, the center panel shows the residual \( r = t \ast a \), and the right-hand panel shows the synthesized image, \( t_{\text{syn}} = r' / a \). A 10x10 PEF is used in each case. The blank areas in the residual panel correspond to regions where the PEF falls outside the bounds of the known data.

Figure 5: Smoothed random 2-D image and PEF-based texture synthesis result. The TI is quite simple (stationary, low correlation), so as expected, the synthesized image and the TI are almost indistinguishable. To the naked eye, the residual appears effectively white.
Figure 6: “Ridges” image and PEF-based texture synthesis result. Recall that the complicated connected features of this image were not completely synthesized by the Fourier transform method (Figure 3), of which the PEF method is an approximation. This synthesized image bears even less resemblance to the TI, exhibiting only a general southwest-to-northeast trend. The wavy, ridge-like features have many different dips, making them difficult to predict with a PEF, and with two point statistics in general. The same can be said for the ubiquitous hyperbolic features of reflection seismology. [texture/fft/ridges-pfensyn]

Figure 7: “Wood” image and PEF-based texture synthesis result. The synthesis result is pleasing. The PEF-based method preserves the general trend and relative scale length of the lineations in the TI. The correlation of the TI is relatively long-range, in that the lineations cross a large portion of the image, but the features are merely straight lines at one dip. [texture/fft/wood-pfensyn]
WHY USE THE PEF?

PEF-based texture synthesis can only achieve the results of the Fourier transform method (Figures 2 and 3) in the limit $N_a \to N_d$, which is unrealistic in practical situations, where $N_d$ is very large. Least squares estimation of the filter in this case is certainly costlier than three Fast Fourier transforms. On the other hand, if the filter size can be limited without compromising quality, which is the case for stationary, simply correlated images, then the PEF-based method is more flexible. Unlike the Fourier transform a PEF can be estimated easily when data are missing. Figure 8 shows that the PEF estimated from the incomplete data captures enough features of the data’s spectrum to make a fairly convincing texture synthesis result. The output of the PEF-based method can be of any size, while the output of a Fourier transform is generally constrained to be the same size as the input.

APPLICATIONS

PEF Estimation with incomplete data

Modern reservoir characterization efforts take a pragmatic view of collected data. Rather than wait for collection of the elusive “perfect” dataset, the desire is to incorporate a wide variety of possibly incomplete data types into a single inversion scheme (Caers and Journel, 1998). Often the only data available is spatially incomplete. Figure 8 shows the result of texture synthesis on training images with large void regions. As noted earlier, the blank areas in the center panels of the figure correspond to regions where the filter can’t fit without falling on one or more missing points. Each of the “in-bounds” data points contributes one equation to the LS estimation of the 100 or so filter coefficients. Even when well over half of the data points are removed from the training image this result shows that we can still safely estimate a filter and synthesis a believable texture.

2-D Stacked Seismic Section

Figure 9 shows the result of applying PEF texture synthesis to a 2-D stacked seismic section. The residual panel is interesting; notice uncollapsed diffraction hyperbolae, two highlighted fault planes, and also statics-like artifacts in the earlier times. PEF’s easily predict straight lines (plane waves) and sinusoids, but hyperbolae and discontinuities are quite another matter.

Matthias Schwab used the “plane wave prediction” property of the PEF in his Ph.D. thesis (Schwab, 1998) to create so-called “coherency cubes” from 3-D seismic data by nonstationary convolution with small PEF’s. Development of viable seismic coherency attributes merits considerable industrial interest, as evidenced by the concentration of related articles in the March, 1999 edition of The Leading Edge.
Figure 8: Comparison showing the effects of missing data on the PEF texture synthesis result, for two different “holes”. Although half or more of the equations are removed from the PEF estimation problem, the synthesized textures still capture the character of the training image. Fourier transforms are ill-defined on irregular coordinate systems, but the PEF makes an estimate of the known data’s spectrum regardless. [texture/fft/holes]
If a good velocity model is used, poststack migration should collapse these hyperbolas, so one measure of the fitness of a given velocity model could be the relative amount of residual energy in the data*PEF panel. Additionally, to the same end, this technique could be used to measure the relative amount of residual curvature in common reflection point (CRP) gathers, which are flattened when the correct migration velocity is used (Biondi [1997]). This preprocessing could be done quickly, for the necessary PEF’s are small.

![Figure 9: Stacked 2-D seismic section.](texture/fft/WGstack-pefsyn)

**Preconditioned Missing Data Infill**

To fill “holes” in collected data, we have the familiar SEP formulation (Claerbout [1998a]):

\[
Km - d \approx 0 \quad (5)
\]

\[
\epsilon Am \approx 0 \quad (6)
\]

\[5\] is the “data matching” goal, which states that the model \( m \) must match the known data \( d \), while \[6\] is the “model smoothness” goal, where \( A \) is an arbitrary roughening operator. To combat slow convergence, Claerbout (1998a) preconditions with the inverse of the convolutional operator \( A \) (multidimensional deconvolution). Provided that \( A \) is minimum phase or factorizable into the product of minimum phase filters (Sava et al. [1998]), the helix transform now permits stable multidimensional
deconvolution. Making the change of variables \( m = A^{-1}x \), we have the equivalent preconditioned problem:

\[
KA^{-1}x - d \approx 0
\]

(7)

\[
ex \approx 0
\]

(8)

The operator \( K \) effectively maps vectors in model space into a smaller-dimension “known data space”, so it has a nonempty nullspace. Missing points in model space are completely unconstrained by \( K \), so our choice of \( A \) wholly determines the behavior of the missing model points, i.e., their texture (Fomel et al., 1997). The PEF is a perfect choice for \( A \), as shown in Figure 10. The preconditioned, PEF-regularized result fills the hole quite believably after only 20 iterations, as opposed to the case where \( A = \nabla^2 \), which imposes an unrealistically smooth texture on the missing model points.

**DISCUSSION**

The goal of this paper is not to make slick surface textures for computer games. Nevertheless, as a tutorial device, texture synthesis using the PEF is valuable, since it concretely and intuitively illustrates in two dimensions some of the fundamental concepts of autoregression which are proved only in the one dimensional case (Claerbout, 1976). In fact, some of the results shown here and in Claerbout and Brown (1999) have recently been incorporated into Jon Claerbout’s textbook, *Geophysical Estimation by Example* (Claerbout, 1998a).

Both the Fourier transform and PEF-based texture synthesis operate under the assumption that the training image is sufficiently well characterized by amplitude spectrum alone. For some images (Figures 2, 5, and 7) the assumption holds, but for others (Figures 3, 6) it is obviously violated. Real digital images and earth phenomena alike often exhibit complex spatial correlation which are modelable only with multiple point templates (Caers and Journel, 1998; Malzbender and Spach, 1993). Additionally, I have ignored the interesting subjects of nonstationarity and spatial scale variance. By scale-variant, I mean that the characteristic scale of an image’s features is not constant with respect to spatial frequency. Many methods for characterizing scale-variant images appeal to the world of wavelets for a methodology known as *multiresolution analysis* (Simoncelli and Portilla, 1998; Heeger and Bergen, 1995; Strang and Nguyen, 1997). The notion of texture synthesis for nonstationary images is ill-defined, since it amounts to a random reordering of filters estimated on locally-stationary patches, followed by deconvolution on the corresponding patches.

When the training image has missing values, as in Figure 8, the PEF-based texture synthesis method performs favorably. As shown in the missing data interpolation example (Figure 10), the ability of the PEF to reliably estimate the data spectrum, even with missing data, makes it an ideal regularization operator. Figure 9 illustrates the fact that the PEF primarily predicts plane waves. I proposed using a PEF
Figure 10: Clockwise from top left: Data with hole, impulse response of “inverse PEF” (deconvolution of the PEF estimated from the data and a spike), data in-filled using $\nabla^2$ regularization, data in-filled using preconditioned PEF regularization. [texture/fft/ tree-hole-filled]
residual measure to determine the viability of a given migration velocity. In general, PEF estimation/convolution might have value as a preprocessing step for a variety of applications. For instance, a very small PEF (2 columns) has a relatively large residual in the presence of conflicting dips, and thus may help in determining local filter size or patch size.

ACKNOWLEDGEMENTS

All the results in this paper were generated quite easily using Sergey Fomel’s helix inversion library. All important programs have been included in the most recent release of SEPLib. The programs are conducive to curious experimentation, so I encourage the reader to use the makefile which accompanies the source code for this paper as a template.

REFERENCES

———, 1998b, Multi-dimensional recursive filtering via the helix: Geophysics, 63, 1532–1541.
Multi-dimensional Fourier transforms in the helical coordinate system

James Rickett and Antoine Guitton

ABSTRACT

For every two-dimensional system with helical boundary conditions, there is an isomorphic one-dimensional system. Therefore, the one-dimensional FFT of a 2-D function wrapped on a helix is equivalent to a 2-D FFT. We show that the Fourier dual of helical boundary conditions is helical boundary conditions but with axes transposed, and we explicitly link the wavenumber vector, \(k\), in a multi-dimensional system with the wavenumber of a helical 1-D FFT, \(k_h\). We illustrated the concepts with an example of multi-dimensional multiple prediction.

INTRODUCTION

If helical boundary conditions are imposed on a multi-dimensional system, an isomorphism exists between that system and an equivalent one-dimensional system. Previous authors, for example Claerbout (1998a), take advantage of this isomorphism to perform rapid multi-dimensional inverse filtering by recursion.

The Fourier analogue of convolution is multiplication: to convolve a 2-D signal with a 2-D filter, take their 2-D Fourier transforms, multiply them together and return to the original domain. The relationship between 1-D and 2-D convolution, FFT’s and the helix is illustrated in Figure 1. With helical boundary conditions, we can take advantage of the isomorphism described above, and perform multi-dimensional convolutions by wrapping multi-dimensional signals and filters onto a helix, taking their 1-D FFT’s, multiplying them together, and then returning to the original domain.

If we can use 1-D FFT’s to do 2-D convolutions, the isomorphism due to the helical boundary conditions must extend into the Fourier domain. In this paper, we explore the relationship between 1-D and multi-dimensional FFT’s in helical coordinate systems. Specifically we demonstrate the link between the wavenumber vector, \(k\), in a multi-dimensional system, and the wavenumber of a helical 1-D FFT, \(k_h\).

*e-mail: james@sep.stanford.edu, antoine@sep.stanford.edu*
Figure 1: Relationship between 1-D and 2-D convolution, FFT’s and the helical boundary conditions.

**THEORY**

For simplicity, throughout this section we refer to a two-dimensional sampled image, $b$; however, the beauty of the helical coordinate system is that everything can be trivially extended to an arbitrary number of dimensions.

We employ two equivalent subscripting schemes for referring to an element of the two-dimensional image, $b$. Firstly, with two subscripts, $b_{p_x, p_y}$ refers to the element that lies $p_x$ increments along the $x$-axis, and $p_y$ increments along the $y$-axis. Ranges of $p_x$ and $p_y$ are given by $0 \leq p_x < N_x$, and $0 \leq p_y < N_y$ respectively. Helical coordinates suggest an alternative subscripting scheme: We can use a single subscript, $p_h = p_x + p_y N_x$, such that $b_{p_x, p_y} = b_{p_h}$ and the range of $p_h$ is given by $0 \leq p_h < N_x N_y$. Moreover, if we impose helical boundary conditions, we can treat $b$ as a one-dimensional function of subscript $p_h$.

**Linking 1-D and 2-D FFT’s**

Taking the one-dimensional $Z$ transform of $b$ in the helical coordinate system gives

$$B(Z_h) = \sum_{p_h=0}^{N_x N_y-1} b_{p_h} Z_h^{p_h}. \quad (1)$$

Here, $Z_h$ represents the unit delay operator in the sampled (helical) coordinate system. The summation in equation (1) can be split into two components,

$$B(Z_h) = \sum_{p_y=0}^{N_y-1} \sum_{p_x=0}^{N_x-1} b_{p_x, p_y} Z_h^{p_x + p_y N_x} \quad (2)$$

$$= \sum_{p_y=0}^{N_y-1} \sum_{p_x=0}^{N_x-1} b_{p_x, p_y} Z_h^{p_x} Z_x^{p_y N_x}. \quad (3)$$

Ignoring boundary effects, a single unit delay in the helical coordinate system is equivalent to a single unit delay on the $x$-axis; similarly, but irrespective of boundary conditions, $N_x$ unit delays in the helical coordinate system are equivalent to a single delay on the $y$-axis. This leads to the following definitions of $Z_h$ and $Z_h^{N_x}$ in terms of delay operators, $Z_x$ and $Z_y$, or wavenumbers, $k_x$ and $k_y$:

$$Z_h \approx Z_x = e^{ik_x \Delta x},$$

$$Z_h^{N_x} = Z_y = e^{ik_y \Delta y}. \quad (4)$$

$$Z_h^{N_x} = Z_h^{N_y} = Z_{xy} = e^{ik_{xy} \Delta xy}. \quad (5)$$
where $\Delta x$ and $\Delta y$ define the grid-spacings along the $x$ and $y$-axis respectively.

Substituting equations (4) and (5) into equation (3) leaves

$$B(k_x, k_y) = B(Z_h) = B(Z_h) = \sum_{p_y=0}^{N_y-1} \sum_{p_x=0}^{N_x-1} b_{p_x, p_y} Z_{p_x}^{x} Z_{p_y}^{y}$$

$$= \sum_{p_y=0}^{N_y-1} \sum_{p_x=0}^{N_x-1} b_{p_x, p_y} e^{ik_x \Delta x p_x} e^{ik_y \Delta y p_y}. \quad (6)$$

Equation (7) implies that, if we ignore boundary effects, the one-dimensional FFT of $b(x, y)$ in helical coordinates is equivalent to its two-dimensional Fourier transform.

**Wavenumber in helical coordinates**

With the understanding that the 1-D FFT of a multi-dimensional signal in helical coordinates is equivalent to the 2-D FFT, a natural question to ask is: how does the helical wavenumber, $k_h$, relate to spatial wavenumbers, $k_x$ and $k_y$?

The helical delay operator, $Z_h$, is related to $k_h$ through the equation,

$$Z_h = e^{ik_h \Delta x}. \quad (8)$$

In the discrete frequency domain this becomes

$$Z_h = e^{iq_h \Delta k_h \Delta x}, \quad (9)$$

where $q_h$ is the integer frequency index that lies in the range, $0 \leq q_h < N_x N_y$. The uncertainty relationship, $\Delta k_h \Delta x = \frac{2\pi}{N_x N_y}$, allows this to be simplified still further, leaving

$$Z_h = e^{2\pi i \frac{q_h}{N_x N_y}}. \quad (10)$$

If we find a form of $q_h$ in terms of Fourier indices, $q_x$ and $q_y$, that can be plugged into equation (10) in order to satisfy equations (4) and (5), this will provide the link between $k_h$ and spatial wavenumbers, $k_x$ and $k_y$.

The idea that $x$-axis wavenumbers will have a higher frequency than $y$-axis wavenumbers, leads us to try a $q_h$ of the form,

$$q_h = N_y q_x + q_y. \quad (11)$$

Substituting this into equation (10) leads to

$$Z_h = e^{2\pi i \frac{(N_y q_x + q_y)}{N_x N_y}} \quad (12)$$

$$= e^{2\pi i \left( \frac{q_x}{N_x} + \frac{q_y}{N_y} \right)}. \quad (13)$$
Since $q_y$ is bounded by $N_y$, for large $N_x$ the second term in braces $\frac{q_y}{N_x N_y} \approx 0$, and this reduces to

$$Z_h \approx e^{2\pi i \frac{q_x}{N_x}} = Z_x,$$

(14)

which satisfies equation (4).

Substituting equation (11) into equation (10), and raising it to the power of $N_x$ leads to:

$$Z_h^{N_x} = e^{2\pi i \frac{(N_y q_x + q_y)}{N_y}} = e^{2\pi i \left(q_x + \frac{q_y}{N_y}\right)},$$

(15)

$$Z_h^{N_y} = e^{2\pi i \frac{q_y}{N_y}} = Z_y,$$

(16)

Since $q_x$ is an integer, $e^{2\pi i q_x} = 1$, and this reduces to

$$Z_h^{N_x} = e^{2\pi i \frac{q_y}{N_y}} = Z_y,$$

(17)

which satisfies equation (5).

Equation (11), therefore, provides the link we are looking for between $q_x$, $q_y$, and $q_h$. It is interesting to note that not only is there a one-to-one mapping between 1-D and 2-D Fourier components, but equation (11) describes helical boundaries in Fourier space: however, rather than wrapping around the $x$-axis as it does in physical space, the helix wraps around the $k_y$-axis in Fourier space (Figure 2). This provides the link that is missing in Figure 1 but shown in Figure 3.

Figure 2: Fourier dual of helical boundary conditions is also helical boundary conditions with axis of helix transposed.

Figure 3: Relationship between 1-D and 2-D convolution, FFT’s and the helix, illustrating the Fourier dual of helical boundary conditions.

As with helical coordinates in physical space, equation (11) can easily be inverted to yield

$$k_x = \Delta k_x q_x = \frac{2\pi}{N_x \Delta x} \left[ q_h \left\lfloor \frac{q_h}{N_y} \right\rfloor \right], \quad \text{and}$$

$$k_y = \Delta k_y q_y = \frac{2\pi}{N_y \Delta y} \left( q_h - N_y \left\lfloor \frac{q_h}{N_y} \right\rfloor \right)$$

(18)

(19)

where $[x]$ denotes the integer part of $x$. 
**Speed comparison**

For a two-dimensional dataset with dimensions, $N_x \times N_y$, the cost of a 1-D FFT in helical coordinates is proportional to

$$N_x N_y \log (N_x N_y).$$  \hfill (20)

For the same dataset, the cost of a 2-D FFT is

$$N_y (N_x \log N_x) + N_x (N_y \log N_y) = N_x N_y (\log N_x + \log N_y)$$

$$= N_x N_y \log (N_x N_y).$$  \hfill (21)

Therefore, the cost of a 1-D helical FFT of a 2-D dataset is exactly the same as the cost of an 2-D FFT of the same dataset. The link between the two leads to no computational advantages in the number of operations.

However, other differences may lead to computational savings. For example, a 2-D FFT with a power-of-two algorithm requires both $N_x$ and $N_y$ to be powers of two. However, the 1-D helical FFT requires just $N_x N_y$ to be a power of two, and so less zero-padding may be required.

The corollary, that a large 1-D FFT can be computed (with small inaccuracies) using a 2-D FFT algorithm, also leads to potential computational savings. Two-dimensional FFT’s are easier to code to run both in parallel and out-of-core than 1-D FFT’s, leading to significantly faster code and a lower memory requirement without the additional complexity of Singleton’s algorithm [Press et al. 1992].

**EXAMPLES**

Figure 4 compares the real part of the 2-D Fourier transform of a single spike with the equivalent real part after a 1-D FFT in helical boundary conditions. The Fourier transforms are centered, so that zero frequency is at the center of the plot. This has the effect that the artifacts that would appear at the vertical boundaries ($k_y = 0$) of the image are more visible since they appear at the center of the plot.

Figure 5 compares amplitude spectra for a broader band 2-D seismic VSP gather. Artifacts from the helical boundaries are very difficult to see on the spectra themselves, and the difference image is very low amplitude.

**Application to the multiple prediction**

Multiple prediction is the first step in the class of adaptive multiple suppression methods ([Verschuur et al. 1992]). In a laterally homogeneous earth, [Kelamis and Verschuur 2000] show that surface-related multiples can be predicted by taking the
Figure 4: Comparison of real part of 2-D spectra: (a) input spike (single frequency), (b) real part of 2-D FFT, (c) real part of 1-D helical FFT, and (d) difference between (b) and (c) clipped to same level.
Figure 5: Comparison of 2-D amplitude spectra: (a) input 2-D VSP gather, (b) amplitude spectrum from 2-D FFT, (c) amplitude spectrum from 1-D helical FFT, and (d) difference between (b) and (c) clipped to same level.
multi-dimensional auto-convolution of a common midpoint (CMP) gather. This auto-convolution reduces to a multiplication in the $f$-$k$ domain, and so it can be performed rapidly with multi-dimensional FFT’s.

Since multi-dimensional FFT’s can be computed with a one-dimensional Fourier transform in helical coordinates, we can predict multiples by wrapping a CMP gather onto a helix, taking its 1-D FFT, squaring the result, and returning to the original domain.

We tested this algorithm on a single CMP from the synthetic BP multiple dataset [Clapp 1999]. Figure 6 displays the multiple prediction result using the helical coordinate system and only a single one-dimensional FFT. Theoretically, only first-order multiples should have correct relative amplitudes, and the source wavelet appears twice in the multiple prediction. However, the kinematics of all multiples are almost exact, even for higher-order multiples below 5 s two-way traveltime.

![Figure 6: The left panel shows the multiple model obtained with the helix and a 1-D FFT. The right panel shows the input CMP gather with the offset axis reversed to facilitate the comparison. Some wrap-around effects appear at the top of the multiple model.](helft/mult/ BP2)
CONCLUSION

We have explicitly found the relationship between multi-dimensional FFT’s and 1-D FFT’s on a helix, linking the wavenumber vector, $\mathbf{k}$, in a multi-dimensional system with the wavenumber of a helical 1-D FFT, $k_h$. Specifically, the Fourier dual of helical boundary conditions is helical boundary conditions but with axes transposed. We have illustrated the concepts with an example of multi-dimensional multiple prediction.

REFERENCES

Passive seismic imaging applied to synthetic data

James Rickett and Jon Claerbout

ABSTRACT

It can be shown that for a 1-D Earth model illuminated by random plane waves from below, the cross-correlation of noise traces recorded at two points on the surface is the same as what would be recorded if one location contained a shot and the other a receiver. If this is true for real data, it could provide a way of building ‘pseudo-reflection seismograms’ from background noise, which could then be processed and used for imaging. This conjecture is tested on synthetic data from simple 1-D and point diffractor models, and in all cases, the kinematics of observed events appear to be correct. The signal to noise ratio was found to increase as $\sqrt{n}$, where $n$ is the length of the time series. The number of incident plane waves does not directly affect the signal to noise ratio; however, each plane wave contributes only its own slowness to the common shot domain, so that if complete hyperbolas are to be imaged then upcoming waves must be incident from all angles.

INTRODUCTION

Conventional seismic reflection methodology relies on having a source of seismic waves at the surface and studying the reflections from impedance contrasts in the earth. Ambient noise present in the subsurface is also reflected by impedance contrasts in the same way. Therefore, do we need the source or can we determine earth structure simply by listening to the background noise for a long enough time?

With 4-D seismic surveys aimed at monitoring fluid movements becoming more and more popular, a number of oil fields are having geophone arrays laid out permanently to reduce acquisition costs. These geophones will typically only record while a survey is being shot, and for most of their life they will be turned off. However, if the geophones are left recording while surveys are not being shot, we hypothesize the information contained in the background seismic energy could be used for imaging the subsurface between main surveys.

For this to be realized a technique has to be developed that: firstly, is able to extract the useful information from the background noise; and secondly, is able to do this quickly enough, ideally in real-time, so the huge amounts of raw data that would be recorded would not have to be stored.

*e-mail: james@sep.stanford.edu, jon@sep.stanford.edu
With this in mind, a technique for creating ‘pseudo-cmp gathers’ out of background noise traces using cross-correlation is explored and tested on a variety of synthetic models.

**CONJECTURE**

By cross-correlating noise traces recorded at two locations on the surface, we can construct the wavefield that would be recorded at one of the locations if there was a source at the other. In this way we can create ‘pseudo-reflection seismograms’ that include such effects as NMO and DMO.

**PROOF FOR A 1-D EARTH**

The proof of this conjecture for a one dimensional earth is given as a problem set in [Claerbout (1976)](https://example.com). The outline of the derivation that follows uses the $Z$ transform approach developed there, where $Z$ is the unit delay operator $e^{-i\omega \Delta t}$.

Consider a plane-layered Earth model with the reflection seismology geometry shown in Figure 1. If the system is lossless then the energy flux through the top layer has to be equal to the flux through the half-space below. Therefore,

$$Y_1 \left\{ R \left( \frac{1}{Z} \right) R(Z) - \left[ 1 + R \left( \frac{1}{Z} \right) \right] \left[ 1 + R(Z) \right] \right\} = -Y_k E \left( \frac{1}{Z} \right) E(Z) \quad (1)$$

where $Y_1$ is the impedance of the top layer and $Y_k$ is the impedance of the half-space, or

$$1 + R \left( \frac{1}{Z} \right) + R(Z) = \frac{Y_k}{Y_1} E \left( \frac{1}{Z} \right) E(Z) \quad (2)$$

Figure 1: Reflection seismology geometry - Claerbout (1979).
Comparing the reflection seismology geometry with the earthquake seismology geometry shown in Figure 2 gives $E(Z) = X(Z)$ by reciprocity. Therefore

$$1 + R\left(\frac{1}{Z}\right) + R(Z) = (\text{constant}) \ X\left(\frac{1}{Z}\right) X(Z) \tag{3}$$

Since $R(Z) = 0$ for times less than zero, this means the positive time part of the ‘earthquake’ seismogram’s auto-correlation function equals the reflection seismogram.

This theorem can be extended to a two dimensional plane-layered Earth by considering slant stacks (Claerbout, 1985).

Figure 2: Earthquake seismology geometry - from Claerbout (1979). All the waves (1, X and P) could be multiplied by the same random signal, and this would not effect their spectra or auto-correlation functions.

SYNTHETIC DATA

Cole (1995) tested this conjecture on both synthetic and real data. However his field data was very noisy, and he did not draw any solid conclusions.

With this as a starting point, however, I continued modeling a single reflection, using a program based on the flow

\[
\text{loop over each plane wave \{ }
\text{calculate a random slowness, } p \\
\text{calculate the time delay due to a reflection, } \Delta t \\
\text{loop over each frequency, } \omega \{ }
\text{calculate a random amplitude} \\
\text{loop over each spatial location, } x \{ }
\text{multiply each frequency by a factor } (1 + re^{i\omega \Delta t}) e^{i\omega px} \\
\text{\}} \\
\text{\}\}
\]
Having produced synthetics, it was then possible to go ahead and cross-correlate traces to try and create pseudo-reflection seismograms. Figure 3 is a pseudo-shot gather generated by cross-correlating one trace with every other. The center panel shows how the clarity of the signal was improved by applying a $\sqrt{-\omega}$ filter. The black line which has been overlain in the left panel corresponds to the expected hyperbola which would be observed in a real shot gather, offset by 0.05 s so it does not obscure the data. Therefore the kinematics in this case appear to be consistent with the conjecture.

Figure 3: Pseudo-shot gather over model with single horizontal layer and 200 incoming plane waves. The left panel is raw cross-correlations, the center panel has a half differentiation filter applied and the right panel is labeled with the correct kinematics shifted by 0.05 s.

Limited angular bandwidth

Figure 4 shows how the hyperbolas build up from the individual plane waves. In the left panel there is only one plane wave present, instead of 200 as in Figure 3. The trace at zero spatial offset is the autocorrelation function, and it has two spike as expected: one at zero lag and one corresponding to the reflected event. Other traces, corresponding to cross-correlations, have the two same spikes, but these are offset in time due to the different receiver positions and the apparent velocity of the incident wave. In the right panel of Figure 4 there are 5 plane waves incident, and a hyperbola is beginning to form. Only one part of each plane wave adds coherently with the others, the rest of the energy is smeared out.

The corollary of this is that, in order to see a slope in the pseudo-reflection seismograms, we need incident plane waves with the relevant slowness. This will become
an issue when looking at real data where the direction of incoming waves will not be spatially white.

Figure 4: Pseudo-shot gather over model with single horizontal layer. Left panel has one incoming plane wave and right panel has five. 

More complex Earth models - point diffractors and multiple layers

The horizontal bed code shown earlier was easy to adapt to test slightly more complicated models. The left panel of Figure 5 shows a pseudo-shot gather with a point diffractor in the subsurface. Again the kinematics are correct, as they are in the zero-offset section of the same model shown in the right panel of Figure 5. Notice the offset center of the hyperbola in the pseudo-shot gather.

Figure 5: Model with single point diffractor. Left panel shows common-shot gather, and right panel shows zero-offset section.
With a model of two horizontal layers, the results (Figure 6) seem to contain a spurious event at the time of the inter-bed multiple. This is not the multiple itself but comes from the correlation between the first layer and the second layer. If the inter-bed multiple was included in the modeling process, it would arrive at the same time but with opposite amplitude and so cancel this event out. Therefore this event is an artifact of the modeling technique and does not contradict the conjecture.

Figure 6: Common shot gather for a model with two horizontal layers.

Noise suppression

So far the reflection coefficients have been very high (0.5) and still the images have been relatively noisy. If this technique is to be used for looking at real targets, reflection coefficient of an order of magnitude smaller will have to be imaged.

Tests showed that the signal to noise ratio decreased as the $\sqrt{n}$, where $n$ is the number of time samples, and by increasing the length of the time series used I was able to clearly image reflection coefficients of 0.05, as shown in Figure 7.

A similar study, but this time comparing signal to noise ratio with the number of incident plane waves, was also conducted. Interestingly, it showed that more plane waves did not increase the amplitude of the observed signal to noise; however, more plane waves do improve the shape and definition of the hyperbola.

CONCLUSIONS AND FURTHER WORK

So far this method has worked well on a variety of simple Earth models that have been illuminated by plane waves coming from all angles in the subsurface.

Before the method is tested on real data, the following points need to be addressed:
Figure 7: Common shot gathers for a model with a single horizontal layer with reflection coefficient 0.05. Left panel used 65,000 points in the time series, the center panel about 130,000 and the right panel about 260,000.

1. A method of spatially pre-whitening upcoming waves has to be developed, so that all dips can be imaged even if the angular distribution of upcoming waves is uneven.

2. The conjecture should be tested with full wave-form modeling on more complicated Earth models (multiple dipping beds and velocities which vary both laterally and vertically, for example).

3. In order to avoid storing huge amounts of data, the cross-correlations should be done in the field in real time. Implementation of this will be difficult.

REFERENCES

When is anti-aliasing needed in Kirchhoff migration?

Dimitri Bevc and David E. Lumley

ABSTRACT
We present criteria to determine when numerical integration of seismic data will incur operator aliasing. Although there are many ways to handle operator aliasing, they add expense to the computational task. This is especially true in three dimensions. A two-dimensional Kirchhoff migration example illustrates that the image zone of interest may not always require anti-aliasing and that considerable cost may be spared by not incorporating it.

INTRODUCTION
In this paper we establish some rules of thumb as to when anti-aliasing is required in Kirchhoff migration. The same criteria are applicable to other processes such as DMO, velocity analysis, and wave-equation datuming.

There are many methods of handling operator aliasing. Gray (1992) presented a method which involves low-pass filtering data traces with a variety of pass bands and then selecting input data from these sets of traces so that operator aliasing does not occur. Spatial trace interpolation is another method of dealing with the operator aliasing problem (Yilmaz, 1987). A draw back of the latter two methods is increased data volumes. Methods which limit the dip or aperture of the operator reduce aliasing without increasing the data volume, but at the expense of losing high-angle and wide-aperture information. An attractive and computationally efficient method of handling operator aliasing has been implemented by Claerbout (1992). His dip-dependant triangular weighting method does not require multiple copies of the data to be kept in memory since the weights are generated and applied quickly on-the-fly.

Claerbout’s triangular weighting method has been demonstrated to be efficient for 2-D (Bevc and Claerbout, 1992, 1993) and 3-D (Lumley, 1993; Lumley et al., 1994) Kirchhoff time and depth migration. It has also been successfully adapted to DMO and wave-equation datuming operators (Blondel, 1993; Bevc, 1992). Even though the triangular weighting method is very efficient, it still involves an extra computational cost. When the anti-aliased algorithm is implemented on the Connection Machine in FORTRAN 90, calls to an indirect addressing subroutine are required to extract

*e-mail: james@sep.stanford.edu, jon@sep.stanford.edu
data points from individual traces for summing into output locations. These calls turn out to be a bottleneck. In order to perform an anti-aliased migration with linear interpolation, six calls to the indirect addressing subroutine are required for each input trace location. For a 3-D migration, the indirect addressing is substantial.

Because this anti-aliasing is currently expensive on the CM5, we are motivated to determine when we can get away with not using it. While doing away with anti-aliasing is generally not a good idea, there are situations in which we may be able to live without it. For example, if we are running trial migrations to determine velocity models we may concentrate our efforts on portions of the data where operator aliasing is not a factor.

After developing criteria which link frequency and dip content of seismic data, we migrate a 2-D salt dome data set with and without anti-aliasing. The examples illustrate the effects of operator aliasing, how it can be ameliorated by aperture limitation and triangle weighted migration, and when anti-aliasing is unnecessary.

**OPERATOR ALIASING**

Operator aliasing most often occurs when operator moveout across adjacent traces exceeds the time sampling rate. Cycle skips can occur when the operator is aliased. For a moveout curve with slope $dt/dx$, and data with a spatial Nyquist frequency of $k_n$, temporal frequencies above

$$\omega = \frac{k_n}{dt/dx}$$

are aliased. In terms of the mesh spacing $\Delta x$ and operator slope $dt/dx$, operator aliasing will occur for all frequencies above $f_{op}$, where $f_{op}$ is given by:

$$f_{op} = \frac{1}{2(\frac{dt}{dx})\Delta x}. \quad (1)$$

Defining the maximum stepout as $p = \delta x/\delta t$, the highest dip frequency in the data is given by

$$f_d = \frac{1}{2p\Delta x}. \quad (2)$$

When the stepout is captured by the mesh spacing, $\delta x = \Delta x$, and $\delta t = \Delta t$, the highest unaliased dip frequency is equal to the Nyquist frequency $f_n = 1/2\Delta t$. In areas of economic interest, steep dips are often present in the data and $f_d > f_n$.

Anti-aliasing is called for when the frequency content of the data, $f_s$, falls between

$$f_{op} \leq f_s \leq f_d. \quad (3)$$

This situation is illustrated in Figure 1.
GULF OF MEXICO SALT DOME EXAMPLES

In this section, we migrate a data set from the Gulf of Mexico to illustrate when anti-aliasing is not required. Migration is performed using an implementation of Claerbout’s triangle weighted anti-aliasing scheme. Corresponding examples of standard Kirchhoff migration with and without angle limitation, are computed for comparison.

Anti-aliased migration

The diffractions from the salt flanks are obvious in the near-offset section (Figure 2a) and can be seen to be spatially aliased. The data exhibit some overall speckling because of temporal aliasing. This temporal aliasing is due to recording or processing which was performed on the original data before it arrived at SEP. The anti-aliased migration is displayed in Figure 2b. The salt flanks are nicely imaged and there is no evidence of operator aliasing. There are some artifacts due to the temporal aliasing; temporal aliasing is a different phenomena than operator aliasing and cannot be ameliorated by modifying the operator.

Aliased migration

In Figure 3a, the data are migrated without triangular weighting. The effect of operator aliasing is most evident in the seafloor arrivals. We see precursors to the actual event. The top of the salt dome (earlier than \( t = 0.6 \) s) is poorly imaged and there is more overall speckling than in Figure 2b, suggesting that the effects of data aliasing are compounded by operator aliasing. Other prominent operator aliasing artifacts are seen at about midpoints 13000 to 14000 and time 0.6 s to 0.9 s as cross-cutting dipping events. Figure 4 is a comparison of the anti-aliased migration and the aliased migration (Figure 4a is a closeup of Figure 2b, and Figure 4b is a closeup of Figure 3a). The seafloor precursor artifacts before \( t = 0.18 \) s and the cross-cutting dipping event artifacts are marked.

The operator aliasing has been somewhat contained by limiting the migration aperture to 45° in Figure 3b; however, the seafloor event still has a precursor, the
Figure 2: (a) Near-offset section from the Gulf of Mexico. (b) Kirchhoff migration with triangle weighted anti-aliasing.
top of the salt dome is still poorly imaged, and there is more coherent noise than in Figure 2b.

**Anti-aliasing is not needed to image the salt flank**

The interesting thing about the images migrated without anti-aliasing is that in all cases the salt flank is nicely imaged at late times. This is because in this region the migration velocity is fast and the operator does not have much dip, so that $f_{op} > f_s$ and operator aliasing is not a problem. At earlier times, the migration velocity is lower and the operator has significant dip so that $f_{op} < f_s$ and operator aliasing is a problem.

**Migration with low velocity**

In Figure 5 the data have been deliberately migrated with an unrealistic low velocity of 1000 m/s in order to confirm that operator aliasing will occur when the criteria of equation 3 are met. In these examples $f_{op} < f_s$ so that the operator is aliased. The diffraction from the salt flank is much better imaged in the anti-aliased migration (Figure 5a) than in the aliased migration (Figure 5b). All of the same operator aliasing artifacts that were visible at early times in Figure 2 are still present, but now the events at late times also suffer. The right hand diffraction in Figure 5b is weaker and less continuous than the anti-aliased diffraction in Figure 5a. Some of the diffractions on the left side of Figure 5b are completely lost in the aliased migration.

**CONCLUSIONS**

We have presented a simple inequality which can be used to determine whether operator aliasing is a factor in Kirchhoff migration. The same criteria can be used for DMO, velocity analysis, wave-equation datuming or any other integral operator which is applied to seismic data. The salt dome example illustrates that sometimes some portions of a data set may be sampled adequately, so that operator aliasing is not a problem. If these are the regions of interest, computational effort and time can be reduced by not undertaking the added expense of anti-aliasing.

**ACKNOWLEDGEMENTS**

The salt dome data were graciously provided by Halliburton Geophysical Services. We thank Mihai Popovici for obtaining the data and making it available to us.
Figure 3: (a) Kirchhoff migration without any aperture limitation or anti-aliasing. The effect of operator aliasing is noticeable at the seafloor where migration velocity is slow and where there is significant operator dip. At later times, the migration velocity is fast and there is not much operator dip, so there is no operator aliasing. (b) Kirchhoff migration of Gulf of Mexico data with 45° aperture limitation. Limiting the aperture reduces some, but not all, of the operator aliasing at the seafloor.
Figure 4: Close up of (a) the anti-aliased migration, and (b) the standard Kirchhoff migration without anti-aliasing. The seafloor operator aliasing artifacts and the dipping artifacts are marked.
Figure 5: Kirchhoff migration of Gulf of Mexico data with an unrealistically low velocity: (a) with anti-aliasing, and (b) without anti-aliasing.
REFERENCES


Imaging complex structures with first-arrival traveltimes

Dimitri Bevc

ABSTRACT

I present a layer-stripping Kirchhoff migration algorithm which is capable of obtaining accurate images of complex structures by downward continuing the data and imaging from a lower datum. I use eikonal traveltimes in a Kirchhoff datuming algorithm for the downward continuation. After downward continuation, I perform Kirchhoff migration. The method alternates steps of datuming and imaging. Because traveltimes are computed for each step, the adverse effects of caustics, headwaves, and multiple arrivals do not develop. In principal, this method only requires the same number of traveltime calculations as a standard migration. Tests on the Marmousi data set produce excellent results.

INTRODUCTION

Kirchhoff migration is generally accepted to be the most efficient method of imaging 2-D and 3-D prestack seismic data. The Marmousi synthetic data set (Versteeg, 1994) has been a popular testbed for migration algorithms and many researchers have discovered that Kirchhoff algorithms using first-arrival traveltimes do a poor job of imaging the target zone (Audebert et al., 1994; Gray and May, 1994; Geoltrain and Brac, 1993). Even methods which calculate most energetic arrivals and estimate amplitude and phase do not always result in images which compare favorably with finite-difference shot-profile migration.

In their 1993 Geophysics article, Geoltrain and Brac ask the question “Can we image complex structures with first-arrival traveltimes?” They conclude that they cannot, and that they should either ray trace to find the most energetic arrivals, or calculate multiple-arrival Green’s functions. Nichols (1994) calculates band-limited Green’s functions to estimate the most energetic arrivals. He estimates not only traveltimes, but also amplitude and phase. My approach is simpler; by breaking up the complex velocity structure, I am able to calculate traveltimes in velocity models where finite-differencing the eikonal equation is valid. This results in images comparable to those obtained by Nichols’ method and by shot-profile migration at a reduced computational cost.

Like most of the other researchers in the field, I test my method on the ever-popular Marmousi synthetic.

*e-mail: dimitri@sep.stanford.edu*
TRAVELTIME CALCULATION

Green’s functions based on first-arrival traveltime calculation methods result in poor images in structurally complex areas. Several reasons have been given for this failure:

- The high-frequency approximation of ray and eikonal methods breaks down in complex velocity models. In rapidly varying velocity models, different frequency components of the wavefield propagate at different velocities, so summation trajectories based on only the high-frequency components may not capture the desired events.

- When high velocity zones are present, the first-arrivals may be non-energetic headwaves.

- As energy propagates in complex models, raypaths tend to eventually cross. This causes phase shifts and triplications. First-arrival traveltimes follow the fastest branch of the triplication bow-tie, which is also the low energy branch.

Complexity of velocity models and validity of high-frequency approximations can be defined in various ways. The larger the velocity model, the more variation there will tend to be and the more opportunity there will be for things to go wrong: There will be more opportunity for frequency components to separate, for headwaves to develop, and for triplications to occur.

The Marmousi velocity model (Figure 1) results in complex propagation paths where late energetic arrivals are not fit well by first-arrival finite-difference traveltimes. In Figure 2, an acoustic modeling program was used to generate snapshots of the wavefield from two surface source locations in the Marmousi model. The corresponding contours of finite-difference traveltime at 1.05 s have been overlain. Clearly, the finite-difference traveltime contours do not always correspond to energetic portions of the wavefield. If these traveltimes were used for migration, the resulting image would suffer because parts of the summation trajectories would not correspond to energetic arrivals. By contrast, snapshots for the same source locations at an earlier time of 0.6 s (Figures 3a and 3b) show that the finite-difference traveltime curves overlay the high energy portions of the wavefields nicely. This is because there has not been enough time for adverse propagation effects to fully develop. Figures 3c and 3d are generated by starting the acoustic modeling and the finite-difference traveltime calculation from a depth of 1500 m. The 0.3 s contours correspond nicely to the high energy portions of the wavefields. There is some deviation in the shallow part of Figure 3a, but for the most part the finite-difference traveltime contour fits the bulk of the acoustic energy pretty well. Overall, the contours in Figure 3 have not pulled away from the wavefield as they have in Figure 2.

These properties of first-arrival traveltimes, and the observation that Kirchhoff migration using first-arrival traveltimes usually produces an acceptable image of the upper 1500 m of the Marmousi synthetic, motivated the development of the layer-stripping method described here.
Figure 1: The Marmousi velocity model.

Figure 2: The result of acoustic wavefield modeling for the Marmousi model overlain by contours of finite-difference travelt ime. Snapshots (a) and (b) are taken at a time of 1.05 s for two different source locations.
Figure 3: The result of acoustic wavefield modeling for the Marmousi model overlain by contours of finite-difference traveltime. Snapshots (a) and (b) are taken at a time of 0.6 s for the same two source locations depicted in Figure 2. Snapshots (c) and (d) are taken at a time of 0.3 s with sources at a depth of 1500 m and the same lateral positions as in (a) and (b).
Throughout this study, I use a finite difference solution to the eikonal equation to generate traveltimes (van Trier and Symes, 1991). I use my Kirchhoff datuming algorithm (Bevc, 1993), and an industry standard Kirchhoff migration code to generate the results.

LAYER-STRIPPING KIRCHHOFF MIGRATION

The layer-stripping migration method can be thought of as a hybrid algorithm that incorporates some of the advantages of shot-profile migration with the efficiency of Kirchhoff migration. This is illustrated by Figures 4 through 6.

In shot-profile migration, shots and geophones are alternately downward continued through each depth level. Figure 4 illustrates this for one line of shots or geophones. It is evident that there are many propagation paths from the surface to the image point, therefore multiple arrivals are handled. The computation is performed for all frequencies.

By contrast, first-arrival Kirchhoff migration is performed by summing data over trajectories defined by the propagation paths illustrated in Figure 5. There is only one path linking each surface position to the image point, therefore multiple arrivals are not handled. Although crossing paths are not illustrated here, they can occur.

The layer-stripping method is illustrated in Figure 6. It is a combination of Kirchhoff wave-equation datuming (Berryhill, 1979, 1984) and Kirchhoff migration. The computation proceeds as follows:

1. Migrate from surface to some depth level $z_1$.
2. Downward continue to depth level $z_1$.
3. Migrate from depth level $z_1$.
4. Downward continue to depth level $z_2$.
5. Migrate from depth level $z_2$.
6. etc...

The depth step of downward continuation is much larger than the $\Delta z$ used in shot-profile migration. Since there are multiple paths from the surface to the image point, multiple arrivals are handled. Because the paths are shorter than in Figure 5, first-arrival traveltimes are more likely to be valid.

MARMOUSI EXAMPLE

Figure 7 is the result of an industry standard Kirchhoff migration of the Marmousi synthetic using eikonal traveltimes. The upper portion is well imaged; however, the
Figure 4: Propagation paths from the surface to an image point for shot profile migration.

Figure 5: Propagation paths from the surface to an image point for Kirchhoff downward continuation.

Figure 6: Propagation paths from the surface to a depth point for layer-stripping Kirchhoff downward continuation.
anticlinal structure below 2200 m and the target zone at a lateral position of about 6500 m and depth of 2500 m is not imaged.

Figure 8 is generated by downward continuing the data to a depth of 1500 m in one datuming step. The downward continued data are then migrated and combined with the previous image of the upper 2000 m. The anticlinal structure and the target are now clearly imaged.

Continuing the data to 1500 m in three steps of 500 m each, results in an even crisper image of the anticline and the target (Figure 9). In both of these images, the events which unconformably define the top of the anticline, the anticline events themselves, and the target events, are clearly imaged.

In Figure 10, I compare the images in the vicinity of the target zone to the velocity model and a filtered reflectivity model which represents the desired image. Both images compare favorably to the desired reflectivity. The image obtained by downward continuing the data in three steps of 500 m is superior since the events display better lateral continuity and the image is clearer. This is because the traveltimes calculated for each of the 500 m steps are simpler and better behaved than the traveltimes calculated for one step of 1500 m.

**CONCLUSION**

I obtain images comparable to shot-profile migration results by combining wave-equation datuming and Kirchhoff migration into a layer-stripping migration method. In this case, eikonal traveltimes produce satisfactory images because the velocity
Figure 8: Migrated image using traveltimes calculated from the surface, and travel-times calculated from a depth of 1500 m. The lower part of the image was obtained by migrating data which was redatumed to a depth of 1500 m in one step of downward continuation.

Figure 9: Migrated image using traveltimes calculated from the surface, and travel-times calculated from a depth of 1500 m. The lower part of the image was obtained by migrating data which was redatumed to a depth of 1500 m in three steps of 500 m each.
Figure 10: Comparison of the velocity, the reflectivity, and the images in the target zone.
model is subdivided and traveltimes are calculated under conditions where finite-differencing the eikonal equation is valid. By dividing the imaging problem in this way, the traveltimes are better behaved and some multiple arrivals are accounted for.

REFERENCES


Evaluating the Stolt-stretch parameter

Sergey Fomel* and Louis Vaillant†

ABSTRACT

The Stolt migration extension to a variable velocity case describes the velocity heterogeneity with a constant parameter, which is related to the stretch transformation of the time axis. We exploit a connection between modified dispersion relations and nonhyperbolic traveltime approximations to derive an explicit expression for the stretch parameter. This analytical expression allows one to achieve the highest possible accuracy within the Stolt stretch approximation. Using a real data example, we demonstrate an application of the explicit Stolt stretch formula for an optimal partitioning of the migration velocity in the method of cascaded migrations.

INTRODUCTION

Although Stolt migration is regarded as the fastest of all the known seismic migration algorithms, it has a limited applicability because of the intrinsic constant velocity assumption. The time-stretching trick proposed in Stolt’s classic paper (Stolt 1978) provides an approximate extension of the method to a variable velocity case. Implicitly, Stolt stretch transforms reflection traveltime curves to fit an approximate constant velocity pattern (Levin 1983, 1985; Claerbout 1985). In other words, the wave equation with variable velocity is transformed by a particular stretch of the time axis to an approximate differential equation with constant coefficients. The two constant coefficients are an arbitrarily chosen frame velocity and a special non-dimensional parameter ($W$ in Stolt’s original notation). In the constant velocity case $W$ is equal to 1, and the transformed equation coincides with the exact constant velocity wave equation. In variable velocity media, $W$ is generally assumed to lie between 0 and 1. As shown by Larner and Beasley (1987), the cascaded $f$-$k$ migration approach can move the value of $W$ for each migration in a cascade closer to 1, thus increasing the accuracy of the Stolt stretch approximation.

The $W$ factor is defined by Stolt (1978) as an approximate average of a complicated function, which depends on both time and space coordinates and cannot be computed...
directly. Therefore, in practice, the estimation of $W$ is always replaced by a heuristic guess. That is why Levin (1983) jokingly called the $W$ parameter “infamous”, and Larner and Beasley (1987) called it “esoteric.”

In this paper, we use an analytic technique to evaluate the Stolt stretch parameter explicitly. The main idea is to constrain this parameter by fitting the Stolt-stretch traveltime function to the exact one. It turns out that in the isotropic case, the $W$ parameter is connected to the “parameter of heterogeneity” (Malovichko, 1978; Sword, 1987; Castle, 1988; de Bazelaire, 1988). The definition of heterogeneity is modified for the case of an anisotropic (transversally isotropic) media.

We demonstrate an application of the Stolt stretch analytical expression on a real data example from the North Sea. The velocity profile is optimally partitioned for the method of cascaded migration, which allows us to image steeply dipping reflectors at the accuracy comparable to that of the phase-shift method but at a much smaller cost.

Although Stolt migration is not currently at the forefront of geophysical research, it is still widely used in practice (Yilmaz, 2001; Yilmaz et al., 2001) and keeps recurring in different contexts. Popovici et al. (1996) propose a new interpolation scheme for improving the practical accuracy of the method. Sava (2000) uses a variation of Stolt migration - Stolt residual migration (Stolt, 1996) - in the context of wave-equation migration velocity analysis.

The growth in computer speed does not automatically make fast algorithms obsolete, because the amount of processed data tends to grow at the same rate or even faster. The researchers working in the field of seismic imaging are often interested in the following questions: What is the fastest possible migration algorithm? How accurate can it get? Stolt migration answers the first question. The answer to the second question is developed in this paper.

**STOLT STRETCH THEORY REVIEW**

In order to simplify the references, we start with definitions of the Stolt migration method. The reader familiar with the Stolt stretch theory can skip this section and go on to new theoretical results in the next section.

The basic migration theory reduces post-stack migration to a two-stage process. The first stage is a downward continuation of the wavefield in depth $z$ based on the wave equation

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial z^2} = \frac{1}{v^2(x,z)} \frac{\partial^2 P}{\partial t^2}.$$  \hspace{1cm} (1)

The second stage is the imaging condition $t = 0$ (here the velocity $v$ is twice as small as the actual wave velocity). Stolt time migration performs both stages in one step,
applying the frequency-domain operator

$$\tilde{P}_0 (k_x, \omega_0) = \tilde{P}_v (k_x, \omega_v (k, \omega_0)) \left| \frac{d\omega_v (k, \omega_0)}{d\omega_0} \right| ,$$

(2)

where

$$\tilde{P}_v (k_x, \omega_v) = \iint P_v (x, t_v) \exp (i\omega_v t_v - ik_x x) \, dt_v \, dx ,$$

$$\tilde{P}_0 (k_x, \omega_0) = \iint P_0 (x, t_0) \exp (i\omega_0 t_0 - ik_x x) \, dt_v \, dx ,$$

$P_0 (x, t_0)$ stands for the initial zero-offset (stacked) seismic section defined on the surface $z = 0$, $P_v (x, t_v)$ is the time-migrated section, and $t_v$ is the vertical traveltime

$$t_v = \int_0^z \frac{dz'}{v(x, z')} .$$

(3)

The function $\omega_v (k, \omega_0)$ in (2) corresponds to the dispersion relation of the wave equation (6) and in the constant velocity case has the explicit expression

$$\omega_v (k, \omega_0) = \text{sign} (\omega_0) \sqrt{\omega_0^2 - v^2 k^2} .$$

(4)

The choice of the sign in equation (4) is essential for distinguishing between upgoing and downgoing waves. The upgoing part of the wavefield is the one used in migration.

The case of a varying velocity complicates the frequency-domain algorithm and therefore requires special consideration. Stolt (1978) suggested the following change of the time variable (referred to in the literature as Stolt stretch):

$$s(t) = \left( \frac{2}{v_0^2} \right)^{1/2} \left( \int_0^t \eta d\tau \right)^1 ,$$

(5)

where $v_0$ is an arbitrarily chosen constant velocity, and $\eta$ is a function defined by the parametric expressions

$$\eta(\zeta) = \int_0^\zeta v(x, z) \, dz , \tau(\zeta) = \int_0^\zeta \frac{dz}{v(x, z)} .$$

(6)

Applying equation (5), we can connect seismic time migration to the transformed wave equation

$$\frac{\partial^2 P}{\partial x^2} + W \frac{\partial^2 P}{\partial \tilde{z}^2} + 2 \left( 1 - W \right) \frac{\partial^2 P}{\partial \tilde{z} \partial \tilde{t}} = \left( 2 - W \right) \frac{\partial^2 P}{\partial \tilde{t}^2} .$$

(7)

The variables $\tilde{z}$ and $\tilde{t}$ correspond to the transformed depth and time coordinates, which possess the following property: if $\tilde{z} = 0$, $\tilde{t} = s(t_0)$, and if $\tilde{t} = 0$, $\tilde{z} = v_0 s(t_v)$. $W$ is a varying coefficient defined as

$$W = a^2 + 2b (1 - a^2) ,$$

(8)
where

\[ b = \frac{\eta(z)}{\eta(\zeta)}, \quad a = \frac{s(\tau) v_0 v(x, z)}{\eta(\zeta)}, \quad \tau = \int_0^\zeta \frac{dz}{v(x, z)} = t + \int_0^z \frac{dz'}{v(x, z')} . \]

Since the \( W \) parameter varies slowly with \( x \) and \( \zeta \), Stolt suggested to replace it with its average value. Thus equation (7) is then approximated by an equation with constant coefficients, which has the dispersion relation

\[ \omega_v(k, \omega_0) = \left( 1 - \frac{W}{W} \right) \omega_0 + \frac{\text{sign}(\hat{\omega}_0)}{W} \sqrt{\omega_0^2 - Wv_0^2k^2} . \]  

(9)

As outlined above, Stolt’s approximate method for migration in heterogeneous media consists of the following steps:

1. stretching the time variable according to equation (5),
2. interpolating the stretched time to a regular grid,
3. double Fourier transform,
4. \( f-k \) time migration by the operator (2) with the dispersion relation (9),
5. inverse Fourier transform,
6. inverse stretching (that is, shrinking) of the vertical time variable on the migrated section.

The value of \( W \) must be chosen prior to migration. According to Stolt’s original definition (8), the depth variable \( z \) gradually changes in the migration process from zero to \( \zeta \), causing the coefficient \( b \) in (8) to change monotonically from 0 to 1. If the velocity \( v \) monotonically increases with depth, then \( \eta''(z) = \frac{dv}{dz} \geq 0 \), and the average value of \( b \) is

\[ \bar{b} = \frac{1}{\zeta \eta(\zeta)} \int_0^\zeta \eta(z) dz \leq \frac{1}{\zeta \eta(\zeta)} \int_0^\zeta \eta(\zeta) \frac{z}{\zeta} dz = \frac{1}{2} . \]  

(10)

As follows from equations (8) and (10), in the case of monotonically increasing velocity, the average value of \( W \) has to be less than 1 (\( W \) equals 1 in a constant-velocity case). Analogously, in the case of a monotonically decreasing velocity, \( W \) is always greater than 1. In practice, \( W \) is included in migration routines as a user-defined parameter, and its value is usually chosen to be somewhere in the range of 1/2 to 1. The next section describes a straightforward way to determine the most appropriate value of \( W \) for a given velocity distribution.

A useful tool for that purpose is Levin’s equation for the traveltime curve. Levin (1985) applied the stationary phase technique to the dispersion relation (9) to obtain
an explicit equation for the summation curve of the integral migration operator analogous to the Stolt stretch migration. The equation evaluates the summation path in the stretched coordinate system, as follows:

\[
s(t_0) = \left(1 - \frac{1}{W}\right)s(t_v) + \frac{1}{W}\sqrt{s^2(t_v) + W\left(\frac{x-x_0}{v_0^2}\right)^2},
\]

(11)

where \(x_0\) is the midpoint location on a zero-offset seismic section, and \(x\) is the space coordinate on the migrated section. Equation (11) shows that, with the stretch of the time coordinate, the summation curve has the shape of a hyperbola with the apex at \(\{x, s(t_v)\}\) and the center (the intersection of the asymptotes) at \(\{x, (1 - \frac{1}{W})s(t_v)\}\). In the case of homogeneous media, \(W = 1\), \(s(t) \equiv t\), and equation (11) reduces to the known expression for a hyperbolic diffraction traveltime curve. It is interesting to note that inverting equation (11) for \(s(t_v)\) determines the impulse response of the migration operator:

\[
\tilde{z} - \tilde{z}_0 = \left(\frac{1}{Q} - 1\right)R \pm \frac{1}{Q}\sqrt{R^2 - Q(x - x_0)^2},
\]

(12)

where \(R = v_0\hat{t}\), and \(Q = 2 - W\). Equation (12) can be interpreted as the wavefront from a point source in the \(\{x, \tilde{z}, \hat{t}\}\) domain of equation (7). Wavefronts from a point source in the stretched coordinates for \(W < 2\) have an elliptic shape, with the center of the ellipse at \(\{x, \tilde{z}_0 + \left(\frac{1}{Q} - 1\right)R\}\) and the semi-axes \(a_x = \frac{R}{\sqrt{Q}}\) and \(a_z = \frac{r}{Q}\). The ellipses stretch differently for \(W < 1\) and \(W > 1\), as shown in Figure 1. In the upper part that corresponds to the upgoing waves, the ellipses look nearly spherical, since the radius of the front curvature at the top apex equals the distance from the source.

Figure 1: Wavefronts from a point source in the stretched coordinate system. Left: velocity decreases with depth (\(W=1.5\)). Right: velocity increases with depth (\(W=0.5\)).
EVALUATING THE W PARAMETER

A remarkable connection between the Stolt stretch equation and different three-parameter traveltime approximations leads to a constructive estimate of the $W$ parameter. The first useful observation is a formal similarity between equation (11) and Malovichko’s approximation for the reflection traveltime curve in vertically inhomogeneous media (Malovichko 1978; Sword 1987; Castle 1988; de Bazelaire 1988) defined by

$$t_0 = \left(1 - \frac{1}{S(t_v)}\right) t_v + \frac{1}{S(t_v)} \sqrt{t_v^2 + S(t_v) \frac{(x-x_0)^2}{v_{rms}(t_v)}} .$$  \hspace{1cm} (13)

In equation (13), $v_{rms}$ is the effective (root mean square) velocity along the vertical ray

$$v_{rms}^2(t_v) = \frac{\eta(z)}{t_v} = \frac{1}{t_v} \int_{0}^{t_v} v^2(t) dt ,$$  \hspace{1cm} (14)

and $S$ is the parameter of heterogeneity, defined by the equation:

$$S(t_v) = \frac{1}{v_{rms}^4 t_v} \int_{0}^{t_v} v^4(t) dt .$$  \hspace{1cm} (15)

In terms of the $S$ parameter, the variance of the squared velocity distribution along the vertical ray is

$$\sigma^2 = \frac{1}{t_v} \int_{0}^{t_v} v^4(t) dt - v_{rms}^4 = v_{rms}^4(S - 1) .$$  \hspace{1cm} (16)

As follows from equality (16), $S \geq 1$ for any type of velocity distribution ($S$ equals 1 in a constant velocity case). For most of the distributions occurring in practice, $S$ ranges between 1 and 2.

Since reflection from a horizontal reflector in vertically-heterogeneous media is kinematically equivalent to diffraction from a point, we can regard equation (13), which is known as the most accurate three-parameter approximation of the NMO curve, as an approximation of the summation path for the post-stack Kirchhoff migration operator. In this case, it has the same meaning as equation (11). An important difference between the two equations is the fact that equation (13) is written in the initial coordinate system and includes coefficients varying with depth, while equation (11) applies the transformed coordinate system and constant coefficients. Using this fact, we compare the accuracy of the approximations and derive the following explicit expression, which relates Stolt’s $W$ factor to Malovichko’s parameter of heterogeneity:

$$W = 1 - \frac{v_0^2 s^2(t_v)}{v_{rms}^2(t_v) t_0^2} \left( \frac{v^2(t_v)}{v_{rms}^2(t_v)} - S(t_v) \right) .$$  \hspace{1cm} (17)

The details of the derivation are given in the appendix. Expression (17) is derived so as to provide the best possible value of $W$ for a given depth (or vertical time $t_v$).
To get a constant value for a range of depths, one should take an average of the right-hand side of (17) in that range. The error associated with Stolt stretch can be approximately estimated from (A-1) as the difference between the fourth-order terms:

\[ \delta = \frac{l^4}{8} \frac{W(t_v)}{t_v s^2(t_v) v_{rms}^2(t_v) v_0^2} , \]

where \( W(t_v) \) is the right-hand side of (17), and \( W \) is the constant value of \( W \) chosen for Stolt migration.

**Analytic Example**

A simple analytic example is the case of a constant velocity gradient. In this case the velocity distribution can be described by the linear function \( v(z) = v(0)(1 + \alpha z) \). The Stolt stretch transform for this case can be derived directly from equation (5) and takes the form

\[ s(t) = \left( e^{2\alpha v(0)t} - 1 - 2\alpha v(0) t \right)^{1/2} . \]

Let \( \kappa \) be the logarithm of the velocity change \( v(z)/v(0) \). Then an explicit expression for \( W \) factor is found according (17) as

\[ W = \frac{2\kappa}{e^{2\kappa} - 1} = \frac{v^2(0)}{v_{rms}^2(z)} . \]

In the case of a small \( \kappa \), which corresponds to a small depth or a small velocity gradient, \( W \approx 1 - \kappa \). In the case of a large \( \kappa \), \( W \) monotonically approaches zero. Equation (20) can be a useful rule of thumb for a rough estimation of \( W \).

**Stolt stretch for anisotropic media**

As follows from the analysis of the reflection moveout in a vertically heterogeneous transversely isotropic medium (Fomel and Grechka, 1996), expression (17) for the Stolt stretch parameter will remain valid in this case if the values of \( v_{rms} \) and \( S \) are computed according to equations

\[ v_{rms}^2(t_v) = \frac{1}{t_v} \int_0^{t_v} v^2(t) (1 + 2\delta(t)) \, dt , \]

\[ S(t_v) = \frac{1}{v_{rms}^4 t_v} \int_0^{t_v} v^4(t) (1 + 2\delta(t))^4 (1 + 8\eta(t)) \, dt , \]

where \( \delta \) and \( \eta \) are the conventional anisotropic parameters (Thomsen, 1986; Alkhalifah and Tsvankin, 1995), which may vary with depth.
As we demonstrate in the next section, the method of cascaded migrations (Larner and Beasley, 1987) can improve the performance of Stolt migration in the case of variable velocity (Beasley et al., 1988). However, this method affects only the isotropic part of the model and cannot change the contribution of the anisotropic parameters. Therefore, in the anisotropic case, it is important to incorporate anisotropic parameters into the Stolt stretch correction.

APPLICATION

Following the study by Larner et al. (1989), we selected a dataset that includes steep dips in order to test the accuracy of our algorithms. The dataset is courtesy of Elf Aquitaine. It was recorded in the North Sea over a salt-dome structure. Figure 2 shows the data after NMO-stack and after post-stack Stolt migration, using a constant velocity of 2000 m/s. The Stolt method creates visible undermigrated events on both sides of the salt body. Using a higher velocity to focus them better would have created overmigration artifacts at shallow reflectors. Stolt-stretch migrated section using \( W = 0.5 \) is shown in Figure 2c. It should be compared with an improved result shown in Figure 3a.

Using the Stolt-stretch method with the optimal choice for \( W \) estimated from equation (17) yields a better focusing of events at all depths (Figure 3a), compared to other values of \( W \) (Figures 2b and 2c, respectively for \( W \) equals 1.0 and 0.5). The \( v(z) \) model used for migration is shown in Figure 4a and was obtained by averaging laterally the reference velocity model.

The reference method of migration for our study is the phase-shift method (Gazdag, 1978). It is known to be perfectly accurate for all dips up to 90° in a \( v(z) \) velocity field. A comparison between the phase-shift migration result (Figure 3b) and the section migrated with the Stolt-stretch approach shows almost no difference for flat events. However, a more detailed analysis reveals significant errors for steep events inside and around the salt body. The approximation made by stretching the time axis breaks for recovering steep events.

A way to overcome the difficulties encountered by Stolt’s migration is to divide the whole process into a cascade, as suggested by Beasley et al. (1988). The theory of cascaded migration proves that \( f-k \) migration algorithms with a \( v(t) \) velocity model like Stolt-stretch can be performed sequentially as a cascade of \( n \) migrations with smaller interval velocities \( v_i(t) \), \( i = 1, \ldots, n \), such then

\[
v^2(t) = \sum_{i=1}^{n} v_i^2(t).
\]

At a given vertical traveltime \( t \), all the successive velocity models have to be constant, except the last one (Larner and Beasley, 1987). Typically, the first stage is done with a constant velocity model and can be computed using Stolt’s method, which is then
Figure 2: (a) Section of the North Sea data, after NMO-stack. (b) Section migrated using Stolt’s method with \( v_0 = 2000 \) m/s. (c) Section migrated using Stolt–stretch with an arbitrary value \( W = 0.5 \) for the parameter of heterogeneity.
Figure 3: (a) Section migrated with the Stolt-stretch method using the optimal value ($\approx 0.67$) for the parameter $W$. (b) Section migrated with the phase-shift method. (c) Section migrated using the cascaded Stolt-stretch approach (5 velocities).
accurate for all dips. Figure 4 illustrates such a cascade of velocity models in our particular case, with 3 and 5 stages.

Figure 4: (a) Interval velocity model $v(t)$ estimated from the 2-D reference model. (b) Decomposition in a cascade of 3 models, such as $v^2 = v_1^2 + v_2^2 + v_3^2$. (c) Decomposition in a cascade of 5 models, such as $v^2 = v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2$.

As a consequence of this decomposition, each intermediate velocity model shows not only a smaller velocity but also less vertical heterogeneity. In other words, the Stolt-stretch parameter $W$ estimated for each stage tends to be closer to 1.0, thus reducing the migration errors due to the approximation. Figure 3c shows the migration result using a 5-stage cascaded scheme. All the successive values of $W$ were greater than 0.8. There are almost no differences with the phase-shift result (Figure 3b).

An accuracy of the cascaded stolt-stretch migration is additionally verified by comparing its impulse response with that of the phase-shift migration (Figure 5). The impulses are generated using the same velocity model as shown in Figure 4a. Figure 6 provides a more detailed comparison. We can see a kinematic difference in the impulse response of Stolt-stretch compared to phase-shift. While Gazdag's phase-shift honor ray bending in any $v(z)$ model, Stolt-stretch is only designed to make the fitting curve look like an hyperbola close to the apex (Levin 1983), and therefore induces residual migration errors. As seen in Figure 3a, Stolt-stretch result displays residual hyperbolic migration artifacts that are due to this fundamental kinematic difference. Cascading Stolt-stretch makes the impulse response of the migration converge towards the one of phase-shift.

Figure 7 shows a close-up of the salt body region for all migration algorithms. The methods have a different accuracy with respect to steep dips. We notice a gradual
improvement of the result from Stolt-stretch to phase-shift as we increase the number of velocities in the cascaded Stolt-stretch scheme. In theory, the migration errors in the cascaded approach can be made as small as desired by increasing the number of stages. At the limit, it corresponds to the velocity continuation concept (Fomel, 1994, 1997).

In our case, six stages were enough to obtain a result comparable to phase-shift. In their comparative study on time migration algorithms, Larner et al. (1989) have shown that four-stage cascaded f-k migration is accurate for dips up to 85°, which is almost comparable to phase-shift, accurate for all dips. It is worth noting the computational cost difference between the two: on our example, phase-shift migration is about 80 times more expensive than Stolt-stretch.

![3-D impulses responses of the cascaded Stolt-stretch (a) and phase-shift (b) operators.](stoltst/imps/ imp-mig3)

**CONCLUSIONS**

An explicit expression for the Stolt-stretch parameter, derived in this paper, allows us to achieve optimal accuracy when applying Stolt migration in vertically heterogeneous media.

Combining an optimal analytical choice for the Stolt-stretch parameter with the cascaded f-k migration approach, we manage to obtain time migration results comparable to Gazdag’s phase-shift migration. The Stolt method is considerably more computer-efficient and remains accurate for steeply dipping events.

**ACKNOWLEDGMENTS**

The authors would like to thank the sponsors of the Stanford Exploration Project for financial support and Elf Aquitaine for providing the data used in this work.
Figure 6: Impulses responses of the different operators. (a) Stolt-stretch. (b) Phase-shift. (c) and (d) Cascaded Stolt-stretch, with 3 and 5 velocities, respectively.
Figure 7: Zoom in the salt body area where steep dips are located. (a) Migration with the Stolt-stretch method using optimal $W$. (b) Migration with the phase-shift method. (c) Migration with the Stolt-stretch method using $W = 0.5$. (d) Migration with the cascaded Stolt-stretch approach, using 5 velocities. 

[stlst/elfst/ dip-zoom]
REFERENCES


APPENDIX A

In this Appendix, we derive an explicit expression for the Stolt-stretch parameter $W$ by comparing the accuracy of equations (11) and (13), which approximate the traveltime curve in the neighborhood of the vertical ray. It is appropriate to consider a series expansion of the diffraction traveltime in the vicinity of the vertical ray:

$$t_0(l) = t_0|_{l=0} + \frac{1}{2} \frac{d^2 t_0}{dl^2} \bigg|_{l=0} l^2 + \frac{1}{4!} \frac{d^4 t_0}{dl^4} \bigg|_{l=0} l^4 + \cdots , \quad (A-1)$$

where $l = x - x_0$. Expansion (A-1) contains only even powers of $l$ because of the obvious symmetry of $t_0$ as a function of $l$.

Matching the series expansions term by term is a constructive method for relating different equations to each other. The special choice of parameters $t_v$, $v_{rms}$, and $S$ allows Malovichko’s equation (13) to provide correct values for the first three terms of expansion (A-1):

$$t_0|_{l=0} = t_v ; \quad (A-2)$$
$$\frac{d^2 t_0}{dl^2} \bigg|_{l=0} = \frac{1}{t_v v_{rms}^2 (t_v)} ; \quad (A-3)$$
$$\frac{d^4 t_0}{dl^4} \bigg|_{l=0} = -\frac{3 S (t_v)}{t_v^3 v_{rms}^4 (t_v)} . \quad (A-4)$$

Considering Levin’s equation (11) as an implicit definition of the function $t_0 (t_v)$, we can iteratively differentiate it, following the rules of calculus:

$$\left. \frac{ds}{dl} \right|_{l=0} = s' (t_0) \left. \frac{dt_0}{dl} \right|_{l=0} = 0 ;$$

$$\left. \frac{d^2 s}{dl^2} \right|_{l=0} = \left( s' (t_0) \left. \frac{d^2 t_0}{dl^2} \right|_{l=0} + s'' (t_0) \left( \left. \frac{dt_0}{dl} \right|_{l=0} \right)^2 \right) \left|_{l=0} = s' (t_v) \left. \frac{d^2 t_0}{dl^2} \right|_{l=0} = \frac{1}{v_{rms}^2 (t_v)} ; \quad (A-5)$$

$$\left. \frac{d^3 s}{dl^3} \right|_{l=0} = \left( 3 s'' (t_0) \left. \frac{dt_0}{dl} \right|_{l=0} \left. \frac{d^2 t_0}{dl^2} \right|_{l=0} + s' (t_0) \left. \frac{d^3 t_0}{dl^3} \right|_{l=0} + s'' (t_0) \left( \left. \frac{dt_0}{dl} \right|_{l=0} \right)^3 \right) \left|_{l=0} = 0 \right.$$
\[
\left. \frac{d^4 s}{dt^4} \right|_{t=0} = \left( 6 s'''(t_0) \left( \frac{dt_0}{dl} \right)^2 + 3 s''(t_0) \left( \frac{d^2 t_0}{dl^2} \right)^2 + 4 s''(t_0) \frac{dt_0}{dl} \frac{d^3 t_0}{dl^3} + \right.
\]
\[
+ s'(t_0) \left( \frac{d^4 t_0}{dl^4} + s^{IV}(t_0) \left( \frac{dt_0}{dl} \right)^4 \right) \left|_{t=0} = \right.
\]
\[
= \left( s''(t_v) \left( \frac{d^2 t_0}{dl^2} \right)^2 + s'(t_v) \frac{d^3 t_0}{dl^3} \right) \left|_{t=0} = \right. - \frac{3 W}{v_0^4 s^3(t_0)}. \quad (A-6)
\]

Substituting the definition of Stolt stretch transform (5) into (A-5) produces an equality similar to (A-3), which means that approximation (11) is theoretically accurate in depth-varying velocity media up to the second term in (A-1). It is this remarkable property that proves the validity of the Stolt stretch method (Levin, 1983; Claerbout, 1985). Moreover, equation (11) is accurate up to the third term if the value of the fourth-order traveltime derivative in (A-6) coincides with (A-4). Substituting equation (A-4) into (A-6) results in the expression
\[
\frac{1 - W}{v_0^2 s^2(t_v)} = \frac{v^2(t_v) - S(t_v)}{v_{rms}^2(t_v)} t_v^2. \quad (A-7)
\]

It is now easy to derive from equation (A-7) the desired explicit expression for the Stolt stretch parameter \( W \): equation (17) in the main text.
Antialiasing of Kirchhoff operators by reciprocal parameterization

Sergey Fomel

ABSTRACT

I propose a method for antialiasing Kirchhoff operators, which switches between interpolation in time and interpolation in space depending on the operator dips. The method is a generalization of Hale’s technique for dip moveout antialiasing. It is applicable to a wide variety of integral operators and compares favorably with the popular temporal filtering technique. Simple synthetic examples demonstrate the performance and applicability of the proposed method.

INTRODUCTION

Integral (Kirchhoff-type) operators are widely used in seismic imaging and data processing for such tasks as migration, dip moveout (Hale 1991), azimuth moveout (Biondi et al. 1998), and shot continuation (Bagaini and Spagnolini 1996). In theory, the operators correspond to continuous integrals. In practice, the integration is replaced by summation and becomes prone to sampling errors. A common problem with practical implementation of integral operators is the operator aliasing, caused by spatial undersampling of the summation path (Lumley et al. 1994). When the integration path is parameterized in the spatial coordinate, as it is commonly done in practice, the steeper part of the summation path becomes undersampled.

The operator aliasing problem, as opposed to the data aliasing and image aliasing problems, is discussed in detail by Lumley et al. (1994) and Biondi (2001). It arises when the slope of the operator traveltime exceeds the limit, defined by the time and space sampling of the data - the Nyquist frequencies (Claerbout 1992a). Even if the input data are not aliased, operator aliasing can cause severe distortions in the output. Several successful techniques have been proposed in the literature to overcome the operator aliasing problem. Different versions of the temporal filtering method were suggested by Gray (1992) and Lumley et al. (1994) and further enhanced by Abma et al. (1999) and Biondi (2001). This method reduces the aliasing error by limiting the rate of change in the integrand (the input data) with temporal filtering. Unfortunately, this approach is suboptimal in the case of rapid changes in the summation path gradient. A different approach to antialiasing was suggested by Hale (1991) for the integral dip moveout. Hale’s approach provides accurate results by
parameterizing the operator in the time coordinate rather than the space coordinate. Unfortunately, this approach requires an additional expense of interpolation in both space and time coordinates for computing the flat part of the operator.

In this paper, I propose a new antialiasing method derived from the time-slice technique, developed by [Hale (1991)]. The method switches between interpolation in time and interpolation in space depending on the local operator dips. It is particularly attractive for computing 3-D operators with rapidly varying dips and limited aperture ([Fomel and Biondi (1995)]. Synthetic examples show the superior performance of the new method in comparison the temporal filtering approach.

OVERVIEW OF EXISTING METHODS

I start with reviewing the existing approaches to operator antialiasing and discussing their main principles and limitations. The two reviewed approaches are temporal filtering, as suggested by [Gray (1992)] and [Lumley et al. (1994)], and Hale’s spatial filtering technique, developed originally for an integral implementation of the dip moveout operator ([Hale (1991)]).

Temporal filtering

The temporal filtering idea follows from the well-known Nyquist sampling criterion. With application to integral operators, the Nyquist criterion takes the form

\[ \Delta x \leq \frac{\Delta t}{|\partial t/\partial x|} , \]  

where \( t(x) \) is the traveltime of the operator impulse response, \( \Delta x \) is the space sampling interval and \( \Delta t \) is the time sampling interval. In the steep parts of the traveltime curve, the sampling criterion (1) is not satisfied, which causes aliasing artifacts in the output data. To overcome this problem, the method of local triangle filtering ([Claerbout (1992a)] [Lumley et al. (1994)]) suggests convolving the traces of the generated impulse response with a triangle-shaped filter of the length

\[ \delta t = \Delta x |\partial t/\partial x| . \]  

Cascading operators of causal and anticausal numerical integration is an efficient way to construct the desired filter shape.

Triangle filters approximate the ideal (sinc) low-pass time filters. The idea of using low-pass filtering for antialiasing ([Gray (1992)]) is illustrated in Figure 1. When a steeply dipping event is included in the operator, its counterpart in the frequency domain wraps around to produce the aliasing artifacts. Those are removed by a dip-dependent low-pass filtering.
Figure 1: Schematic illustration of low-pass antialiasing (triangle filters). The aliased events are removed by low-pass filtration on the temporal frequency axis. The width of the low-pass filter depends on dips of the aliased events.

The method of low-pass filtering is less evident in the case of a three-dimensional integral operator. We can take the length of a triangle filter proportional to the absolute value of the time gradient (Lumley et al., 1994), the maximum of the gradient components in the two directions of the operator space (Abma et al., 1999), or the sum of these components. The latter follows from considering the 3-D operator as a double integration in space. Decoupling the 3-D integral into a cascade of two 2-D operators suggests convolving two triangle filters designed with respect to two coordinates of the operator. In this case, the length of the resultant filter is approximately equal to

$$\delta t = \Delta x |\partial t/\partial x| + \Delta y |\partial t/\partial y|,$$

and its shape is smoother than that of a triangle filter (Figure 2).

Figure 2: Building the smoothed filter for 3-D antialiasing by successive integration of a five-point wavelet. C denotes the operator of causal integration, C' denotes its adjoint (the anticausal integration). The result is equivalent to the convolution of two equal triangle filters.

The temporal filtering method was proven to be an efficient tool in the design of
stacking operators of different types. However, when the operator introduces rapid changes in the length and direction of the traveltime gradient, it leads to an inexact estimation of the filter cutoff (triangle length for the method of triangle filtering) at the curved parts of the operator. Consequently, the high-frequency part of the output can be distorted, causing a loss in the image resolution.

Hale’s method

Considering the case of integral dip moveout, [Hale (1991)] points out that the steep parts of the operator, while aliased in the space (midpoint) coordinate, are not aliased with respect to the time coordinate. He suggests replacing the conventional $t(x)$ parameterization of the DMO impulse response by $x(t)$ parameterization. Conventionally, the integral operators are implemented by shifting the input traces in space and transforming them in time. According to Hale’s method, the traces are shifted in time and transformed along the $x(t)$ trajectories in space. Interpolation in time, required in the conventional approach, is replaced by interpolation in space. The idea of Hale’s method is related to the idea of the “pixel-precise velocity transform” (Claerbout [1992b]).

The steep parts of the operator satisfy the criterion

$$\Delta t \leq \frac{\Delta x}{|\partial x/\partial t|},$$

(4)

which is the obvious reverse of inequality (1). Therefore, they are not aliased if defined on the time grid. In these parts, one can implement the operator by constant time shifts equal to the time sampling interval $\Delta t$. In the parts where the criterion (4) is not valid (the flat part of the DMO operator), Hale suggests reducing the length of the time shifts according to equality (2), where $\delta t$ becomes less than $\Delta t$. He formulates the following principle of operator antialiasing:

To eliminate spatial aliasing, simply never allow successive time shifts applied to the input trace to differ by more than one time sampling interval. Further restrict the difference between time shifts so that the spacing between the corresponding output trajectories never exceeds the CMP sampling interval.

The idea of Hale’s method is illustrated in Figure [3]. Increasing the density of spatial sampling by small successive time shifts implies increasing the Nyquist boundaries of the spatial wavenumber. Further interpolation is a low-pass spatial filtering that removes the parts of the spectrum beyond the Nyquist frequency of the output. If the dip of the operator does not vary between neighboring traces (the operator is a straight line as in the slant stack case), Hale’s approach will produce essentially the same result as that of temporal filtering. Triangle filters in this case approximately correspond to linear interpolation in space between adjacent traces [Nichols [1993]].
The difference between the two approaches occurs if the local dip varies in space as in the case of a curved operator, such as DMO. In this case, Hale’s approach provides a more accurate space interpolation of the operator and preserves the high-frequency part of its spectrum from distortion.

Hale’s method has proven to preserve the amplitude of flat reflectors from aliasing distortions, which is the simplest antialiasing test on a DMO operator. The most valuable advantage of this method in the fact that the implied low-pass spatial filtering (interpolation) does not depend on the operator dip and is controlled by the Nyquist boundary of the spectrum only (compare Figures 1 and 3). This is especially important, when the local dip of the operator changes rapidly and therefore cannot be estimated precisely by finite-difference approximation at spatially separated traces. Such a situation is common in dip moveout and azimuth moveout integral operators, as well as in prestack Kirchhoff migration.

A weakness of the method is the necessity to switch from interpolation in space to two-dimensional interpolation in both the time and the space variables, when trying to construct the flat part of the operator. In the next section, I show how to avoid the expense of the additional time interpolation required by Hale’s method of antialiasing.

**PROPOSED TECHNIQUE**

We can use the reciprocity of the time parameterization and the space parameterization of integral operators, discovered by Hale, to arrive at the following antialiasing technique.

For simplicity, let us consider the two-dimensional case first. The linearity of a two-dimensional integral operator allows us to decompose this operator into two parts. The first part corresponds to the steep part of the travel-time function, which satisfies the time-sampling criterion (1). The second term corresponds to the flat part of the traveltime, which satisfies the space-sampling criterion (1). The first part is not aliased with respect to the time sampling interval, while the second one is not
aliased with respect to the space sampling. We can apply interpolation in time to construct the flat part. Reciprocally, interpolation in space is applied to construct the steep part of the operator in the fashion of Hale’s time-shifting method. The amplitude difference between the two integrals is simply the Jacobian term

\[
\frac{\text{amp}_t}{\text{amp}_x} = \left| \frac{\partial x}{\partial t} \right| \frac{\Delta t}{\Delta x} = \frac{\Delta t}{\delta t} \leq 1.
\] (5)

According to the proposed modification, Hale’s antialiasing principle is reformulated, as follows:

*In the steep part of an integral operator, never allow successive time shifts applied to the input trace to differ by more than one time sampling interval.*

*In the flat part of the operator, never allow successive space shifts to differ by more than one space sampling interval.*

Figure 4, borrowed from Claerbout (1995), illustrates the basic idea of the proposed technique. It clearly shows the difference between the flat and steep parts of migration hyperbolas. To observe the reciprocity, rotate the figure by 90 degrees.

![Figure 4](image)

Figure 4: Figure borrowed from Claerbout (1995) to illustrate the reciprocity antialiasing. The flat parts of the hyperbolas require interpolation in time. The steep parts of the hyperbolas require interpolation in space.

To compare the proposed antialiasing method with the temporal filtering method, I test the antialiased migration program on simple 2-D synthetic tests. Figure 5 shows a simple model and the modeling results from modeling without antialiasing, with temporal filtering, and with the proposed reciprocity method. The modeling results were migrated with the corresponding migration operators to obtain the image of the model in Figure 6. Both the temporal filtering and the proposed method succeed in removing the major aliasing artifacts. However, the reciprocity method demonstrates a higher resolution and a better preservation of the frequency content.

These properties are examined more closely in the next synthetic example. Figure 7 shows a more sophisticated synthetic model that contains a fault, an unconformity
and layered structures (Claerbout, 1995). For better displaying, I extract the central part of the model and compare it with the migration results of different methods in Figure 8. Comparing the plots shows that the reciprocity method successfully removes the aliasing artifacts (round-off errors) of the aliased (nearest neighbor interpolation) migration. At the same time, it is less harmful to the high-frequency components of the data than triangle filtering. This conclusion finds an additional support in Figure 9 that displays the average spectrum of the image traces for different methods. Both of the antialiasing methods remove the high-frequency artifacts of the nearest neighbor modeling and migration. The reciprocity method performs it in a gentler way, preserving the high-frequency components of the model.

The algorithm sequence of the antialiased migration is illustrated in Figures 10 and 11. The two plots in Figure 10 show the steep-dip and flat-dip modeling respectively. The superposition of these two terms is the resultant antialiased data shown in the left plot of Figure 12. The right plot of Figure 12 shows the migrated image obtained by adding the flat-dip (left of Figure 11) and steep-dip (right of Figure 11) migrations.
Figure 6: Top left plot is the synthetic model. The other plots are migrations of the corresponding data shown in the previous figure. Top right is a migration without antialiasing. Bottom left is a migration with reciprocity antialiasing (the proposed method). Bottom right is a migration with triangle filter antialiasing.

antial/mod/amomig

Figure 7: Synthetic model used to test the antialiased migration program.

antial/sig/amosmo
Figure 8: Top left plot is a zoomed portion of the synthetic model. The other plots are migrated images. Top right is a migration without antialiasing. Bottom left is a migration with reciprocity antialiasing (the proposed method). Bottom right is a migration with triangle filter antialiasing.
Figure 9: Top is the spectrum of the model. The other plots are the spectra of the migrated images. The second plot corresponds to the modeling/migration without account for antialiasing. The third plot is modeling/migration with the reciprocity antialiasing. The bottom plot is modeling/migration with triangle antialiasing.

Figure 10: Antialiased modeling. Left corresponds to the flat-dip term. Right corresponds to the steep-dip term.
Figure 11: Antialiased migration. Left corresponds to the flat-dip term. Right corresponds to the steep-dip term.

Figure 12: Antialiased modeling and migration. Left is the superposition of the flat-dip and steep-dip modeling. Right is superposition of the flat-dip and steep-dip migration.
3-D antialiasing

The proposed method of antialiasing is easily generalized to the case of a three-dimensional integral operator. In this case, we need to consider three different parameterizations: \( t(x, y) \), \( x(t, y) \), and \( y(t, x) \) and switch from one of them to another according to the rule:

- if \( \Delta t \geq \Delta x |\partial t/\partial x| \) and \( \Delta t \geq \Delta y |\partial t/\partial y| \), use \( t(x, y) \),
- if \( \Delta x \geq \Delta t |\partial x/\partial t| \) and \( \Delta x \geq \Delta y |\partial x/\partial y| \), use \( x(t, y) \),
- if \( \Delta y \geq \Delta t |\partial y/\partial t| \) and \( \Delta y \geq \Delta x |\partial y/\partial x| \), use \( y(t, x) \).

Following [Biondi](2001), I illustrate 3-D antialiasing by applying prestack time migration on a single input trace. The results are shown in Figures 13, 14 and 15. The result without any antialiasing protection (Figure 13) contains clearly visible aliasing artifacts caused by the steeply dipping parts of the operator. Antialiasing by temporal filtering (Figure 14) removes the artifacts but also attenuates the steeply dipping events. Antialiasing by the proposed reciprocal parameterization (Figure 15) removes the aliasing artifacts while preserving the steeply dipping events and the image resolution.

CONCLUSIONS

I have introduced a new method of antialiasing integral operators, modified from Hale’s approach to antialiased dip moveout. The method compares favorably with the popular temporal filtering technique. The main advantages are:

1. Accurate handling of variable operator dips.
2. Consequent preservation of the high-frequency part of the data spectrum, leading to a higher resolution.
3. Easy control of operator amplitudes.
4. Easy generalization to 3-D.

The method possesses a sufficient numerical efficiency in practical implementations. Its most appropriate usage is for antialiasing operators with analytically computed summation paths, such as prestack time migration, dip moveout, azimuth moveout, and shot continuation.

ACKNOWLEDGMENTS

I thank Biondo Biondi for many helpful discussions. The financial support for this work was provided by the sponsors of the Stanford Exploration Project.
Figure 13: Prestack 3-D time migration of a single input trace. Top: time slice at 1 s. Bottom: vertical slice. No antialiasing protection has been applied. As a result, aliasing artifacts are clearly visible in the image.
Figure 14: Prestack 3-D time migration of a single input trace. Top: time slice at 1 s. Bottom: vertical slice. Antialiasing by temporal filtering has been applied. Aliasing artifacts are removed, steeply dipping events are attenuated.
Figure 15: Prestack 3-D time migration of a single input trace. Top: time slice at 1 s. Bottom: vertical slice. The proposed reciprocal antialiasing has been applied. Aliasing artifacts are removed, steeply dipping events are preserved.
REFERENCES

Angle-gather time migration

Sergey Fomel and Marie Prucha

ABSTRACT
Angle-gather migration creates seismic images for different reflection angles at the reflector. We formulate an angle-gather time migration algorithm and study its properties. The algorithm serves as an educational introduction to the angle gather concept. It also looks attractive as a practical alternative to conventional common-offset time migration both for velocity analysis and for AVO/AVA analysis.

INTRODUCTION
Angle-gather migration creates seismic images collected by the reflection angle at the point of reflection. Major advantages of this approach are apparent in the case of prestack depth migration. As shown by Prucha et al. (1999), the ray pattern of angle-gather migration is significantly different from that of the conventional common-offset migration. The difference can be exploited for overcoming illumination difficulties of the conventional depth migration in complex geological areas.

In this paper, we explore the angle-gather concept in the case of prestack time migration. The first goal of this study is educational. Since we can develop the complete mathematical theory of angle-gather time migration analytically, it is much easier to understand the most basic properties of the method in the time migration domain. The second goal is practical. Angle gathers present an attractive tool for post-migration AVO/AVA studies and velocity analysis, and even the most basic time migration approach can find a valuable place in the complete toolbox of seismic imaging.

We start with analyzing the traveltime relations for the basic Kirchhoff implementation of angle-gather time migration. The analysis follows Fowler’s general approach to prestack time migration methods (Fowler, 1997). Next, we derive formulas for the amplitude weighting and discuss some frequency-domain approaches to angle gathers. Finally, we present simple synthetic tests of the method and discuss further research directions.
TRAVELTIME CONSIDERATIONS

Let us consider a simple reflection experiment in an effectively constant-velocity medium, as depicted in Figure 1. The pair of incident and reflected rays and the line between the source $s$ and the receiver $r$ form a triangle in space. From the trigonometry of that triangle we can derive simple relationships among all the variables of the experiment (Fomel, 1995, 1996a, 1997).

Introducing the dip angle $\alpha$ and the reflection angle $\gamma$, the total reflection travel-time $t$ can be expressed from the law of sines as

$$t = \frac{2h \cos(\alpha + \gamma) + \cos(\alpha - \gamma)}{v \sin 2\gamma} = \frac{2h \cos \alpha}{v \sin \gamma},$$  

(1)

where $v$ is the medium velocity, and $h$ is the half-offset between the source and the receiver.

Additionally, by following simple trigonometry, we can connect the half-offset $h$ with the depth of the reflection point $z$, as follows:

$$h = \frac{z}{2} \frac{\sin 2\gamma}{2 \cos(\alpha + \gamma) \cos(\alpha - \gamma)} = \frac{z \sin \gamma \cos \gamma}{\cos^2 \alpha - \sin^2 \gamma}.$$  

(2)

Finally, the horizontal distance between the midpoint $x$ and the reflection point $\xi$ is

$$x - \xi = h \frac{\cos(\alpha - \gamma) \sin(\alpha + \gamma) + \cos(\alpha + \gamma) \sin(\alpha - \gamma)}{\sin 2\gamma} = h \frac{\sin \alpha \cos \alpha}{\sin \gamma \cos \gamma}.$$  

(3)

Equations (1–3) completely define the kinematics of angle-gather migration. Re-
grouping the terms, we can rewrite the three equations in a more symmetric form:

\[
\begin{align*}
t &= \frac{2 z \cos \alpha \cos \gamma}{v \cos^2 \alpha - \sin^2 \gamma} \\
h &= \frac{z \sin \gamma \cos \gamma}{\cos^2 \alpha - \sin^2 \gamma} \\
x - \xi &= \frac{z \sin \alpha \cos \alpha}{\cos^2 \alpha - \sin^2 \gamma}
\end{align*}
\] (4-6)

For completeness, here is the inverse transformation from \( t, h, \) and \( x - \xi \) to \( z, \gamma, \) and \( \alpha \):

\[
\begin{align*}
z^2 &= \frac{[(vt/2)^2 - (x - \xi)^2][vt/2]^2 - h^2}{(vt/2)^2} \\
\sin^2 \gamma &= \frac{h^2[(vt/2)^2 - (x - \xi)^2]}{(vt/2)^4 - h^2(x - \xi)^2} \\
\cos^2 \alpha &= \frac{(vt/2)^2[(vt/2)^2 - (x - \xi)^2]}{(vt/2)^4 - h^2(x - \xi)^2}
\end{align*}
\] (7-9)

The inverse transformation (107-9) can be found by formally solving system (4-6).

The lines of constant reflection angle \( \gamma \) and variable dip angle \( \alpha \) for a given position of a reflection (diffraction) point \( \{z, \xi\} \) have the meaning of summation curves for angle-gather Kirchhoff migration. The whole range of such curves for all possible values of \( \gamma \) covers the diffraction traveltime surface - “Cheops’ pyramid” (Claerbout, 1985) in the \( \{t, x, h\} \) space of seismic reflection data. As pointed out by Fowler (1997), this condition is sufficient for proving the kinematic validity of the angle-gather approach. For comparison, Figure 2 shows the diffraction traveltime pyramid from a diffractor at 0.5 km depth. The pyramid is composed of common-offset summation curves of the conventional time migration. Figure 3 shows the same pyramid composed of constant-\( \gamma \) curves of the angle-gather migration.

Figure 2: Traveltime pyramid, composed of common-offset summation curves.
Figure 3: Traveltime pyramid, composed of common-reflection-angle summation curves.

- For each reflection angle $\gamma$ and each dip angle $\alpha$,
  - For each output location $\{z, \xi\}$,
    1. Find the traveltime $t$, half-offset $h$, and midpoint $x$ from formulas (4), (5), and (6) respectively.
    2. Stack the input data values into the output.

As follows from equations (4-6), the range of possible $\alpha$’s should satisfy the condition

$$\cos^2 \alpha > \sin^2 \gamma \quad \text{or} \quad |\alpha| + |\gamma| < \frac{\pi}{2}. \quad (10)$$

The described algorithm is not the most optimal in terms of the input/output organization, but it can serve as a basic implementation of the angle-gather idea. The stacking step requires an appropriate weighting. We discuss the weighting issues in the next section.

**AMPLITUDE CONSIDERATIONS**

One simple approach to amplitude weighting for angle-gather migration is based again on Cheops’ pyramid considerations. Stacking along the pyramid in the data space is a double integration in midpoint and offset coordinates. Angle-gather migration implies the change of coordinates from $\{x, h\}$ to $\{\alpha, \gamma\}$. The change of coordinates leads to weighting the integrand by the following Jacobian transformation:

$$dxdh = \left| \det \left( \begin{array}{cc} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial h}{\partial \alpha} & \frac{\partial h}{\partial \gamma} \end{array} \right) \right| d\alpha d\gamma \quad (11)$$

Substituting formulas (5) and (6) into equation (8) gives us the following analytical expression for the Jacobian weighting:

$$W_J = \left| \det \left( \begin{array}{cc} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial h}{\partial \alpha} & \frac{\partial h}{\partial \gamma} \end{array} \right) \right| = \frac{z^2}{(\cos \alpha^2 - \sin \gamma^2)^2} \quad (12)$$

Weighting (12) should be applied in addition to the weighting used in common-offset migration. By analyzing formula (12), we can see that the weight increases with the reflector depth and peaks where the angles $\alpha$ and $\gamma$ approach condition (10).

The Jacobian weighting approach, however, does not provide physically meaningful amplitudes, when migrated angle gathers are considered individually. In order
to obtain a physically meaningful amplitude, we can turn to the asymptotic theory of true-amplitude migration (Goldin, 1992; Schleicher et al., 1993; Tygel et al., 1994). The true-amplitude weighting provides an asymptotic high-frequency amplitude proportional to the reflection coefficient, with the wave propagation (geometric spreading) effects removed. The generic true-amplitude weighting formula (Fomel, 1996b) transforms in the case of 2-D angle-gather time migration to the form:

$$W_{TA} = \frac{1}{\sqrt{2\pi v \cos \gamma}} \left( \frac{\partial^2 L_s}{\partial \xi \partial \gamma} + \frac{\partial^2 L_r}{\partial \xi \partial \gamma} \right),$$

(13)

where $L_s$ and $L_r$ are the ray lengths from the reflector point to the source and the receiver respectively. After some heavy algebra, the true-amplitude expression takes the form

$$W_{TA} = \frac{2z \sin \alpha}{\sqrt{2\pi v}} \frac{\cos^2 \alpha + \sin^2 \gamma}{\left(\cos^2 \alpha - \sin^2 \gamma\right)^{5/2}}.$$

(14)

Under the constant-velocity assumption and in high-frequency asymptotic, this weighting produces an output, proportional to the reflection coefficient, when applied for creating an angle gather with the reflection angle $\gamma$. Despite the strong assumptions behind this approach, it might be useful in practice for post-migration amplitude-versus-angle studies. Unlike the conventional common-offset migration, the angle-gather approach produces the output directly in reflection angle coordinates. One can use the generic true-amplitude theory (Fomel, 1996b) for extending formula (14) to the 3-D and 2.5-D cases.

**EXAMPLES**

We created some simple synthetic models with constant velocity backgrounds to test our angle-gather migration method. One model is a simple dome (Figure 4). The other has a series of flat reflectors of various dips (Figure 5). Both of these figures also show the corresponding data that will be generated by Kirchhoff methods for zero and far offsets.

![Figure 4: Left: Model. Center: Data at zero offset. Right: Data at far offset.](angle/agmig/data-dome)
Dome model

This model contains a wide range of geologic dips across the dome as well as having a flat reflector at the base of the dome. Figure 6 shows the resulting common offset sections from traditional Kirchhoff migration. As is expected for such a simple model, the near and far offset sections are very similar and the stacked section is almost perfect. We are more interested in the result of the angle-gather migration.

Figure 7 shows the zero and large angle sections as well as the stack for angle-gather Kirchhoff migration. The zero-angle section is weak but clearly shows the correct shape and position. The large-angle section is actually only for $\gamma = 25^\circ$. The reason for this is clear if you consider Figure 1. At greater depths, the rays associated with large reflection angles ($\gamma$) will not emerge at the surface within the model space. Therefore at angles greater than $25^\circ$ (the maximum useful angle), the information at later times disappears.

We expect the stacked sections for the offset method and the angle method to be identical. Although we sum over different paths for the offset-domain migration (Figure 2) and the angle-domain migration (Figure 3), the stack should sum all of the same information together for both methods. Fortunately, a comparison of the stacked sections in Figures 6 and 7 show that the results are identical as expected.
Figure 7: Left: Migrated angle section at small angle. Center: Migrated offset section at large angle. Right: Stack.

Dipping reflectors model

This model contains fewer dips than the dome model but it allows us to see what is happening at later times. Figure 8 shows the common offset sections and stacked section from offset-domain Kirchhoff migration. Once again, they are practically perfect. The only problem is near the bottom of the section where we lose energy because the data was truncated.

The zero-angle and large-angle sections from the angle-domain migration are in Figure 9, along with the stacked section. Once again, the zero angle section is very weak and the large angle section only contains information down to a time of \( \approx 0.85 \) seconds, for the same reason as explained for the dome model.

Once again, we expect the stacked sections in Figures 8 and 9 to be the same. Although the angle-domain stack is slightly lower amplitude throughout the section, it is clear that this is a simple scale factor so our expectations remain intact.

Figure 8: Left: Migrated offset section at zero offset. Center: Migrated offset section at far offset. Right: Stack.
Reflectivity variation with angle

Amplitude variation with offset (AVO) would not be expected to be very interesting for the simple models just shown. Consider Figure 10, which contains an offset gather and a reflection angle gather taken from space location zero from the dome model in Figure 4. The offset gather shows exactly what we expect for such a model - no variation. The angle gather also shows no variation for angles less than the maximum useful angle (25°) as discussed in the previous two subsections. However, when the angle exceeds the maximum useful angle, the event increases in amplitude and width. This is the phenomenon seen in de Bruin et al. (1990).

Velocity sensitivity

When dealing with real data we almost never know what the true velocity of the subsurface is. Therefore it is important to understand the effects of velocity on our angle-gather time migration algorithm. To do this we simply created data for the dome model in Figure 4 at a fairly high velocity (3 km/s) and migrated it using a low velocity (1.5 km/s). The results are in Figure 11. For angles less than the maximum
useful angle ($\gamma = 25^\circ$), the angle-domain gather behaves exactly as the offset-domain gather does. Beyond the maximum useful angle, the events become even more curved and the amplitudes begin to change.

The behavior of the angle-gather migration is very similar to that of offset-domain migration as long as the limitation of the maximum useful angle is recognized. Therefore, we can probably expect angle-gather migration to behave like offset-domain migration in $v(z)$ media also.

Figure 11: Gathers taken from space location zero in the dome model and migrated at too low a velocity. Left: Offset domain. Center: Angle domain less than $25^\circ$. Right: Angle domain.

FREQUENCY-DOMAIN CONSIDERATIONS

As pointed out by Prucha et al. (1999), the angle gathers can be conveniently formed in the frequency domain. This conclusion follows from the simple formula (Fomel, 1996a)

$$\tan \gamma = \frac{\partial z}{\partial h},$$

(15)

where $z$ refers to the depth coordinate of the migrated image. In the frequency-wavenumber domain, formula (15) takes the trivial form

$$\tan \gamma = \frac{k_h}{k_z}.$$  

(16)

It indicates that angle gathers can be conveniently formed with the help of frequency-domain migration algorithms (Stolt, 1978). This interesting opportunity requires further research.

CONCLUSIONS

We have presented an approach to time migration based on angle gathers. The output of this procedure are migrated angle gathers - images for constant reflection angles. When stacked together, angle gathers can produce the same output as the
conventional common-offset gathers. Looking at angle gathers individually opens new possibilities for amplitude-versus-angle studies and for velocity analysis.

Our first synthetic tests produced promising results. In the future, we plan to study the amplitude behavior of angle-gather migration and the velocity sensitivity more carefully. We also plan to investigate the frequency-domain approaches to this method. Initial results indicate that angle-gather migration is comparable to offset-domain migration for angles less than the angle at which rays exit the sides of the model, but further study will hopefully allow us to extract useful information from the larger angles as well. Although the major advantages of angle gathers lay in the depth migration domain, it is easier to analyze the time migration results because of their theoretical simplicity.

REFERENCES

——–, 1996a, Migration and velocity analysis by velocity continuation, in SEP-92: Stanford Exploration Project, 159–188.
Wavefront construction using waverays

Hector Urdaneta

ABSTRACT

A method for computing first arrival traveltimes and amplitudes in a general two-dimensional (2-D) velocity model is presented. The method is the result of merging two recently published ray tracing methods. The product is a very robust algorithm that is able to produce broadband wave phenomena, such as dispersion and wavelength dependent scattering. Its ability to produce broadband wave phenomena, is achieved by performing a wavelength-dependent smoothing of the velocity model across wavefronts. In the limit of high frequency, the method reduces to geometrical ray theory. The method is able to illuminate areas of large geometrical spreading where conventional ray tracing methods may give no arrivals. The method is tested on synthetic complex velocity models.

INTRODUCTION

Traditionally traveltime and amplitude calculations have been performed by ray tracing. Different ray tracing algorithms exist that are well known and well documented. They include ray bending (Julian and Gubbins, 1977), shooting rays (Dines and Lytle, 1979) and paraxial extrapolation (Cerveny, 1987). More recently, several new methods have appeared and are enjoying an increasing popularity. They include finite differences (Vidale, 1990; Podvin and Lecomte, 1991; van Trier and Symes, 1991) and shortest path rays (Moser, 1991).

This paper presents a review of two new ray tracing methods and explores some of the possibilities produced by their fusion. The first method is Lomax’s waveray method for approximating broadband wave propagation through complex velocity structures (Lomax, 1994). The second method was developed at the NORSAR institute in Norway by Vinje et al. (1993). As will be shown later, both methods have their own advantages and drawbacks, but when they are fused, they interfere positively. The combined product produces a very robust method, which approximates broadband wave phenomena in complex velocity models.

The first two parts of this paper describe the basic characteristics of each method and their implementations. The paper also reviews some of the work done in the last two references listed above. In the last part, I discuss the combined method. Implementation issues and synthetic examples are shown.

*e-mail: sergey@sep.stanford.edu, marie@sep.stanford.edu
LOMAX’S WAVERAYS

Two basic ideas characterize Lomax’s method: (1) it does a wavelength-dependent velocity smoothing and (2) it uses Huygen’s principle to track the motion of narrow-band wavefronts at a number of center frequencies. Narrow-band wavefronts are defined as surfaces (lines in 2-D) of constant phase or traveltime in a narrow-band “wavefield”. As these narrow-band wavefronts propagate with time they define a wavepath, which is frequency dependent. The wavelength-dependent smoothing of the velocity is done by averaging with a Gaussian weighting curve. The smoothing is done along the wavefronts. The final result; i.e., the broadband wavefield, is constructed by summing the results of independent narrow-band wavefields at many center frequencies.

Three important advantages of using Lomax’s waveray over conventional ray tracing methods are:

- It increases the stability of wavepaths compared to the paths produced by high frequency methods, due to the wavelength-dependent smoothing.
- It provides waverays with a sensitivity that produces frequency dependent scattering as a function of the ratio of wavelength to the characteristic size of the scattering region. Figure 3 illustrates this point.
- It is capable of handling large to small inhomogeneity sizes, since in the former case it is similar to ray theory and in the latter it responds to a smooth, averaged velocity structure.

Lomax (1994) approximates the narrow-band wavefronts at any time by a plane wavefront (see Figure 1). This approximation requires that the radius of curvature of the wavefront be large relative to a wavelength. A better approximation to the wavefronts could probably be obtained using a parabolic approximation. For the sake of computational time, plane wavefronts are used.

Before considering the details of the waveray technique, two points need to be emphasized. First, Lomax (1994) points out, “it is the wavelength dependent smoothing that makes the waveray method a broadband wave propagation technique, and distinguishes it from the high frequency ray methods.” Second, there are no equations that give the waveray method a theoretical basis. Its support comes from the fact that it reproduces high-frequency ray propagation and produces a good approximation of broadband wave phenomena.

Waveray implementation

In the waveray method, the wavelength-dependent averaging of the velocity is done dynamically as a function of the position and orientation of the plane wavefronts. The velocity averaging is done using a Gaussian weight curve, centered at the wavepath.
Figure 1: Waveray wavepath and wavelength-dependent velocity smoothing at point $\mathbf{x}_\nu$. Adapted from Lomax (1994).

location (see Figure 1). Equation 1 expresses the wavelength averaged velocity $\overline{v}$ at a point ($\mathbf{x}_\nu$) for a wave period $T$ (Lomax, 1994),

$$
\overline{v} (\mathbf{x}_\nu, T) = \frac{\int_{-\infty}^{\infty} \omega(\gamma) c [\mathbf{x}(\gamma, T)] d\gamma}{\int_{-\infty}^{\infty} \omega(\gamma) d\gamma}
$$

(1)

where $\gamma$ is the arc length along the wavefront away from $\mathbf{x}_\nu$ expressed in wavelengths. $c (\mathbf{x})$ is the velocity at point $\mathbf{x}$ and $\omega(\gamma)$ is the Gaussian weight curve:

$$
\omega(\gamma) = e^{-4 \ln 2 \cdot (\gamma/\alpha)^2}
$$

(2)

where $\alpha$ specifies the half width of the Gaussian bell in wavelengths.

$(\mathbf{x}(\gamma, T))$ is the position along the instantaneous straight wavefront given by the recursive relation (Lomax, 1994):

$$
\mathbf{x}(\gamma, T) = \mathbf{x}_\nu + \frac{T}{2\pi} \int_{0}^{\gamma} c [\mathbf{x}(\gamma', T)] \mathbf{n}(T) \, d\gamma'
$$

(3)

where $\mathbf{n}$ is the unit normal to the wavepath at point $\mathbf{x}_\nu$.

The discrete representation of equation 1 is given by equation 4:

$$
\overline{v} (\mathbf{x}_\nu, T) = \frac{\sum_{n=-N}^{N} \omega_n c [\mathbf{x}_\nu^n (T)]}{\sum_{n=-N}^{N} \omega_n}
$$

(4)

where the integral has been replaced by a finite sum over $2N + 1$ control points. The position of the control points along the wavefronts are given by equations 5 and 6:

$$
\mathbf{x}_\nu = \mathbf{x}_\nu^0
$$

(5)
\[ \mathbf{x}_\nu^n = \begin{cases} \mathbf{x}_{\nu}^{n-1} + \frac{\gamma_{\text{max}}}{4\pi N} \lambda(\mathbf{x}_\nu^{n-1}) \mathbf{n} & n = 1, 2, \ldots, N, \\ \mathbf{x}_{\nu}^{n+1} - \frac{\gamma_{\text{max}}}{4\pi N} \lambda(\mathbf{x}_\nu^{n+1}) \mathbf{n} & n = -1, -2, \ldots, -N. \end{cases} \] (6)

These two equations are the discrete version of equation 3, but the dependence on the wavelength has been made explicit. Notice that the subscript \( \nu \) of \( \mathbf{x}_\nu^n \) runs along the wavepath and the superscript \( n \) runs along the wavefront. \( \gamma_{\text{max}} \) specifies the largest distance in wavelengths along the wavefront at which smoothing is applied.

The discrete equivalent of the Gaussian weight function is:

\[ \omega_n = e^{-4\ln 2 \cdot \left(\frac{\gamma_n}{\alpha}\right)^2} \] (7)

where the distance \( \gamma_n \) along the wavefront in wavelengths is expressed as:

\[ \gamma_n = n \frac{\gamma_{\text{max}}}{N} \] (8)

The motion of the waverays along the direction of propagation is expressed by the following equation:

\[ \mathbf{x}_{\nu}^{n+1} = \mathbf{x}_{\nu}^n + \mathbf{v}_\nu \Delta t \mathbf{s} \] (9)

where \( \mathbf{v}_\nu = \mathbf{v}(\mathbf{x}_\nu, T), \Delta t \) is the time step and \( \mathbf{s} \) is a unit vector that moves along the direction of propagation.

The change in direction of the waverays is approximated by the difference in movement between the first control point on either side of the wave location \( \mathbf{x}_\nu \) as shown in Figure 2:

\[ \Delta \mathbf{s} = -\frac{(\mathbf{v}_\nu^1 - \mathbf{v}_\nu^{-1})}{|\mathbf{x}_\nu^1 - \mathbf{x}_\nu^{-1}|} \Delta t \mathbf{n} \] (10)

Figure 2: Waveray wavepath calculation. Huygen’s principle is used to obtain the bending \( \Delta \mathbf{s} \) of the wavepath from points \( \mathbf{x}_\nu^1 \) and \( \mathbf{x}_\nu^{-1} \). Adapted from Lomax (1994).
where $\bar{v}^1_\nu$ and $\bar{v}^{-1}_\nu$ are the wavelength averaged velocities at the wavefront points $\vec{x}^1_\nu$ and $\vec{x}^{-1}_\nu$ respectively.

Finally, the half width parameter $\alpha$ and the truncation parameter $\gamma_{\text{max}}$ are set at $\alpha = 2.0$ and $\gamma_{\text{max}} = 1.5$, based on Lomax (1994) calibration. The number of control points $N$ is set proportional to the ratio $T/\Delta t$ of the wave period over the time step.

Figure 3 shows the significant differences between the waveray and ray methods. Notice how a high frequency ray is scattered by the small velocity anomaly, while the waveray’s wavepath is little deflected. Note also, how the third ray (from right to left) is not perturbed by the low velocity anomaly, while the waveray wavepath is deflected. The wavelength-dependent velocity averaging smooths out small velocity variations and causes the wavepath to be affected from velocity variations away from it.

Figure 3: Left frame shows a fan of high frequency ray paths. Right frame shows a fan of 12 Hz waveray wavepaths. The straight segments perpendicular to the waverays represent the instantaneous wavefronts. The velocity model is defined by two circular anomalies drawn in a homogeneous background. The black dot (located at 1000 m. by 1250 m. in depth) depicts a low velocity anomaly. The white circle a high one.

NORSAR WAVEFRONT CONSTRUCTION

The main idea behind the “NORSAR” method (Vinje et al., 1993) is to compute ray parameters along wavefronts instead of computing them from independently traced rays, as conventional ray methods do. Wavefronts are defined as isochron traveltime curves (lines in 2-D) from the source. New wavefronts are constructed from previous ones, by ray tracing over a time step. As wavefronts expand out, new rays are interpolated between rays that go further apart than a predefined distance $D_{\text{Smax}}$. Figure 4 illustrates how wavefronts expand out, by ray tracing from a time $\tau$ to a time $\tau + \Delta \tau$. The dashed line on Figure 4 represents the new wavefront. The solid dots
represent the end points of the rays. The distance between contiguous end points is checked against the predefined maximum distance $DS_{max}$. If they are located further than $DS_{max}$, a new point (empty dots) is interpolated.

The interpolation of these new points over the wavefront is done using a vectorial third order polynomial $\tilde{x}(s) = \tilde{a}s^3 + \tilde{b}s^2 + \tilde{c}s + \tilde{d}$. The polynomial is evaluated as a function of the normalized distance $s$ between points $\tilde{x}^i$ and $\tilde{x}^{i+1}$. Also a scalar third order polynomial is used to interpolate amplitude values and the ray’s angle of direction [see Vinje et al. (1993)].

The key property of this procedure is that it produces a fairly constant density of rays over C1 models [Vinje et al. 1993] (see Figure 8), illuminating zones with high geometrical spreading where conventional ray tracing have shadow zones.

Figure 4: New wavefronts (dashed lines) are constructed from the previous wavefront (solid line), by ray tracing a fix number of time steps. New rays are interpolated between points on the wavefront that lay further than a predefined distance. Adapted from Vinje et al. (1993).

Rays are eliminated if they go out of the model boundaries. They may also be eliminated if a wavefront crosses over itself, as shown on Figure 5. The “self-crossing” of the wavefronts may correspond to a caustic or to the intersection of rays from different parts of the model. Once again, as in the Lomax algorithm, for the sake of computational time and also of memory, a “first arrival” mode should be used. This first arrival mode removes all later arrivals. When the number of points in a wavefront becomes less than a certain value (e.g. 4 points), the algorithm stops.

Traveltimes and amplitudes are interpolated into a rectangular grid. Ray cells, defined as the area enclosed by a pair of contiguous rays and wavefronts, are checked for the presence of grid points (see Figure 6). Traveltimes at the receivers are estimated by computing the following quantities:

1. the distances $d_1$ and $d_2$ from the receiver perpendicular to the two rays.

2. the normalized distance $s$ along the wavefront $s = d_1/(d_1 + d_2)$. 
3. the interpolated point $\tilde{x}(s)$ over the old wavefront.

4. the distance $l_r$ from $\tilde{x}(s)$ to the receiver

5. the velocity $v_{mid}$ in the midpoint of the segment $l_r$.

The traveltime at the receiver is then estimated to be:

$$t_{rec} = t + \frac{l_r}{v_{mid}}$$  \hspace{1cm} (11)

where $t$ is the traveltime to the old wavefront.

Figure 5: The new wavefront crosses itself. If only first arrivals are wanted, the points behind the crossing (points no. 7, 8, 9) are removed from the wavefront. Adapted from Vinje et al. (1993).

In computing amplitudes, the geometrical spreading factor $\sqrt{(r_1 + r_2)/(R_1 + R_2)}$, gives the ratio between the amplitude of one wavefront to the next one. $R_1$, $R_2$, $r_1$ and $r_2$ are shown on Figure 7. The amplitude estimation at the receivers is also obtained in this way, where the distances $d_1$ and $d_2$ are used for $R_1$ and $R_2$.

Figure 8 shows an example of the Norsar method run over a highly contrasted velocity model. The velocity model is a pair of Gaussian bell curves. The distance between the peaks is 48 meters with a drop of 4 km/s in velocity.

A final point on Norsar’s method is, as said by Vinje et al. (1993): “the way the ray tracing between each wavefront is performed is irrelevant to the idea of the wavefront construction”. We notice that all along the discussion on the NORSAR method, ray tracing was kept as an abstract idea. With this in mind we proceed to merge the Lomax algorithm, as the ray tracing algorithm for the NORSAR method. Another advantage of the NORSAR method is that the estimation of ray parameters (as traveltimes, amplitudes, etc.) does not come from a posteriori interpolation between single, separate rays, but instead directly from previously constructed wavefronts.
Figure 6: Traveltimes and amplitudes are found at receivers by interpolating within each ray cell. The ray cell is defined by $Ray_1$ and $Ray_2$, and by the new wavefront and the previous wavefront. Adapted from Vinje et al. (1993).

Figure 7: Amplitudes are computed from the previous wavefront. The geometrical spreading factor gives the ratio between the amplitude $A_i$ at the previous wavefront and the new amplitude value $A_{i+1}$. 
Figure 8: The “NORSAR” or wavefront construction method covers the whole model with wavefronts and rays. The model presents a strong variation in velocity, given by a Gaussian bell with a maximum amplitude of 5000 m/s and a second Gaussian bell with a minimum amplitude of 1000 m/s. The background is 3000 m/s.
WAVERAYS AND WAVEFRONTS

The product of merging the two previously discussed ray tracing methods is sketched in the following pseudo-code:

1 Allocate memory for wavefronts and output
2 for each source
  3 initialize
4 while number of points in wavefront > 4
  5 propagate wavefront
  6 check for self crossing
  7 check for rays that go out of bounds
  8 eliminate self crossing and outofbound rays
  9 calculate amplitudes
10 gridding
11 interpolate new rays
12 movwavf

The algorithm defines the data structure “cube” at line 1. In it, the ray parameters are stored as wavefronts propagate.

```
struct heptagon {
  struct point {
    float x;
    float z; } x0, x1;
  float angle;
  float ampl;
  char cf; } *cube;
```

where:

- $cube[ii].x0$ contains the starting position of a ray on a wavefront at a time $\tau$.
- $cube[ii].x1$ contains the arriving position of a ray on a wavefront at a time step later, $\tau + \Delta\tau$.
- $cube[ii].angle$ gives the arriving angle at point $cube[ii].x1$.
- $cube[ii].ampl$ is the computed amplitude at point $cube[ii].x1$.
- $cube[ii].cf$ is a flag that defines if point $cube[ii].x1$ is in bounds, out of bounds, or belongs to the inner section of a self crossing wavefront.
The index \( ii \) runs over the points of a wavefront. Since there is no a priori way of determining how large can a wavefront grow over a velocity model, a predefined limit (\( nrmax \)) of \( cube \) elements is established at the beginning of the algorithm, which defines the total memory allocated for \( cube \). In other words, \( nrmax \) is the maximum number of points that a wavefront can have. If the wavefront grows bigger than \( nrmax \) points the algorithm is stopped and an error message is produce indicating that a bigger value for \( nrmax \) should be used. This integer depends directly on the maximum allowed separation \( DSmax \) between two contiguous points in the wavefront. For the examples shown on Figure 9 through 12, the program ran with a value of \( nrmax=1300 \). \( DSmax \) was set to 21 meters for those examples.

In the case that the algorithm is run in a "all arrivals" mode, which could be done by eliminating line 6 out of the algorithm, the number of crossing points on a wavefront could become considerably large. As the wavefronts crosses and crosses many times over itself, for a velocity model with strong variations (as for example the Marmousi model), the number of crossing points can easily reach the 6 digits figure. This translates directly into a bigger need of computer resources, in use of memory and time.

Subroutine \textit{initialize} defines the initial wavefront. It assigns an initial amplitude and take-off angle to the points on the initial wavefront.

The \textit{propagate wavefront} subroutine ray traces using Lomax’s waverays. The waverays are traced starting at \( cube[ii].x0 \) with a take-off angle \( cube[ii].angle \) during one time step at a certain frequency. The time step and the frequency are user predefined.

Subroutine on line 6 checks for self crossed wavefronts and flags the points that belong to the inner crossed section of the wavefront.

Line 7 checks for points that fall out of boundaries, raising a flag. Notice from Figure 8 that the rays cross over the boundaries of the model. This is done in order to obtain arrivals at the receivers that lie on the boundaries.

Subroutine at line 8 eliminates the points on a wavefront that are flagged for laying out of bounds or belong to self crossed wavefronts.

Subroutines on lines 9, 10 and 11 are implemented as previously explained for the NORSAR method on Figures 7, 6 and 4 respectively. On line 11 the number of rays that may be interpolated between any two contiguous rays, is given by the number of times the distance between the two rays is bigger than the maximum allowed distance \( DSmax \).

Subroutine \textit{gridding} is a very time consuming, due to the irregular distribution of the data in the model. First, the subroutine checks for receivers inside the ray cells of two contiguous wavefronts. If a receiver is found, the ray parameters are interpolated to it.

Subroutine \textit{movwavf} prepares the structure \( cube \) for a new wavefront to be propagated. It takes as the new starting point the previous arriving point (\( cube[ii].x0 = \)
Travel-times and amplitudes in the Marmousi model

Figure 9: Traveltime contours computed from the combined method, overlaid on a section of the Marmousi model. A frequency of 80 Hz is used.

Figures 9 to 12 display the results of a simulation that used the combined method. The underlying subsurface structure is the Marmousi model (Versteeg, 1993). A source was put at the surface, 5200 meters away from the left edge of the model, and the wavefronts were propagated until they crossed the boundaries of the model. Figure 9 shows the first-arrival traveltime contours calculated at a frequency of 80 Hz. Figure 10 shows the same experiment at a frequency of 10 Hz. Not much difference is apparent.

Figure 11 shows the amplitude estimates for the 80 Hz shot and Figure 12 shows the 10 Hz estimates. We see that more energy gets propagated down in the case of the low frequency, illuminating part of the high frequency shadow zones.

We have seen that the combined method accomplishes two important tasks, it can be used to compute first arrival traveltimes and amplitudes over any general velocity model and is it able to illuminate high frequency shadow zones.

For these experiments, the mesh of the model is re-sampled from the original model at 8 x 8 meters. The traveltimes and amplitude outputs are placed in a mesh of 25 x 12.5 meters.
Figure 10: As in the previous Figure, but for a frequency of 10 Hz.

Figure 11: Amplitude maps from the combined method at a frequency of 80 Hz.
CONCLUSIONS

I have presented a review of two ray tracing methods. I have implemented both of them in a combined version. The method computes first arrival travel-times and amplitudes of seismic waves in complex 2-D velocity structures. The method uses a wavefront construction technique that produces a complete coverage of the medium by a fairly constant density of wavefronts and rays. Wavefronts are propagated using a wavelength-dependent smoothing ray tracing technique, called the waveray method, which leads to an increased stability of the ray paths relative to high frequency rays. Also, it gives a sensitivity to the rays to larger velocity anomalies that lay within a fraction of a wavelength of the ray path. The data (traveltimes and amplitudes) is computed on an irregular grid. As the wavefronts are constructed the data is interpolated into a regular grid.

The result is a very robust ray tracing method that is able to illuminate areas of large geometrical spreading zones where conventional ray tracing methods produce shadow zones. Portions of the diffracted energy is produced in these shadow zones.

Further work should be done on calibrating and testing the results produced by the combined method against other methods. Future work should be done on a formal derivation of the waveray method. Production of seismograms and a 3-D version are also sources of future work.
REFERENCES


INTRODUCTION

Traveltime computation is an important part of seismic imaging algorithms. Conventional implementations of Kirchhoff migration require precomputing traveltime tables or include traveltime calculation in the innermost computational loop. The cost of traveltime computations is especially noticeable in the case of 3-D prestack imaging where the input data size increases the level of nesting in computational loops.

The eikonal differential equation is the basic mathematical model, describing the traveltime (eikonal) propagation in a given velocity model. Finite-difference solutions of the eikonal equation have been recognized as one of the most efficient means of traveltime computations \cite{Vidale1990, vanTrierSymes1991, Popovici1991}. The major advantages of this method in comparison with ray tracing techniques include an ability to work on regular model grids, a complete coverage of the receiver space, and a fair numerical robustness. The most common implementations of the finite-difference eikonal equation compute the first-arrival traveltimes, though frequency-dependent enhancements \cite{Biondi1992, Nichols1994} can extend the method to computing the most energetic arrivals. The major numerical complexity of the finite-difference eikonal computations arises from the fundamental non-linearity of the eikonal equation. The numerical complexity is related not only to the direct cost of the computation, but also to the accuracy and stability of finite-difference schemes.

It is important to note that the current practice of seismic imaging is not limited to a single migration. Moreover, it is repeated migrations, with velocity analysis and refinement of the velocity model at each step, that take most of the computational effort. When the changes in the velocity model at each step are small compared to the initial model, it is appropriate to linearize the eikonal equation with respect to the slowness and traveltime perturbations. Mathematically, the linearized eikonal equation corresponds precisely to the linearization assumption, commonly used in traveltime tomography.

In this paper, I propose an algorithm of finite-difference traveltime computations, based on an iterative linearization of the eikonal equation. The algorithm takes advantage of an implicit finite-difference scheme with superior stability and accuracy properties. I test the algorithm on a simple synthetic example and discuss its possible applications in residual traveltime computation, interpolation, and tomography.

THE LINEARIZED EIKONAL EQUATION

The eikonal equation, describing the traveltime propagation in an isotropic medium, has the form

\[ (\nabla \tau)^2 = n^2(x, y, z), \]

where \( \tau(x, y, z) \) is the traveltime (eikonal) from the source to the point with the coordinates \( (x, y, z) \), and \( n \) is the slowness at that point (the velocity \( v \) equals \( 1/n \)). In Appendix A, I review a basic derivation of the eikonal and transport equations. To
formulate a well-posed initial-value problem on equation (1), it is sufficient to specify \( \tau \) at some closed surface and to choose one of the two branches of the solution (the wave going from or to the source.)

Equation (1) is nonlinear. The nonlinearity is essential for producing multiple branches of the solution. Multi-valued eikonal solutions can include different types of waves (direct, reflected, diffracted, head, etc.) as well as different branches of caustics. To linearize equation (1), we need to assume that an initial estimate \( \tau_0 \) of the eikonal \( \tau \) is available. The traveltime \( \tau_0 \) corresponds to some slowness \( n_0 \), which can be computed from equation (1) as

\[
    n_0 = |\nabla\tau_0|.
\]  

(2)

Let us denote the residual traveltime \( \tau - \tau_0 \) by \( \tau_1 \) and the residual slowness \( n - n_0 \) by \( n_1 \). With these definitions, we can rewrite equation (1) in the form

\[
    (\nabla\tau_0 + \nabla\tau_1)^2 = (\nabla\tau_0)^2 + 2 \nabla\tau_0 \cdot \nabla\tau_1 + (\nabla\tau_1)^2 = (n_0 + n_1)^2 = n_0^2 + 2 n_0 n_1 + n_1^2 , \tag{3}
\]

or, taking into account equality (2).

\[
    2 \nabla\tau_0 \cdot \nabla\tau_1 + (\nabla\tau_1)^2 = 2 n_0 n_1 + n_1^2 . \tag{4}
\]

Neglecting the squared terms, we arrive at the equation

\[
    \nabla\tau_0 \cdot \nabla\tau_1 = n_0 n_1 , \tag{5}
\]

which is the linearized version of the eikonal equation (1). The accuracy of the linearization depends on the relative ratio of the slowness perturbation \( n_1 \) and the true slowness model \( n \). Though it is difficult to give a quantitative estimate, the ratio of 10% is generally assumed to be a safe upper bound.

The intimate connection of the linearized eikonal equation and traveltime tomography is discussed in Appendix B.

**ALGORITHM**

Linearization of the eikonal equation suggests the following algorithm of traveltime computation:

1. Start with an initial traveltime field \( \tau_0 \). The initial traveltime may be the result of a previous computation or (for simple models) the result of an approximate analytic evaluation.

2. Compute the finite-difference gradient \( \nabla\tau_0 \) and the corresponding slowness model \( n_0 \) with equation (2).

3. Compute the slowness perturbation \( n_1 \) as the difference between the true slowness model \( n \) and \( n_0 \). Exit the computation if the perturbation is smaller than the desired accuracy.
4. Solve numerically equation (5) for the traveltime perturbation $\tau_1$.

5. Update the traveltime field $\tau_0$ by adding $\tau_1$ to it.

6. Repeat the loop.

Equation (5) can be solved numerically with a simple explicit upwind finite-difference method.

**NUMERICAL TEST**

For the first numerical test, I used a model with a smooth anomaly inside a constant slowness background. The initial traveltime was computed analytically, using the background slowness. The result of the computation is shown in Figure 1. The computation involved 5 re-linearization cycles.

![Figure 1: The traveltime contours for a smooth anomaly, computed by the linearized eikonal solver. The background slowness is 1 s/km. The maximum anomaly slowness is 2.25 s/km. The wave source is in on the top plane of the model.](image)

The result shows the expected behavior of the wavefronts. It agrees with the result of a direct eikonal computation, shown in Figure 2. The direct computation was done with Mihai Popovici’s TTGES eikonal solver, which has outstanding efficiency and stability properties. Obviously, more tests are required to evaluate the comparative performance of the algorithm and the limits of its practical applicability. The discussion section contains some speculations about the perspective usage of the linearized algorithm.
DISCUSSION

Although the first numerical experiments have been too incomplete for drawing any solid conclusions, it is interesting to discuss the possible applications of the linearized eikonal.

Multi-valued traveltimes Conventional eikonal solvers usually force the choice of a particular branch of the multi-valued traveltime, most commonly the first-arrival branch. However, in some cases other branches may in fact be more useful for imaging or velocity estimation [Gray and May, 1994]. When the linearization assumption is correct, the linearized eikonal should follow the branch of the initial traveltime. This branch does not have to be the first arrival. It can correspond to any other arrival, such as reflected waves or multiple reflections.

Spherical Coordinates Though the eikonal equation itself does not favor any particular direction, its solution for the case of a point source lands more naturally into a spherical coordinate system. van Trier and Symes [1991], Popovici (1991), Fowler [1994], and Schneider (1995) presented upwind finite-difference eikonal schemes based on a spherical computational grid. To use the linearized equation [5] on such a grid, it is necessary to rewrite the gradient operator in the spherical coordinates, as follows:

$$
\nabla \tau = \left\{ \frac{\partial \tau}{\partial r}, \frac{1}{r} \frac{\partial \tau}{\partial \theta}, \frac{1}{r \sin^2 \theta} \frac{\partial \tau}{\partial \phi} \right\}.
$$
Interpolation One of the most natural applications for the linearized eikonal is interpolation of traveltimes. Interpolating regularly gridded input (such as subsampled traveltime tables) reduces to masked inversion of equation (5). Interpolating irregular input (such as the result of a ray tracing procedure) reduces to regularized inversion. In both cases, a simpler way of traveltime binning would be required to initiate the linearization.

Tomography Tomographic velocity estimation is possible when the input traveltime data corresponds to a collection of sources. In this case, we can reduce the linearized traveltime inversion to the system of equations

\[ n_0^{(1)} \cdot \nabla \tau_1^{(1)} = n_0^{(2)} \cdot \nabla \tau_1^{(2)} = \cdots = s_1. \]  

Here, \( \tau_1^{(i)} \) stands for the traveltime from source \( i \). Equations (6) are additionally constrained by the known values of the traveltime fields at the receiver locations.

Amplitudes The amplitude transport equation, briefly reviewed in Appendix A, has the form (A-4). Introducing the logarithmic amplitude \( J = -\ln(A/A_0) \), where \( A_0 \) is the constant reference, we can rewrite this equation in the form

\[ 2 \nabla \tau \cdot \nabla J = \Delta \tau. \]  

The left-hand side of equation (7) has exactly the same form as the left-hand side part of the linearized eikonal equation (5). This suggests reusing the traveltime computation scheme for amplitude calculations. The amplitude transport equation is linear. However, it explicitly depends on the traveltime. Therefore, the amplitude computation needs to be coupled with the eikonal solution.

Anisotropy In a recent paper, Alkhalifah (1997) proposed a simple eikonal-type equation for seismic imaging in vertically transversally-isotropic media. Alkhalifah’s equation should be suitable for linearization, either in the normal moveout velocity \( V_{NMO} \) or in the dimensionless anisotropy parameter \( \eta \). This untested opportunity looks promising because of the validity of the weak anisotropy assumption in many regions of the world.

CONCLUSIONS

I have presented a finite-difference method of traveltime computations, based on the linearized eikonal equation. Preliminary numerical experiments show that the method is as simple and robust as can be expected from the theory. The required assumption is that a reasonable estimate of the traveltime is available prior to linearization. Such an estimate may result from the computation in a different velocity model, with a different method (e.g., ray tracing), or by an analytic evaluation.

In the situations where the underlying assumption is valid, the linearized approach may allow us
to employ unconditionally stable implicit finite-difference schemes with an easy control of the numerical stability,

• to parallelize the essential parts of the algorithm with minimum effort,

• to compute branches of the multi-valued traveltime other than the first arrival,

• to connect traveltime computations with tomographic velocity estimation,

• to couple traveltime and amplitude computations.

Future research is necessary to confirm these expectations.

ACKNOWLEDGMENTS

I thank Biondo Biondi for interesting discussions on the traveltime computation problem and its applications. The TTGES eikonal solver that I used for testing was kindly made available to SEP by Mihai Popovici of 3DGeo.

REFERENCES

Schneider, W. A., 1995, Robust and efficient upwind finite-difference traveltime calculations in three dimensions: Geophysics, 60, 1108–1117.
APPENDIX A

A SIMPLE DERIVATION OF THE EIKONAL AND TRANSPORT EQUATIONS

In this Appendix, I remind the reader how the eikonal equation is derived from the wave equation. The derivation is classic and can be found in many popular textbooks. See, for example, Červeny et al. [1977].

Starting from the wave equation,

\[ \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2} = n^2(x, y, z) \frac{\partial^2 P}{\partial t^2}, \]  \hspace{1cm} (A-1)

we introduce a trial solution of the form

\[ P(x, y, t) = A(x, y, z) f(t - \tau(x, y, z)), \]  \hspace{1cm} (A-2)

where \( \tau \) is the eikonal, and \( A \) is the wave amplitude. The waveform function \( f \) is assumed to be a high frequency (discontinuous) signal. Substituting solution (A-2) into equation (A-1), we arrive at the constraint

\[ \Delta A f - 2 \nabla A \cdot \nabla \tau f' - A \Delta \tau f' + A (\nabla \tau)^2 f'' = n^2 A f''. \]  \hspace{1cm} (A-3)

Here \( \Delta \equiv \nabla^2 \) denotes the Laplacian operator. Equation (A-3) is as exact as the initial wave equation (A-1) and generally difficult to satisfy. However, we can try to satisfy it asymptotically, considering each of the high-frequency asymptotic components separately. The leading-order component corresponds to the second derivative of the wavelet \( f'' \). Isolating this component, we find that it is satisfied if and only if the traveltime function \( \tau(x, y, z) \) satisfies the eikonal equation [1].

The next asymptotic order corresponds to the first derivative \( f' \). It leads to the amplitude transport equation

\[ 2 \nabla A \cdot \nabla \tau + A \Delta \tau = 0. \]  \hspace{1cm} (A-4)

The amplitude, defined by equation (A-4), is often referred to as the amplitude of the zero-order term in the ray series. A series expansion of the function \( f \) in high-frequency asymptotic components produces recursive differential equations for the terms of higher order. In practice, equation (A-4) is sufficiently accurate for describing the major amplitude trends in most of the cases. It fails, however, in some special cases, such as caustics and diffraction.
APPENDIX B

CONNECTION OF THE LINEARIZED EIKONAL EQUATION AND TRAVELTIME TOMOGRAPHY

The eikonal equation \( (1) \) can be rewritten in the form

\[
\mathbf{n} \cdot \nabla \tau = n , \tag{B-1}
\]

where \( \mathbf{n} \) is the unit vector, pointing in the traveltime gradient direction. The integral solution of equation \( (B-1) \) takes the form

\[
\tau = \int_{\Gamma(n)} n dl , \tag{B-2}
\]

which states that the traveltime \( \tau \) can be computed by integrating the slowness \( n \) along the ray \( \Gamma(n) \), tangent at every point to the gradient direction \( \mathbf{n} \).

Similarly, we can rewrite the linearized eikonal equation \( (5) \) in the form

\[
\mathbf{n}_0 \cdot \nabla \tau_1 = n_1 , \tag{B-3}
\]

where \( \mathbf{n}_0 \) is the unit vector, pointing in gradient direction for the initial traveltime \( \tau_0 \). The integral solution of equation \( (B-3) \) takes the form

\[
\tau_1 = \int_{\Gamma(n_0)} n_1 dl , \tag{B-4}
\]

which states that the traveltime perturbation \( \tau_1 \) can be computed by integrating the slowness perturbation \( n_1 \) along the ray \( \Gamma(n_0) \), defined by the initial slowness model \( n_0 \). This is exactly the basic principle of traveltime tomography.

I have borrowed this proof from Lavrentiev et al. (1970), who used linearization of the eikonal equation as the theoretical basis for traveltime inversion.
Huygens wavefront tracing:
A robust alternative to conventional ray tracing

Paul Sava and Sergey Fomel

ABSTRACT
We present a method of ray tracing that is based on a system of differential equations equivalent to the eikonal equation, but formulated in the ray coordinate system. We use a first-order discretization scheme that is interpreted very simply in terms of the Huygens’ principle. The method has proved to be a robust alternative to conventional ray tracing, while being faster and having a better ability to penetrate the shadow zones.

INTRODUCTION
Though traveltime computation is widely used in seismic modeling and routine data processing, attaining sufficient accuracy without compromising speed and robustness is problematic. Moreover, there is no easy way to obtain the traveltimes corresponding to the multiple arrivals that appear in complex velocity media.

The tradeoff between speed and accuracy becomes apparent in the choice between the two most commonly used methods, ray tracing and numerical solutions to the eikonal equation. Other methods reported in the literature (dynamic programming [Moser 1991], wavefront construction [Vinje et al. 1993], etc.) are less common in practice [Audebert et al., 1994].

Eikonal solvers provide a relatively fast and robust method of traveltime computations [Vidale 1990, van Trier and Symes 1991]. They also avoid the problem of traveltime interpolation to a regular grid which imaging applications require. However, the eikonal solvers compute first-arrival traveltimes and lack the important ability to track multiple arrivals. In complex velocity structures, the first arrival does not necessarily correspond to the most energetic wave, and other arrivals can be crucially important for accurate modeling and imaging [Geoltrain and Brad 1993, Gray and May 1994].

On the other hand, one-point ray tracing can compute multiple arrivals with great accuracy. Unfortunately, it lacks the robustness of eikonal solvers. Increasing the accuracy of ray tracing in the regions of complex velocity variations raises the cost of the method and makes it prohibitively expensive for routine large-scale applications. Mathematically, ray tracing amounts to a numerical solution of the initial value

*e-mail: paul@sep.stanford.edu, sergey@sep.stanford.edu

395
problem for a system of ordinary differential equations \[\text{[Cerveny, 1987]}\]. These ray equations describe characteristic lines of the eikonal partial differential equation.

Here, we propose a somewhat different approach to traveltime computation, that is both fast and accurate, and has the ability to find multiple arrival traveltimes. The theoretical construction is based on a system of differential equations, equivalent to the eikonal equation, but formulated in the ray coordinate system. Unlike eikonal solvers, our method produces the output in ray coordinates. Unlike ray tracing, it is computed by a numerical solution of partial differential equations. We show that the first-order discretization scheme has a remarkably simple interpretation in terms of the Huygens’ principle and propose a Huygens wavefront tracing (from now on referred to as \textit{HWT}) scheme as a robust alternative to conventional ray tracing. Numerical examples demonstrate the following properties of the method: stability in media with strong and sharp lateral velocity variations, better coverage of the shadow zones, and greater speed than paraxial ray tracing (from now on referred to as \textit{PRT}).

**CONTINUOUS THEORY**

The eikonal equation, governing the traveltimes from a fixed source in an isotropic heterogeneous medium, has the form

\[
\left( \frac{\partial \tau}{\partial x} \right)^2 + \left( \frac{\partial \tau}{\partial z} \right)^2 = \frac{1}{v^2(x, y, z)}.
\]

(1)

Here \(x, y, \text{ and } z\) are spatial coordinates, \(\tau\) is the traveltime (eikonal), and \(v\) stands for the velocity field. Constant-traveltime contours in the traveltime field \(\tau(x, y, z)\), constrained by equation (19) and appropriate boundary conditions, correspond to wavefronts of the propagating wave. Additionally, each point on a wavefront can be parameterized by an arbitrarily chosen ray parameter \(\gamma\). In three dimensions, \(\gamma\) includes a pair of independent parameters. For brevity, from now on we will restrict the analysis to two dimensions. One can easily generalize it to the 3-D case by considering \(\gamma\) and \(x\) as vector quantities. Thus, we will refer to the following two-dimensional form of equation (19):

\[
\left( \frac{\partial \tau}{\partial x} \right)^2 + \left( \frac{\partial \tau}{\partial z} \right)^2 = \frac{1}{v^2(x, z)}.
\]

(2)

For a point source, \(\gamma\) can be chosen as the initial ray angle at the source. Zhang \(\text{[1993]}\) shows that \(\gamma\) as a function of spatial coordinates satisfies the simple partial differential equation

\[
\frac{\partial \tau}{\partial x} \frac{\partial \gamma}{\partial x} + \frac{\partial \tau}{\partial z} \frac{\partial \gamma}{\partial z} = 0.
\]

(3)

Equation (67) merely expresses the fact that in an isotropic medium, rays are locally orthogonal to wavefronts. The field \(\gamma(x, z)\) has not only theoretical interest as it provides one of the possible ways for evaluating propagation amplitudes. In particular,
the geometrical spreading factor \( J(x, z) \) is connected to \( \gamma \) by the simple relationship 
\[
\left( \frac{\partial \gamma}{\partial x} \right)^2 + \left( \frac{\partial \gamma}{\partial z} \right)^2 = \frac{1}{J^2(x, z)} .
\]  
(4)

It is important to note that for complex velocity fields, both \( \tau \) and \( \gamma \) as functions of \( x \) and \( z \) become multi-valued. In this case, the multi-valued character of the ray parameter \( \gamma \) corresponds to the situation, where more than one ray from the source passes through a particular point \( \{x, z\} \) in the subsurface. This situation presents a very difficult problem when equations (2) and (67) are solved numerically. Typically, only the first-arrival branch of the travelt ime is picked in the numerical calculation. The ray tracing method is free from that limitation because it operates in the ray coordinate system. Ray tracing computes the travelt ime \( \tau \) and the corresponding ray positions \( x \) and \( z \) for a fixed ray parameter \( \gamma \).

Since \( x(\tau, \gamma) \) and \( z(\tau, \gamma) \) are uniquely defined for arbitrarily complex velocity fields, we can now make an important mathematical transformation. Considering equations (2) and (67) as a system and applying the general rules of calculus, we can transform this system by substituting the inverse functions \( x(\tau, \gamma) \) and \( z(\tau, \gamma) \) for the original fields \( \tau(x, z) \) and \( \gamma(x, z) \). The resultant expressions take the form

\[
\left( \frac{\partial x}{\partial \tau} \right)^2 + \left( \frac{\partial z}{\partial \tau} \right)^2 = v^2 (x(\tau, \gamma), z(\tau, \gamma))
\]  
(5)

and

\[
\frac{\partial x}{\partial \tau} \frac{\partial x}{\partial \gamma} + \frac{\partial z}{\partial \tau} \frac{\partial z}{\partial \gamma} = 0 .
\]  
(6)

Comparing equations (5) and (6) with the original system (2-67) shows that equations (5) and (6) again represent the dependence of ray coordinates and Cartesian coordinates in the form of partial differential equations. However, the solutions of system (5-6) are better behaved and have a unique value for every \( \tau \) and \( \gamma \). These values can be computed with the conventional ray tracing. However, the ray-tracing approach is based on a system of ordinary differential equations, which represents a different mathematical model.

We use equations (5) and (6) as the basis of our wavefront tracing algorithm. The next section discusses the discretization of the differential equations and the physical interpretation we have given to the scheme.

**A DISCRETIZATION SCHEME AND THE HUYGENS’ PRINCIPLE**

A natural first-order discretization scheme for equation (5) leads to the difference equation

\[
(x_{j+1}^i - x_j^i)^2 + (z_{j+1}^i - z_j^i)^2 = (r_j^i)^2 ,
\]  
(7)
where the index \(i\) corresponds to the ray parameter \(\gamma\), \(j\) corresponds to the traveltime \(\tau\), \(r^i_j = \Delta \tau v_j^i\), \(\Delta \tau\) is the increment in time, and \(v_j^i\) is the velocity at the \(\{i, j\}\) grid point. It is easy to notice that equation \(7\) simply describes a sphere (or a circle in two dimensions) with the center at \(\{x_j^i, z_j^i\}\) and the radius \(r_j^i\). This sphere is, of course, the wavefront of a secondary Huygens source.

This observation suggests that we apply the Huygens’ principle directly to find an appropriate discretization for equation \(6\). Let us consider a family of Huygens spheres, centered at the points along the current wavefront. Mathematically, this family is described by an equation analogous to \(7\), as follows:

\[
(x - x(\gamma))^2 + (z - z(\gamma))^2 = r^2(\gamma). \tag{8}
\]

Here the ray parameter \(\gamma\) serves as the parameter that distinguishes a particular Huygens source. According to the Huygens’ principle, the next wavefront corresponds to the envelop of the wavefront family. To find the envelop condition, we can simply differentiate both sides of equation \(8\) with respect to the family parameter \(\gamma\). The result takes the form

\[
(x(\gamma) - x)x'(\gamma) + (z(\gamma) - z)z'(\gamma) = r(\gamma)r'(\gamma), \tag{9}
\]

which is clearly a semidiscrete analog of equation \(6\). To complete the discretization, we can represent the \(\gamma\)-derivatives in \(9\) by a centered finite-difference approximation. This representation yields the scheme

\[
(x_j^i - x_{j+1}^i) (x_j^{i+1} - x_j^{i-1}) + (z_j^i - z_{j+1}^i) (z_j^{i+1} - z_j^{i-1}) = r_j^i (r_j^{i+1} - r_j^{i-1}), \tag{10}
\]

which supplements the previously found scheme \(7\) for a unique determination of the point \(\{x_{j+1}^i, z_{j+1}^i\}\) on the \(i\)-th ray and the \((j+1)\)-th wavefront. Formulas \(7\) and \(10\) define an update scheme, depicted in Figure 1. To fill the \(\{\tau, \gamma\}\) plane, the scheme needs to be initialized with one complete wavefront (around the wave source) and two boundary rays.

The solution of system \(7,10\) has the explicit form

\[
x_{j+1}^i = x_j^i - r_j^i \left( \alpha (x_j^{i+1} - x_j^{i-1}) \mp \beta (z_j^{i+1} - z_j^{i-1}) \right), \tag{11}
\]

\[
z_{j+1}^i = z_j^i - r_j^i \left( \alpha (z_j^{i+1} - z_j^{i-1}) \mp \beta (x_j^{i+1} - x_j^{i-1}) \right), \tag{12}
\]

where

\[
\alpha = \frac{r_j^{i+1} - r_j^{i-1}}{(x_j^{i+1} - x_j^{i-1})^2 + (z_j^{i+1} - z_j^{i-1})^2}, \tag{13}
\]

and

\[
\beta = \text{sign}(x_j^{i+1} - x_j^{i-1}) \frac{\sqrt{(x_j^{i+1} - x_j^{i-1})^2 + (z_j^{i+1} - z_j^{i-1})^2 - (r_j^{i+1} - r_j^{i-1})^2}}{(x_j^{i+1} - x_j^{i-1})^2 + (z_j^{i+1} - z_j^{i-1})^2}. \tag{15}
\]
Figure 1: An updating scheme for HWT. Three points on the current wavefront \((A, B, \text{ and } C)\) are used to advance in the \(\tau\) direction.

Figure 2 shows a geometric interpretation of formulas (7) and (10). Formula (10) is clearly a line equation. Thus, the new point \(D\) in Figure 2 is defined as one of the two intersections of this line with the \(B\) sphere, defined by formula (7). It is easy to show geometrically that the newly created ray segment \(BD\) is orthogonal to the common tangent of spheres \(A\) and \(C\). Within the finite-difference approximation, the common tangent reflects local wavefront behavior.

Figure 2: A geometrical updating scheme for HWT in the physical domain. Three points on the current wavefront \((A, B, \text{ and } C)\) are used to compute the position of the \(D\) point. The bold lines represent equations (7) and (10). The tangent to circle \(B\) at point \(D\) is parallel to the common tangent of circles \(A\) and \(C\).

IMPLEMENTATION DETAILS

There are a few problems that have to be addressed for the successful implementation of the algorithm described in the preceding section. The most important are the boundary values, the existence of a double solution (7, 10), and the complications of finding the solution in the vicinity of the cusp points.
Boundary values

As mentioned in the preceding section, the application of formulas (11-12) requires the existence of known boundary values for both the first value of $\tau$ (next to the wave source) and the extreme values of the take-off angle $\gamma$. Therefore, we have to initialize the complete first wavefront as well as two boundary rays that represent all the extreme points of each consequent wavefront (that is, for the first and last considered take-off angle).

To initialize the points on the first wavefront, we consider that the velocity is constant around the source, and therefore this wavefront becomes a circle centered at the source. This is a reasonable assumption because we use a finite difference scheme with very small time steps, and the velocity models have limited local variation.

The values of the boundary rays are externally supplied. This apparent problem is very easy to solve by using a ray tracing program to compute the trajectories of these two boundary rays. We can shoot several “trial” rays and select the ones that are the smoothest and that penetrate the most into the model.

Figure 3: The double solution of the system of equations (7-10). D and E are the intersection points between the circle given by equation (7) and the line given by equation (10). Point O is the previous point on the ray going through B. The distance (OE) is smaller than the distance (OD) and, therefore, D is the next selected point. The middle ray is defined locally by the succession of points (-O-B-D-).

The double solution

The system (7-10) has two theoretical solutions (11-12), though there is only one that makes physical sense given a velocity map. Again, we used a geometrical argument to select the appropriate solution. We observed that even though a wavefront can make a sharp turn, the corresponding rays cannot (see the examples in the next section). We define a turn as “sharp” if it happens over a very small number of samples (say,
three). Consequently, we decided to impose the condition that the correct solution is the one represented by the point farthest away from the preceding one on the same ray (Figure 3).

**Cusp Points**

The final problem to be solved is represented by the cusp points, the case in which the three-point scheme doesn’t provide a satisfactory solution because it tends to decrease in an unnatural way the sharpness of the wavefronts. In this case, we reduce the three-point scheme to a two-point one by assuming that one of the exterior points (either A or C, Figure 4) is merged with the point in the middle (B).

Figure 4: Cusp points. A, B and C are the three points on the current wavefront. Point O is the previous point on the ray going through B. The angle CBA is smaller than the angle OBA, and therefore B is a cusp point. If the angle CBA is closer to 90 degrees than the angle OBA, then C is merged with B; otherwise, A is merged with B. The three-point scheme becomes a two-point scheme without any change in the program.

**EXAMPLES**

This section presents three examples in which we applied the method described in the last section. The first two applications are on simple Gaussian velocity anomalies in a medium of constant velocity. We used these models to check the validity, accuracy, and stability of the HWT method. The third example concerns the very complex Marmousi 2-D model, which is one of the most difficult benchmarks for ray tracing methods. Throughout the test, we have compared our results with those obtained with a ray tracing program for accuracy, speed, and stability.

**Gaussian velocity anomalies**

Our first two examples are Gaussian velocity anomalies (one positive and one negative) with a magnitude of 2.0 km/s in a constant velocity medium of 2.0 km/s for the positive anomaly, shown in Figure 5 and of 3.0 km/s for the negative anomaly in
Figure 6. The anomaly is centered at a depth of 1.0 km and has a half-width of 300 m. The source is placed on the surface directly above the anomaly (at x = 6.0 km).

Figure 5: A Gaussian positive velocity anomaly. The background velocity is 2.0 km/s, and the maximum anomaly at the center is +2.0 km/s. 

We have selected these velocity models to test the way our method applies to different patterns of velocity variation. In the case of the negative anomaly, the rays focus inward, while in the case of the positive anomaly the rays spread outward.

The distribution of rays as obtained with the PRT and HWT methods are presented in Figure 7 for the positive anomaly, and in Figure 8 for the negative.

One way to compare the two methods is to compute the distance between the points that correspond to the same ray, identified by the same take-off angle, at the same traveltimes. This is obviously not a perfect quantitative comparison, because once two rays, obtained with the two methods, become slightly divergent, they keep going in different directions, and thus the distance between corresponding points keeps growing (Figures 9 and 10). However, this effect is not necessarily a manifestation of decreasing precision. It can be easily seen that if such an angular mismatch doesn’t occur, the rays maintain practically the same path (see, for example, the rays shot in the (-20,-40) and (20,40) degree intervals, where the distance decreases in many cases to almost zero). Even in the case of divergent rays, the distance is kept to a reasonable level (less than 1%). Consequently, we do not interpret these differences as error.
Figure 6: A Gaussian negative velocity anomaly. The background velocity is 3.0 km/s, and the maximum anomaly at the center is -2.0 km/s.

Figure 7: The rays obtained in the case of the Gaussian positive velocity anomaly. We present the rays obtained with the PRT method (left) and with the HWT method (right). The source is located on the surface at x=6.0 km.
Figure 8: The rays obtained in the case of the Gaussian negative velocity anomaly. We present the rays obtained with the PRT method (left) and with the HWT method (right). The source is located on the surface at $x=6.0$ km.

Figure 9: The distance between the corresponding points on the rays obtained with the PRT method and with the HWT method. Distances are given in meters.

Figure 10: The distance between the corresponding points on the rays obtained with the PRT method and with the HWT method. Distances are given in meters.
The Marmousi model

In the third example, we have applied the same method to trace rays in the far more complex Marmousi 2-D Model. Figure 11 contains the true velocity (left) and a smoothed version using twice a tridiagonal $5 \times 5$ filter (right). In Figure 12 we present the rays obtained on the unsmoothed Marmousi Model with the PRT method (left) and with the HWT method (right). In Figure 13 we present the rays obtained on the smoothed Marmousi Model with the PRT method (left) and with the HWT method (right).

Figure 11: The Marmousi model. The true velocity appears on the left, the smoothed velocity on the right.

Figure 12: The rays obtained in the true velocity Marmousi model using the PRT method (left) and the HWT method (right).

As expected, the rays traced using the PRT method (Figure 12, left), which represents a more exact solution to the eikonal equation for the given velocity field, have a very rough distribution. Since this erratic result is of no use in practice,
regardless of its accuracy, the only way to get a proper result is to apply the ray tracing to a smoothed velocity model (Figure 13, left).

On the other hand, the result obtained with the HWT method looks a lot better, though some imperfections are still visible. For the case of the unsmoothed velocity medium, the rays have a much smoother pattern, which is less dependent on how rough the velocity model is (Figure 12, right). This feature is preserved in the case of the smoothed model (Figure 13, right) where the distributions of rays displayed by the two methods are much more similar, though some differences remain (see, for example the zone around x=6.5km, z=2.0km).

As with the Gaussian model, we present the distances between the points that correspond to the same ray, identified by the same take-off angle, at the same travel-times (Figure 14). This is another way to interpret what we saw in Figure 13, where most of the rays have a consistent behavior, displaying similar paths regardless of the method used, and therefore small distances, and a few have a different trajectory, resulting in big distances that increase with traveltime.

CONCLUSIONS

The results obtained so far have led us to the following conclusions:

1. Stability: The HWT method is a lot more stable in rough velocity media than the PRT method. The increased stability results from the fact that HWT derives the points on the new wavefronts from three points on the preceding wavefront, compared to only one in the usual PRT, which also means that a certain degree of smoothing is already embedded in the method. This feature
Figure 14: The distance between the corresponding points on the rays obtained with the PRT method and with the HWT method. Distances are given in meters.

allows us to use the HWT method in media of very sharp velocity variation and still obtain results that are reasonable from a geophysical point of view.

2. Coverage: Being more stable and giving smoother rays than the PRT method, enables the HWT method to provide a better coverage of the shadow zones. The idea is that since the wavefront is traced from one ray to the other, it is very easy to introduce in the code a condition to decrease the shooting angle as soon as the wavefront length exceeds a specified upper limit.

3. Speed: Both methods were tested on an SGI 200. The execution time for shooting 90 rays of 130 samples for each ray was 1.31 s for the PRT method and 0.22 s for the HWT method. Even though in the current implementation of HWT we do not compute the amplitudes of the waves, our method has still yielded a big improvement in speed for the 2-D case, which gives us hope of doing even better in the 3-D case.

In our future work, we will implement the 3-D Huygens wavefront tracing method. We expect to preserve its stability, while making it run even faster in comparison to other conventional 3-D ray tracing methods.

REFERENCES
van Trier, J., and W. W. Symes, 1991, Upwind finite-difference calculation of travel-
times: Geophysics, 56, 812–821.
Vidale, J. E., 1990, Finite-difference calculation of traveltimes in three dimensions: 
Geophysics, 55, 521–526.
Vinje, V., E. Iversen, and H. Gjoostdal, 1993, Traveltime and amplitude estimation 
using wavefront construction: Geophysics, 58, 1157–1166.
University.
A variational formulation
of the fast marching eikonal solver

Sergey Fomel

ABSTRACT
I exploit the theoretical link between the eikonal equation and Fermat’s principle to derive a variational interpretation of the recently developed method for fast traveltime computations. This method, known as fast marching, possesses remarkable computational properties. Based originally on the eikonal equation, it can be derived equally well from Fermat’s principle. The new variational formulation has two important applications: First, the method can be extended naturally for traveltime computation on unstructured (triangulated) grids. Second, it can be generalized to handle other Hamilton-type equations through their correspondence with variational principles.

INTRODUCTION
Traveltime computation is one of the most important tasks in seismic processing (Kirchhoff depth migration and related methods) and modeling. The traveltime field of a fixed source in a heterogeneous medium is governed by the eikonal equation, derived about 150 years ago by Sir William Rowan Hamilton. A direct numerical solution of the eikonal equation has become a popular method of computing travel-times on regular grids, commonly used in seismic imaging (Vidale, 1990; van Trier and Symes, 1991; Podvin and Lecomte, 1991). A recent contribution to this field is the fast marching level set method, developed by Sethian (1996a) in the general context of level set methods for propagating interfaces (Osher and Sethian, 1988; Sethian, 1996b). Sethian and Popovici (1997) report a successful application of this method in three-dimensional seismic computations. The fast marching method belongs to the family of upwind finite-difference schemes aimed at providing the viscosity solution (Lions, 1982), which corresponds to the first-arrival branch of the traveltime field. The remarkable stability of the method results from a specifically chosen order

*e-mail: sergey@sep.stanford.edu
of finite-difference evaluation. The order selection scheme resembles the *expanding wavefronts* method of Qin et al. (1992). The fast speed of the method is provided by the heap sorting algorithm, commonly used in Dijkstra’s shortest path computation methods (Cormen et al., 1990). A similar idea has been used previously in a slightly different context, in the *wavefront tracking* algorithm of Cao and Greenhalgh (1994).

In this paper, I address the question of evaluating the fast marching method’s applicability to more general situations. I describe a simple interpretation of the algorithm in terms of variational principles (namely, Fermat’s principle in the case of eikonal solvers.) Such an interpretation immediately yields a useful extension of the method for unstructured grids: triangulations in two dimensions and tetrahedron tessellations in three dimensions. It also provides a constructive way of applying similar algorithms to solving other eikonal-like equations: anisotropic eikonal (Dellinger, 1991), “focusing” eikonal (Biondi et al., 1997), kinematic offset continuation (Fomel, 1995), and kinematic velocity continuation (Fomel, 1996). Additionally, the variational formulation can give us hints about higher-order enhancements to the original first-order scheme.

**A BRIEF DESCRIPTION OF THE FAST MARCHING METHOD**

For a detailed description of level set methods, the reader is referred to Sethian’s recently published book (Sethian, 1996b). More details on the fast marching method appear in articles by Sethian (1996a) and Sethian and Popovici (1997). This section serves as a brief introduction to the main bulk of the algorithm.

The key feature of the algorithm is a carefully selected order of traveltime evaluation. At each step of the algorithm, every grid point is marked as either *Alive* (already computed), *NarrowBand* (at the wavefront, pending evaluation), or *FarAway* (not touched yet). Initially, the source points are marked as *Alive*, and the traveltime at these points is set to zero. A continuous band of points around the source are marked as *NarrowBand*, and their traveltime values are computed analytically. All other points in the grid are marked as *FarAway* and have an “infinitely large” traveltime value.

An elementary step of the algorithm consists of the following moves:

1. Among all the *NarrowBand* points, extract the point with the minimum traveltime.
2. Mark this point as *Alive*.
3. Check all the immediate neighbors of the minimum point and update them if necessary.
4. Repeat.
An update procedure is based on an upwind first-order approximation to the eikonal equation. In simple terms, the procedure starts with selecting one or more (up to three) neighboring points around the updated point. The traveltime values at the selected neighboring points need to be smaller than the current value. After the selection, one solves the quadratic equation

\[ \sum_j \left( \frac{t_i - t_j}{\Delta x_{ij}} \right)^2 = s_i^2 \]  

for \( t_i \). Here \( t_i \) is the updated value, \( t_j \) are traveltime values at the neighboring points, \( s_i \) is the slowness at the point \( i \), and \( \Delta x_{ij} \) is the grid size in the \( ij \) direction. As the result of the updating, either a FarAway point is marked as NarrowBand or a NarrowBand point gets assigned a new value.

Except for the updating scheme (1), the fast marching algorithm bears a very close resemblance to the famous shortest path algorithm of Dijkstra (1959). It is important to point out that unlike Moser’s method, which uses Dijkstra’s algorithm directly (Moser, 1991), the fast marching approach does not construct the ray paths from predefined pieces, but dynamically updates traveltimes according to the first-order difference operator (1). As a result, the computational error of this method goes to zero with the decrease in the grid size in a linear fashion. The proof of validity of the method (omitted here) is also analogous to that of Dijkstra’s algorithm (Sethian, 1996a,b). As in most of the shortest-path implementations, the computational cost of extracting the minimum point at each step of the algorithm is greatly reduced [from \( O(N) \) to \( O(\log N) \) operations] by maintaining a priority-queue structure (heap) for the NarrowBand points (Cormen et al., 1990).

Figure 1 shows an example application of the fast marching eikonal solver on the three-dimensional SEG/EAGE salt model. The computation is stable despite the large velocity contrasts in the model. The current implementation takes about 10 seconds for computing a 100x100x100 grid on one node of SGI Origin 200. Alkhalifah and Fomel (1997) discuss the differences between Cartesian and polar coordinate implementations.

The difference equation (1) is a finite-difference approximation to the continuous eikonal equation

\[ \left( \frac{\partial t}{\partial x} \right)^2 + \left( \frac{\partial t}{\partial y} \right)^2 + \left( \frac{\partial t}{\partial z} \right)^2 = s^2(x, y, z) , \]  

where \( x, y, \) and \( z \) represent the spatial Cartesian coordinates. In the next two sections, I show how the updating procedure can be derived without referring to the eikonal equation, but with the direct use of Fermat’s principle.
THEORETIC GROUNDS OF VARIATIONAL PRINCIPLES

This section serves as a brief reminder of the well-known theoretical connection between Fermat’s principle and the eikonal equation. The reader, familiar with this theory, can skip safely to the next section.

Both Fermat’s principle and the eikonal equation can serve as the foundation of traveltime calculations. In fact, either one can be rigorously derived from the other. A simplified derivation of this fact is illustrated in Figure 2. Following the notation of this figure, let us consider a point $M$ in the immediate neighborhood of a wavefront $t(N) = t_N$. Assuming that the source is on the other side of the wavefront, we can express the traveltime at the point $M$ as the sum

$$t_M = t_N + l(N, M) s_M ,$$  \hspace{1cm} (3)
where $N$ is a point on the front, $l(N, M)$ is the length of the ray segment between $N$ and $M$, and $s_M$ is the local slowness. As follows directly from equation (3),

$$|\nabla t| \cos \theta = \frac{\partial t}{\partial l} = \lim_{M \to N} \frac{t_M - t_N}{l(N, M)} = s_N .$$

(4)

Here $\theta$ denotes the angle between the traveltime gradient (normal to the wavefront surface) and the line from $N$ to $M$, and $\frac{\partial t}{\partial N}$ is the directional traveltime derivative along that line.

If we accept the local Fermat’s principle, which says that the ray from the source to $M$ corresponds to the minimum-arrival time, then, as we can see geometrically from Figure 2, the angle $\theta$ in formula (4) should be set to zero to achieve the minimum. This conclusion leads directly to the eikonal equation (19). On the other hand, if we start from the eikonal equation, then it also follows that $\theta = 0$, which corresponds to the minimum traveltime and constitutes the local Fermat’s principle. The idea of that simplified proof is taken from [Lanczos 1966], though it has obviously appeared in many other publications. The situations in which the wavefront surface has a discontinuous normal (given raise to multiple-arrival traveltimes) require a more elaborate argument, but the above proof does work for first-arrival traveltimes and the corresponding viscosity solutions of the eikonal equation [Lions 1982].

The connection between variational principles and first-order partial-differential equations has a very general meaning, explained by the classic Hamilton-Jacobi theory. One generalization of the eikonal equation is

$$\sum_{i,j} a_{ij}(x) \frac{\partial \tau}{\partial x_i} \frac{\partial \tau}{\partial x_j} = 1 ,$$

(5)

where $x = \{x_1, x_2, \ldots\}$ represents the vector of space coordinates, and the coefficients $a_{ij}$ form a positive-definite matrix $A$. Equation (5) defines the characteristic surfaces $t = \tau(x)$ for a linear hyperbolic second-order differential equation of the form

$$\sum_{i,j} a_{ij}(x) \frac{\partial^2 u}{\partial x_i \partial x_j} + F(x, u, \frac{\partial u}{\partial x_i}) = \frac{\partial^2 u}{\partial t^2} ,$$

(6)

where $F$ is an arbitrary function.

A known theorem [Smirnov 1964] states that the propagation rays [characteristics of equation (5)] and, correspondingly, bi-characteristics of equation (6) are geodesic (extreme-length) curves in the Riemannian metric

$$d\tau = \sqrt{\sum_{i,j} b_{ij}(x) dx_i dx_j} ,$$

(7)

where $b_{ij}$ are the components of the matrix $B = A^{-1}$. This means that a ray path between two points $x_1$ and $x_2$ has to correspond to the extreme value of the curvilinear
integral
\[ \int_{x_1}^{x_2} \sqrt{\sum_{i,j} b_{ij}(x) dx_i dx_j} \, dx \, dx_j. \]

For the isotropic eikonal equation (19), \( a_{ij} = \delta_{ij}/s^2(x) \), and metric (7) reduces to the familiar traveltime measure

\[ d\tau = s(x) d\sigma, \quad (8) \]

where \( d\sigma = \sqrt{\sum_i dx_i^2} \) is the usual Euclidean distance metric. In this case, the geodesic curves are exactly Fermat’s extreme-time rays.

From equation (7), we see that Fermat’s principle in the general variational formulation applies to a much wider class of situations if we interpret it with the help of non-Euclidean geometries.

**VARIATIONAL PRINCIPLES ON A GRID**

In this section, I derive a discrete traveltime computation procedure, based solely on Fermat’s principle, and show that on a Cartesian rectangular grid it is precisely equivalent to the update formula (1) of the first-order eikonal solver.

Figure 3: A geometrical scheme for the traveltime updating procedure in two dimensions.

For simplicity, let us focus on the two-dimensional case. Consider a line segment with the end points \( A \) and \( B \), as shown in Figure 3. Let \( t_A \) and \( t_B \) denote the traveltimes from a fixed distant source to points \( A \) and \( B \), respectively. Define a parameter \( \xi \) such that \( \xi = 0 \) at \( A \), \( \xi = 1 \) at \( B \), and \( \xi \) changes continuously on the line segment between \( A \) and \( B \). Then for each point of the segment, we can approximate the traveltime by the linear interpolation formula

\[ t(\xi) = (1 - \xi)t_A + \xi t_B. \quad (9) \]

\(^1\)A very similar analysis applies in three dimensions, but requires a slightly more tedious algebra. It is left as an exercise for the reader.
Now let us consider an arbitrary point $C$ in the vicinity of $AB$. If we know that the ray from the source to $C$ passes through some point $\xi$ of the segment $AB$, then the total traveltime at $C$ is approximately

$$
t_C = t(\xi) + s_C \sqrt{|AB|^2 (\xi - \xi_0)^2 + \rho_0^2},
$$

where $s_C$ is the local slowness, $\xi_0$ corresponds to the projection of $C$ to the line $AB$ (normalized by the length $|AB|$), and $\rho_0$ is the length of the normal from $C$ to $\xi_0$.

Fermat’s principle states that the actual ray to $C$ corresponds to a local minimum of the traveltime with respect to raypath perturbations. According to our parameterization, it is sufficient to find a local extreme of $t_C$ with respect to the parameter $\xi$. Equating the $\xi$ derivative to zero, we arrive at the equation

$$
t_B - t_A + \frac{s_C |AB|^2 (\xi - \xi_0)}{\sqrt{|AB|^2 (\xi - \xi_0)^2 + \rho_0^2}} = 0,
$$

which has (as a quadratic equation) the explicit solution for $\xi$:

$$
\xi = \xi_0 \pm \frac{\rho_0 (t_A - t_B)}{|AB| \sqrt{s_C^2 |AB|^2 - (t_A - t_B)^2}}.
$$

Finally, substituting the value of $\xi$ from (12) into equation (10) and selecting the appropriate branch of the square root, we obtain the formula

$$
c t_C = \rho_0 \sqrt{s_C^2 c^2 - (t_A - t_B)^2} + c t_A (1 - \xi_0) + c t_B \xi_0 =
 \rho_0 \sqrt{s_C^2 c^2 - (t_A - t_B)^2} + a t_A \cos \beta + b t_B \cos \alpha,
$$

where $c = |AB|$, $a = |BC|$, $b = |AC|$, angle $\alpha$ corresponds to $\hat{BAC}$, and angle $\beta$ corresponds to $\hat{ABC}$ in the triangle $ABC$ (Figure 3).

Figure 4: NR

A geometrical scheme for traveltime updating on a rectangular grid.

To see the connection of formula (13) with the eikonal difference equation (11), we need to consider the case of a rectangular computation cell with the edge $AB$ being
a diagonal segment, as illustrated in Figure 4. In this case, \( \cos \alpha = \frac{a}{c} \), \( \cos \beta = \frac{b}{c} \), \( \rho_0 = \frac{ab}{c} \), and formula (13) reduces to

\[
t_C = \frac{ab}{a^2+b^2} \sqrt{s_C^2(a^2+b^2) - (t_A-t_B)^2 + a^2 t_A + b^2 t_B}.
\]  

(14)

We can notice that (14) is precisely equivalent to the solution of the quadratic equation (13), which in our new notation takes the form

\[
\left( \frac{t_C-t_A}{b} \right)^2 + \left( \frac{t_C-t_B}{a} \right)^2 = s_C^2.
\]  

(15)

What have we accomplished by this analysis? First, we have derived a local traveltime computation formula for an arbitrary grid. The derivation is based solely on Fermat’s principle and a local linear interpolation, which provides the first-order accuracy. Combined with the fast marching evaluation order, which is also based on Fermat’s principle, this procedure defines a complete algorithm of first-arrival traveltime calculation. On a rectangular grid, this algorithm is exactly equivalent to the fast marching method of Sethian (1996a) and Sethian and Popovici (1997). Second, the derivation provides a general principle, which can be applied to derive analogous algorithms for other eikonal-type (Hamilton-Jacobi) equations and their corresponding variational principles.

**SOLVING THE EIKONAL EQUATION ON A TRIANGULATED GRID**

Unstructured (triangulated) grids have computational advantages over rectangular ones in three common situations:

- When the number of grid points can be substantially reduced by putting them on an irregular grid. This situation corresponds to irregular distribution of details in the propagation medium.

- When the computational domain has irregular boundaries. One possible kind of boundary corresponds to geological interfaces and seismic reflector surfaces (Wiggins et al., 1993). Another type of irregular boundary, in application to traveltime computations, is that of seismic rays. The method of bounding the numerical eikonal solution by ray envelopes has been introduced recently by Abgrall and Benamou (1996).

- When the grid itself needs to be dynamically updated to maintain a certain level of accuracy in the computation.

With its computational speed and unconditional stability, the fast marching method provides considerable savings in comparison with alternative, more accurate methods,
such as semi-analytical ray tracing (Guiziou et al., 1991; Stankovic and Albertin, 1995) or the general Hamilton-Jacobi solver of Abgrall (1996).

Computational aspects of triangular grid generation are outlined in Appendix A. A three-dimensional application would follow the same algorithmic patterns.

CONCLUSIONS

Variational principles have played an exceptionally important role in the foundations of mathematical physics. Their potential in numerical algorithms should not be underestimated.

In this paper, I interpret the fast marching eikonal solver with the help of Fermat’s principle. Two important generalizations follow immediately from that interpretation. First, it allows us to obtain a fast method of first-arrival traveltime computation on triangulated grids. Furthermore, we can obtain a general principle, which extends the fast marching algorithm to other Hamilton-type equations and their variational principles. More research is required to confirm these promises.

In addition, future research should focus on 3-D implementations and on increasing the approximation order of the method.

ACKNOWLEDGMENTS

I thank Mihai Popovici and Biondo Biondi for drawing my attention to the fast marching level set method. Discussions with Bill Symes, Dan Kosloff, and Tariq Alkhalifah were crucial for developing a general understanding of the method. Jamie Sethian kindly responded to more specific questions. The conforming triangulation program was developed at the suggestion of Leonidas Guibas as a research project for his Geometrical Algorithms class at Stanford.

REFERENCES

tomography, in SEP-95: Stanford Exploration Project, 61–76.
Cao, S., and S. A. Greenhalgh, 1994, Finite-difference solution of the eikonal equation
(Errata in GEO-64-03-992).
Cormen, T. H., C. E. Leiserson, and R. L. Rivest, 1990, Introduction to algorithms:
McGraw-Hill.
Nauk, 7, 793–800.
Fomel, S., 1995, Amplitude preserving offset continuation in theory Part 1: The offset
——–, 1996, Migration and velocity analysis by velocity continuation, in SEP-92:
Stanford Exploration Project, 159–188.
174.
Guibas, L., and J. Stolfi, 1985, Primitives for the manipulation of general subdivisions
Guibas, L. J., D. Knuth, and M. Sharir, 1992, Randomized incremental construction
of Delaunay and Voronoi diagrams: Algorithmica, 7, 381–413.
Hale, D., and J. K. Cohen, 1991, Triangulated models of the Earth’s subsurface:
Lanczos, C., 1966, The variational principles of mechanics: University of Toronto
Press.
Lee, D. T., and A. K. Lin, 1986, Generalized Delaunay triangulation for planar graphs:
Osher, S., and J. A. Sethian, 1988, Fronts propagating with curvature-dependent
speed: Algorithms based on Hamilton-Jacobi formulation: Jour. of Comp. Phys.,
79, 12–49.
Podvin, P., and I. Lecomte, 1991, Finite difference computation of traveltimes in very


——–, 1996b, Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science: Cambridge University Press.


Smirnov, V. I., 1964, A course on higher mathematics: Pergamon Press.


APPENDIX A

INCREMENTAL DELAUNAY TRIANGULATION AND RELATED PROBLEMS

Delaunay triangulation ([Delaunay, 1934; Sibson, 1978; Guibas and Stolfi, 1985]) is a fundamental geometric construction, which has numerous applications in different computational problems. For a given set of nodes (points on the plane), Delaunay triangulation constructs a triangle tessellation of the plane with the initial nodes as vertices. Among all possible triangulations, the Delaunay triangulation possesses optimal properties, which make it very attractive for practical applications, such as computational mesh generation. One of the most well-known properties is maximizing the minimum triangulation angle. In three dimensions, Delaunay triangulation generalizes naturally to a tetrahedron tessellation.

Several optimal-time algorithms of Delaunay triangulation (and its counterpart–Voronoi diagram) have been proposed in the literature. The divide-and-conquer algorithm ([Shamos and Hoey, 1975; Guibas and Stolfi, 1985]) and the sweep-line algorithm ([Fortune, 1987]) both achieve the optimal $O(N \log N)$ worst-case time complexity. Alternatively, a family of incremental algorithms has been used in practice because of their simplicity and robustness. Though the incremental algorithm can take $O(N^2)$ time in the worst case, the expectation time can still be $O(N \log N)$, provided that the nodes are inserted in a random order ([Guibas et al., 1992]).

The incremental algorithm consists of two main parts:

1. Locate a triangle (or an edge), containing the inserted point.

2. Insert the point into the current triangulation, making the necessary adjustments.

The Delaunay criterion can be reduced in the second step to a simple $InCircle$ test ([Guibas and Stolfi, 1985]): if a circumcircle of a triangle contains another triangulation vertex in its circumcenter, then the edge between those two triangles should be “flipped” so that two new triangles are produced. The testing is done in a recursive fashion consistent with the incremental nature of the algorithm. When a new node is inserted inside a triangle, three new triangles are created, and three edges need to be tested. When the node falls on an edge, four triangles are created, and four edges are tested. In the case of test failure, a pair of triangles is replaced by the flip operation with another pair, producing two more edges to test. Under the randomization assumption, the expected total time of point insertion is $O(N)$. Randomization can be considered as an external part of the algorithm, provided by preprocessing. ([Guibas et al., 1992]) reduce the point location step to an efficient $O(N \log N)$ procedure by maintaining a hierarchical tree structure: all triangles, occurring in the
incremental triangulation process, are kept in memory, associated with their “parents.” One or two point location tests (CCW tests) are sufficient to move to a lower level of the tree. The search terminates with a current Delaunay triangle.

To test the algorithmic performance of the incremental construction, I have profiled the execution time of my incremental triangulation program with the Unix pixie utility. The profiling result, shown in Figures A-1 and A-2, complies remarkably with the theory: $O(N \log N)$ operations for the point location step, and $O(N)$ operations for the point insertion step. The experimental constant for the insertion step time is about 8.6. The experimental constant for the point location step is 4. The CPU time, depicted in Figure A-3, also shows the expected $O(N \log N)$ behavior.

Figure A-1: The number of point insertion operations (InCircle test) plotted against the number of points.

Figure A-2: Number of point location operations (CCW test) plotted against the number of points.

Figure A-3: CPU time (in seconds per point) plotted against the number of points.

A straightforward implementation of Delaunay triangulation would provide an optimal triangulation for any given set of nodes. However, the quality of the result
for unfortunate geometrical distributions of the nodes can be unsatisfactory. In the rest of this appendix, I describe three problems, aimed at improving the triangulation quality: conforming triangulation, triangulation of height fields, and mesh refinement. Each of these problems can be solved with a variation of the incremental algorithm.

Conforming Triangulation

In the practice of mesh generation, the input nodes are often supplemented by boundary edges: geologic interfaces, seismic rays, and so on. It is often desirable to preserve the edges so that they appear as edges of the triangulation (Albertin and Wigging, 1994). One possible approach is constrained triangulation, which preserves the edges, but only approximately satisfies the Delaunay criterion (Lee and Lin, 1986; Chew, 1989). An alternative, less investigated, approach is conforming triangulation, which preserves the “Delaunayhood” of the triangulation by adding additional nodes (Hansen and Levin, 1992) (Figure A-4). Conforming Delaunay triangulations are difficult to analyze because of the variable number of additional nodes. This problem was attacked by Edelsbrunner and Tan (1993), who suggested an algorithm with a defined upper bound on added points. Unfortunately, Edelsbrunner’s algorithm is slow in practice because the number of added points is largely overestimated. I chose to implement a modification of the simple incremental algorithm of Hansen and Levin. Although Hansen’s algorithm has only a heuristic justification and sets no upper bound on the number of inserted nodes, its simplicity is attractive for practical implementations, where it can be easily linked with the incremental algorithm of Delaunay triangulation.

The incremental solution to the problem of conforming triangulation can be described by the following scheme:

- First, the boundary nodes are triangulated.
- Boundary edges are inserted incrementally.
- If a boundary edge is not present in the triangulations, it is split in half, and the middle node is inserted into the triangulation. This operation is repeated for the two parts of the original boundary edge and continues recursively until all the edge parts conform.
- If at some point during the incremental process, a boundary edge violates the Delaunay criterion (the $InCircle$ test), it is split to assure the conformity.

To insert an edge $AB$ into the current triangulation, I use the following recursive algorithm:

Function $InsertEdge$ ($AB$)
1. Define $C$ to be the midpoint of $AB$.

2. Using the triangle tree structure, locate triangle $T = DEF$ that contains $C$ in the current triangulation.

3. **If** $AB$ is an edge of $T$ **then return**.

4. **If** $A$ (or $B$) is a vertex of $T$ (for example, $A = D$) **then** define $C$ as an intersection of $AB$ and $EF$.

5. **Else** define $C$ as an intersection of $AB$ and an arbitrary edge of $T$ (if such an intersection exists).

6. Insert $C$ into the triangulation.

7. **InsertEdge** $(CA)$.

8. **InsertEdge** $(CB)$.

The intersection point of edges $AB$ and $EF$ is given by the formula

$$ C = A + \lambda (B - A) , $$

(A-1)

where

$$ \lambda = \frac{(F_y - E_y)(E_x - A_x) - (F_x - E_x)(E_y - A_y)}{\det \begin{vmatrix} B_x - A_x & B_y - A_y \\ F_x - E_x & F_y - E_y \end{vmatrix}} . $$

(A-2)

The value of $\lambda$ should range between 0 and 1.

If, at some stage of the incremental construction, a boundary edge $AB$ fails the Delaunay $InCircle$ test for the circle $CABD$, then I simply split it into two edges by adding the point of intersection into the triangulation. The rest of the process is very much like the process of edge validation in the original incremental algorithm.
Triangulation of Height Fields

Often, a velocity field (or other object that we want to triangulate) is defined on a regular Cartesian grid. One way to perform a triangulation in this case is to select a smaller subset of the initial grid points, using them as the input to a triangulation program. We need to select the points in a way that preserves the main features of the original image, while removing some unnecessary redundancy in the regular grid description.

![Figure A-5: Illustration of Garland’s algorithm for triangulation of height fields. The left plot shows the input image of a sphere, containing 100 by 100 pixels. The middle plot shows 500 pixels, selected by the algorithm and triangulated. The right plot is the result of local plane interpolation of the triangulated surface.](fmeiko/tri/sphere)

Garland and Heckbert (1996) surveyed different approaches to this problem and proposed a fast version of the incremental greedy insertion algorithm. Their algorithm adds points incrementally, selecting at each step the point with the maximum interpolation error with respect to the current triangulation. Though a straightforward implementation of this idea would lead to an unacceptably slow algorithm, Garland and Heckbert have discovered several sources for speeding it up. First, we can take advantage of the fact that only a small area of the current triangulation gets updated at each step. Therefore, it is sufficient to recompute the interpolation error only inside this area. Second, the maximum extraction procedure can be implemented very efficiently with a priority queue data structure.

Figure A-5 illustrates this algorithm with a simple example. The original image (the left plot) contained 10,000 points, laid out on a regular rectangular grid. The algorithm selects a smaller number of points and immediately triangulates them (the middle plot). The image can be reconstructed by local plane interpolation (the right plot.) The reconstruction quality can be further improved by increasing the number of triangles. Figure A-6 shows the same image as rendered by the OpenGL graphics library (Woo et al., 1997).

Figure 3 shows an application of the height triangulation algorithm to the famous Marmousi model. The left plot shows a smoothed and windowed version of the
Figure A-6: An image from the previous example, as rendered by the OpenGL library. The shades on polygonal (triangulated) sides are exaggerated by a simulation of the direct light source.

Figure A-7: Applying the height triangulation algorithm to the Marmousi model. The left plot shows a smoothed and windowed version of the Marmousi model. The middle plot is a result of 10,000-point triangulation, followed by linear interpolation. The right plot shows the difference between the two images.
Marmousi, plotted on a 501 by 376 computational grid. In the middle plot, 10,000 points from the original 188,376 were selected for triangulation and interpolated back to the rectangular grid. The right plot demonstrates the small difference between the two images.

Mesh Refinement

One the main properties of Delaunay triangulation is that, for a given set of nodes, it provides the maximum smallest angle among all possible triangulations. It is this property that supports the wide usage of Delaunay algorithms in the mesh generation problems. However, it doesn’t guarantee that the smallest angle will always be small. In fact, for some point distributions, it is impossible to avoid skinny small-angle triangles. The remedy is to add additional nodes to the triangulation so that the quality of the triangles is globally improved. This problem has become known as mesh refinement (Ruppert, 1995).

Figure A-8: An illustration of Rivara’s refinement algorithm. The left plot shows an input to the algorithm: a valid Delaunay triangulation with “skinny” triangles. Two other plots show successive applications of the refinement algorithm. One of the recently proposed mesh refinement algorithms is Rivara’s backward longest-side refinement technique (Rivara, 1996). The main idea of the algorithm is to trace the LSPP (longest-side propagation path) for each refined triangle. The LSPP is an ordered list of triangles, connected by a common edge, such that the longest triangle edge is strictly increasing. After tracing the LSPP, we bisect the longest edge and insert its midpoint into the triangulation. Rivara’s algorithm is remarkably efficient and easy to implement. In comparison with alternative methods, it has the additional advantage of being applicable in three dimensions.

Figure A-9 demonstrates an application of different triangulation techniques to a simple layered model, borrowed from the Seismic Unix demos (where it is attributed to V. Červený.) Another model from Hale and Cohen (1991) is used in Figure A-10.
Figure A-9: A comparison of different triangulation techniques on a simple layered model. The top left plot shows the original model; the top right plot, the result of nonconforming triangulation; the two bottom plots, conforming triangulation and an additional mesh refinement.
Figure A-10: A comparison of different triangulation techniques on a simple salt-type model. The top left plot shows the original model; the top right plot, the result of nonconforming triangulation; the two bottom plots, conforming triangulation and an additional two-step mesh refinement.
Implementation Details

Edge operations form the basis of the incremental algorithm. Therefore, it is convenient to describe triangulation with edge-oriented data structures (Guibas and Stolfi, 1985). I have followed the idea of Hansen and Levin (1992) of associating with each edge two pointers to the end points and two pointers to the adjacent triangles. The triangle structure is defined by three pointers to the edges of a triangle. Additionally, as required by the point location algorithm, each triangle has a pointer to its “children.” This pointer is NULL when the triangle belongs to the current Delaunay triangulation.

Many researchers have pointed out that the geometric primitives used in triangulation must be robust with respect to round-off errors of finite-precision calculation. To assure the robustness of the code, I used the adaptive-precision predicates of Shewchuk (1996), available as a separate package from the netlib public-domain archive.
A second-order fast marching eikonal solver

James Rickett and Sergey Fomel

INTRODUCTION

The fast marching method (Sethian, 1996) is widely used for solving the eikonal equation in Cartesian coordinates. The method’s principal advantages are: stability, computational efficiency, and algorithmic simplicity. Within geophysics, fast marching traveltime calculations (Popovici and Sethian, 1997) may be used for 3-D depth migration or velocity analysis.

Unfortunately, first-order implementations lead to inaccuracies in computed traveltimes, which may lead to poor image focusing for migration applications. In addition, first-order traveltimes are not accurate enough for reliable amplitude calculations. This has lead to the development of the fast marching method on non-Cartesian (Alkhalifah and Fomel, 1997; Sun and Fomel, 1998), and even unstructured (Fomel, 1997) grids. These non-Cartesian formulations reduce inaccuracies, while retaining the fast marching method’s characteristic stability and efficiency. Unfortunately, the cost is the loss of algorithmic simplicity.

We implement a second-order fast marching eikonal solver, which reduces inaccuracies while retaining stability, efficiency and simplicity.

FAST MARCHING AND THE EIKONAL EQUATION

Under a high frequency approximation, propagating wavefronts may be described by the eikonal equation,

\[
\left( \frac{\partial t}{\partial x} \right)^2 + \left( \frac{\partial t}{\partial y} \right)^2 + \left( \frac{\partial t}{\partial z} \right)^2 = s^2(x, y, z),
\]

where \( t \) is the travelt ime, \( s \) is the slowness, and \( x, y \) and \( z \) represent the spatial Cartesian coordinates.

The fast marching method solves equation (19) by directly mimicking the advancing wavefront. Every point on the computational grid is classified into three groups: points behind the wavefront, whose traveltimes are known and fixed; points on the wavefront, whose traveltimes have been calculated, but are not yet fixed; and points ahead of the wavefront. The algorithm then proceeds as follows:

1. Choose the point on the wavefront with the smallest travelt ime.
2. Fix this traveltime.

3. Advance the wavefront, so that this point is behind it, and adjacent points are either on the wavefront or behind it.

4. Update traveltimes for adjacent points on the wavefront by solving equation (19) numerically.

5. Repeat until every point is behind the wavefront.

The update procedure (step 4.) requires the solution of the following quadratic equation for \( t \),

\[
\begin{align*}
\max(D_{ijkt}^{-}t, 0)^2 + \min(D_{ijkt}^{+}t, 0)^2 & + \\
\max(D_{ijyt}^{-}t, 0)^2 + \min(D_{ijkt}^{+}y, 0)^2 & + \\
\max(D_{ijzt}^{-}t, 0)^2 + \min(D_{ijkt}^{+}z, 0)^2 & = s_{ijk}
\end{align*}
\]

(2)

where \( D_{ijkt}^{-} \) is a backward \( x \) difference operator at grid point, \( ijk \), \( D_{ijkt}^{+} \) is a forward \( x \) operator, and finite-difference operators in \( y \) and \( z \) are defined similarly. The roots of the quadratic equation, \( at^2 + bt + c = 0 \), can be calculated explicitly as

\[
t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\]

(3)

Solving equation (2) amounts to accumulating coefficients \( a, b \) and \( c \) from its non-zero terms, and evaluating \( t \) with equation (3).

If we choose a two-point finite-difference operator, such as

\[
D_{ijkt}^{-} = \frac{t_{ijk} - t_{(i-1)jk}}{\Delta x}
\]

(4)

then

\[
(D_{ijkt}^{-}t)^2 = \alpha t_{ijk}^2 + \beta t_{ijk} + \gamma
\]

(5)

where \( \alpha = \frac{1}{\Delta x^2}, \beta = -2t_{(i-1)jk}\alpha \) and \( \gamma = t_{(i-1)jk}^2\alpha \). Coefficients \( a, b \) and \( c \) can now be calculated from \( a = \Sigma_l \alpha_l, \beta = \Sigma_l \beta_l, \) and \( c = \Sigma_l \beta_l^2 - s^2 \), where the summation index, \( l \), refers to the six terms in equation (2) subject to the various min/max conditions.

This two-point stencil, however, is only accurate to first-order. If instead we choose a suitable three-point finite-difference stencil, we may expect the method to have second-order accuracy. For example, the second-order upwind stencil,

\[
D_{ijkt}^{-} = \frac{3t_{ijk} - 4t_{(i-1)jk} + t_{(i-2)jk}}{2\Delta x}
\]

(6)

gives

\[
(D_{ijkt}^{-}t)^2 = \alpha' t_{ijk}^2 + \beta' t_{ijk} + \gamma'
\]

(7)

where this time \( \alpha' = \frac{9}{4\Delta x^2}, \beta' = -\frac{3(4t_{(i-1)jk} - t_{(i-2)jk})}{2\Delta x^2} = -2\alpha' t_{(i-1)jk}, \gamma' = \frac{(4t_{(i-1)jk} - t_{(i-2)jk})^2}{4\Delta x^2} = \alpha' t_{(i-1)jk}^2. \)
and \[ t'_{(i-1)jk} = \frac{1}{3} (4t_{(i-1)jk} - t_{(i-2)jk}). \]

Coefficients, \( a, b \) and \( c \) can be accumulated from \( \alpha', \beta' \) and \( \gamma' \) as before, and if the traveltime, \( t_{(i-2)jk} \) is not available, first-order values may be substituted.

**ACCURACY**

Figure 1 shows traveltime contour maps computed with the first and second-order fast marching methods on a sparse (20 \( \times \) 20) grid. The large errors for waves propagating at 45\( ^\circ \) to the grid are visibly reduced by the second-order formulation.

![Traveltime contours](fmsec/cvel/circles)

Figure 1: Traveltime contours in a constant velocity medium. The solid line shows the exact result. The dashed line shows the first-order (left panel) and second-order (right panel) fast marching results, calculated on a 20 \( \times \) 20 grid.

Figure 2 shows the average error as a function of grid spacing for the first and second-order solvers. Not only is the second-order formulation more accurate at large grid spacing, but its accuracy increases more rapidly as grid spacing decreases. Theory predicts the log – log plots of average error against grid spacing to be a linear function with gradient of one for first-order methods, and two for second order methods. In practice, the fast marching results come very close to these criteria up to the limits of machine precision. Figure 2 demonstrates the superiority of the second-order fast marching formulation.

It is worth noting, at this point, that special treatment is required at the source location, since the singularity in wavefront curvature will cause numerical errors to propagate into the traveltime solution. We surround the source with a constant velocity box, within which we calculate traveltimes by ray-tracing. Errors are inversely
proportional to the radius of this box. Therefore, if the radius of the box decrease with grid spacing, errors will increase linearly, reducing the accuracy of the method to first-order. For full second-order accuracy, the box size should be independent of grid spacing.

Figure 2: Average error against grid spacing for a constant velocity model. The solid line corresponds to the first-order eikonal solver, and the dashed line corresponds to the second-order solver. The left panel has linear axes, whereas the right panel is a log–log plot.

**COMPUTATIONAL COST**

The leading term in the computational cost of the fast marching algorithm comes from the first step: choosing the point on the wavefront with the smallest traveltime. Consequently, the cost should not depend strongly on the order of the finite-difference stencil, but rather the sort algorithm used. Heap sorting has a cost of $O(\log N)$, and so in principle, with this algorithm, the fast marching method has a cost of $O(N \log N)$.

The left panel of Figure 4 shows a plot of CPU time against $N$ for the same models as Figure 2. The time shown is elapsed (wall clock) time on a 300 MHz Pentium II. For the largest model computed here, the second-order code takes 11% longer to run than the first-order code, and this percentage decreases as $N$ increases.

Because $\log N$ grows slowly compared to $N$, the plot of CPU time against $N$ is dominated by the linear term. The right panel in Figure 4 addresses this issue by showing CPU time divided by $N$ versus $N$. On this graph, the $\log N$ behaviour is clearly visible.
Figure 3: Traveltime contours calculated through the Marmousi velocity model sampled at 4 m. Solid line shows first-order results, and dashed line shows second-order results.

Figure 4: Elapsed CPU time vs. the number of grid points, $N$, for first-order (solid line) and second order (dashed line) eikonal solvers. Left panel shows CPU time vs $N$. Right panel shows CPU time$/N$ vs $N$. 
CONCLUSIONS

We have shown that a second-order implementation of the fast marching eikonal solver produces traveltimes with a much higher accuracy than the first-order implementation. What is more, the additional accuracy is achieved at only a marginal increase in cost.

This second-order implementation should become the standard method for computing first-arrival traveltimes within SEP.

REFERENCES


The time and space formulation of azimuth moveout

Sergey Fomel and Biondo L. Biondi

ABSTRACT

Azimuth moveout (AMO) transforms 3-D prestack seismic data from one common azimuth and offset to different azimuths and offsets. AMO in the time-space domain is represented by a three-dimensional integral operator. The operator components are the summation path, the weighting function, and the aperture. To determine the summation path and the weighting function, we derive the AMO operator by cascading dip moveout (DMO) and inverse DMO for different azimuths in the time-space domain. To evaluate the aperture, we apply a geometric approach, defining AMO as the result of cascading prestack migration (inversion) and modeling. The aperture limitations provide a consistent description of AMO for small azimuth rotations (including zero) and justify the economic efficiency of the method.

INTRODUCTION

Azimuth moveout (AMO) is by definition an operator that transforms common-azimuth common-offset seismic reflection data to different azimuths and offsets. A constructive approach to AMO was proposed by Biondi and Chemingui (1994). According to this approach, an AMO operator is built by cascading the dip moveout (DMO) operator that transforms the input common-azimuth data to zero offset, and the inverse DMO that transforms the zero-offset data to a new offset and azimuth. Evaluating the cascade of the frequency-domain DMO and inverse DMO operators by means of the stationary phase technique produces the integral (Kirchhoff-type) 3-D AMO operator in the time-space domain.

The first part of this paper applies an analogous idea to construct the AMO operator from the time-space domain DMO and achieves the same result in a simpler way. Cascading DMO and inverse DMO allows us to evaluate the AMO operator’s summation path and the corresponding weighting function. However, it is not sufficient for evaluating the third major component of the integral operator, that is, its aperture (range of integration). To solve this problem, we apply an alternative approach, that defines AMO as a cascade of 3-D migration (inversion) for particular
common-azimuth and common-offset data and 3-D modeling for a different azimuth and offset. This definition resembles the viewpoint on DMO developed by Deregowski and Rocca (1981). As with the DMO case, the migration and modeling approach reveals the physics of the AMO aperture and limits its boundaries. It is the aperture limitation that allows us to overcome the paradoxical inconsistency between 2-D and 3-D AMO operators discussed by Biondi and Chemingui (1994). If the aperture is chosen properly, the AMO operator converges to the 2-D offset continuation limit as the azimuth rotation approaches zero. This remarkable fact supports the proof of economical efficiency of AMO in comparison with the prestack migration operator, which is known to have an unlimited aperture.

CASCADING DMO AND INVERSE DMO IN TIME-SPACE DOMAIN

In this section, we present a new version of the AMO derivation. Since the entire derivation is performed in the time-space domain, it is more straightforward than the stationary phase technique developed for the same purpose by Biondi and Chemingui (1994).

Let \( P_1(x_1, t_1; h_1) \) be the input of an AMO operator (common-azimuth and common-offset seismic reflection data after normal moveout correction) and \( P_2(x_2, t_2; h_2) \) be the output. Here \( x_i \) \((i = 1, 2)\) are midpoint locations on the surface: \( x_i = \{x_i, y_i\} \), and \( h_i \) are half-offset vectors. The 3-D AMO operator has the following general form:

\[
P_2(x_2, t_2; h_2) = D_{t_2} \int \int w_{12}(x_1; x_2, h_2, t_2) P_1(x_1, t_2 \theta_{12}(x_1; x_2, h_2); h_1) \, dx_1,
\]

where \( D \) is the differentiation operator \( (D_t \equiv \frac{d}{dt}) \), \( t_2 \theta_{12} \) is the summation path, and \( w_{12} \) is the weighting function. In this section we will evaluate \( \theta_{12} \) and \( w_{12} \) using the cascade of integral 3-D DMO and inverse DMO operators in the time-space domain.

The idea of this derivation originated in Biondi and Chemingui’s paper (Biondi and Chemingui, 1994), where it was applied with the frequency-domain DMO and inverse DMO operators in the time-space domain. In the next section, we apply a new geometric approach to evaluate the AMO aperture (range of integration in (1)).

To derive (1) in the time-space domain, an integral (Kirchoff-type) DMO operator of the form

\[
P_0(x_0, t_0; 0) = D^{1/2}_{-t_0} \int w_{10}(x_1; x_0, h_1, t_0) P_1(x_1, t_0 \theta_{10}(x_1; x_0, h_1); h_1) \, dx_1
\]

is cascaded with an inverse DMO of the form

\[
P_2(x_2, t_2; h_2) = D^{1/2}_{t_2} \int w_{02}(x_0; x_2, h_2, t_2) P_0(x_0, t_2 \theta_{02}(x_0; x_2, h_2); 0) \, dx_0
\]
where $D_{1/2}^t$ stands for the operator of half-order differentiation (equivalent to $(i\omega)^{1/2}$ multiplication in Fourier domain), $t_0 \theta_{10}$ and $t_2 \theta_{02}$ are the summation paths of the DMO and inverse DMO operators (Deregowski and Rocca 1981):

$$\theta_{10}(x_1; x_0, h_1) = \left(1 - \frac{(x_1 - x_0)^2}{h_1^2}\right)^{-1/2},$$  \hspace{1cm} (4)

$$\theta_{02}(x_0; x_2, h_2) = \left(1 - \frac{(x_0 - x_2)^2}{h_2^2}\right)^{1/2},$$  \hspace{1cm} (5)

$w_{10}$ and $w_{02}$ are the corresponding weighting functions (amplitudes of impulse responses), $\hat{x}_1$ is the component of $x_1$ along the $h_1$ azimuth, and $\hat{x}_0$ is the component of $x_0$ along the $h_2$ azimuth. Integral operators (2) and (90) correspond to the high-frequency asymptotic (the geometrical seismic) description of the wave field. As shown by Stovas and Fomel (1993), operator (90) has an asymptotically equivalent form

$$P_2(x_2, t_2; h_2) = \int \tilde{w}_{02}(x_0; x_2, h_2, t_2) \, D_{1/2}^t P_0(x_0, t_2 \theta_{02}(x_0; x_2, h_2); 0) \, d\hat{x}_0,$$  \hspace{1cm} (6)

where $\tilde{w}_{02} = w_{02} \sqrt{\theta_{02}}$.

Both DMO and inverse DMO operate on 3-D seismic data as 2-D operators, since their apertures are defined on a line. This implies that for a given input midpoint $x_1$, the corresponding location of $x_0$ must belong to the line going through $x_1$, with the azimuth defined by the input offset $h_1$. Similarly, $x_0$ must be on the line going through $x_2$ with the azimuth of $h_2$ (Figure 1). These theoretical facts lead us to the following conclusion:

For a given pair of input and output midpoints $x_1$ and $x_2$ of the AMO operator, the corresponding midpoint $x_0$ on the intermediate zero-offset gather is determined by the intersection of two lines drawn through $x_1$ and $x_2$ in the offset directions.

Applying the geometric connection among the three midpoints, we can find the cascade of the DMO and inverse DMO operators in one step. For this purpose, it is convenient to choose an orthogonal coordinate system $\{x, y\}$ on the surface in such a way that the direction of the $x$ axis corresponds to the input azimuth (Figure 1). In this case the connection between the three midpoints is given by

$$y_0 = y_1; \quad x_0 = x_2 - (y_2 - y_1) \cot \varphi,$$  \hspace{1cm} (7)

$$d\hat{x}_1 = dx_1; \quad d\hat{x}_0 = dy_1 \csc \varphi.$$  \hspace{1cm} (8)

Substituting (2) into (6) and taking into account (8) produces the 3-D integral
Figure 1: Geometric relationships between input and output midpoint locations in AMO.

AMO operator (1), where

\[
\theta_{12}(x_1; x_2, h_2) = \theta_{02}(x_0; x_2, h_2) \theta_{10}(x_1; x_0, h_1) = \frac{h_1}{h_2} \sqrt{\frac{h_2^2 - (x_2 - x_0)^2}{h_1^2 - (x_1 - x_0)^2}}
\]

\[
= \frac{h_1}{h_2} \sqrt{\frac{h_2^2 \sin \varphi^2 - (y_2 - y_1)^2}{h_1^2 \sin \varphi^2 - ((x_2 - x_1) \sin \varphi - (y_2 - y_1) \cos \varphi)^2}}, \quad (9)
\]

\[
w_{12}(x_1; x_2, h_2, t_2) = w_{02}(x_0; x_2, h_2, t_2) w_{10}(x_1; x_0, h_1, t_2 \theta_{02}(x_0; x_2, h_2)) \csc \varphi \sqrt{\theta_{02}(x_0; x_2, h_2)}, \quad (10)
\]

\[dx = dx_1 dy_1.\] Equation \(t_1 = t_2 \theta_{12}(x_1; x_2, h_2)\) is the same as equation (4) in [Biondi and Chemingui, 1994] except for a different notation. The weighting function of the derived AMO operator \(w_{12}\) depends on the weighting functions of DMO and inverse DMO that are involved in the construction. In Appendix A, we apply equation (10) to two popular versions of the DMO weighting functions that correspond to Hale’s [Hale, 1984] and Zhang’s [Zhang, 1988] DMO operators.

Deriving formula (9), we have to assume that the input and output offset azimuths are different \((\varphi \neq 0)\). In the case of equal azimuths, AMO reduces to 2-D offset continuation (OC). The location of \(x_0\) in this case is not constrained by the input and output midpoints and can take different values on the line. Therefore the superposition of DMO and inverse DMO for offset continuation is a convolution on that line. To find the summation path of the OC operator, we should consider the envelope of the family of traveltime curves (where \(x_0\) is the parameter of a curve in
the family):
\[
t_1 = t_2 \theta_{12} (x_1; x_2, h_2) = t_2 \left| \frac{h_1}{h_2} \right| \sqrt{\frac{h_2^2 - (x_2 - x_0)^2}{h_1^2 - (x_1 - x_0)^2}}. 
\] (11)

Solving the envelope condition
\[
\frac{\partial \theta_{12}}{\partial x_0} = 0
\] (12)

with respect to \( x_0 \) produces
\[
x_0 = \frac{(\Delta x)^2 + h_2^2 - h_1^2 + \text{sign} (h_1^2 - h_2^2) \sqrt{((\Delta x)^2 - h_1^2 - h_2^2)^2 - 4 h_1^2 h_2^2}}{2 (\Delta x)}, \tag{13}
\]

where \( \Delta x = x_1 - x_2 \). Substituting (13) into (11), we get the explicit expression of the OC summation path:
\[
t_1 = t_2 \left| \frac{h_1}{h_2} \right| \sqrt{\frac{U + V}{2}} \text{ for } h_2 > h_1, \\
t_2 |h_1| \sqrt{\frac{2}{U + V}} \text{ for } h_2 < h_1, \tag{14}
\]

where \( U = h_1^2 + h_2^2 - (\Delta x)^2 \), and \( V = \sqrt{U^2 - 4 h_1^2 h_2^2} \). Equation (14) corresponds to formula (6) in [Biondi and Chemingui 1994] (with a typo corrected). The same expression was obtained in a different way by [Stovas and Fomel 1993]. The apparent difference between the 2-D and 3-D solutions introduces the problem of finding a consistent description valid for both cases. Such a description is especially important for practical applications dealing with small angles of azimuth rotation, e.g. cable feather correction in marine seismics. The next section develops a way of solving this problem, which refers to the kinematic theory of AMO and follows the ideas that [Deregowski and Rocca 1981] applied to DMO-type operators.

**AMO APERTURE: CASCADING MIGRATION AND MODELING**

The impulse response of the AMO operators corresponds to a spike on the initial constant-offset constant-azimuth gather. Such a spike can physically occur in the case of a focusing ellipsoidal reflector whose focuses are coincident with the initial source and receiver locations (the impulse response of prestack common-offset migration). Therefore, the impulse response of AMO corresponds kinematically to a reflection from this ellipsoid. These considerations allow us to define AMO as the cascade of the 3-D common-offset common-azimuth migration and the 3-D modeling for a different azimuth and offset. An analogous point of view was developed for the 2-D case by [Deregowski and Rocca 1981].
Let’s consider the general symmetric ellipsoid equation
\[ z(x, y) = \sqrt{R^2 - \beta (x - x_1)^2 - (y - y_1)^2} , \tag{15} \]
where \( z \) stands for the depth coordinate, \( R \) is the small semi-axis of the ellipsoid, and \( \beta \) is a nondimensional parameter describing the stretching of the ellipse \((\beta < 1)\).

Deregowski and Rocca (1981) derived the following connections between the geometric properties of the reflector and the coordinates of the corresponding spike in the data:
\[ R = \frac{v t_1}{2} ; \beta = \frac{t_1^2}{t_1^2 + \frac{4h_1^2}{v^2}} , \tag{16} \]
where \( v \) is the wave velocity. The center of the ellipsoid is at the initial midpoint \( x_1 \).

This section addresses the kinematic problem of reflection from the ellipsoid defined by (B-1). In particular, we are looking for the answer to the following question: 
*For a given elliptic reflector defined by the input midpoint, offset, and time coordinates, what points on the surface can form a source-receiver pair valid for a reflection?*
If a point in the output midpoint-offset space cannot be related to a reflection pattern, we should exclude it from the AMO impulse response defined in (1).

Fermat’s principle provides a general method of solving the kinematic reflection problems. Consider a formal expression for the two-point reflection traveltime
\[ t = \frac{\sqrt{(s - \xi)^2 + z^2(\xi_x, \xi_y)}}{v} + \frac{\sqrt{(r - \xi)^2 + z^2(\xi_x, \xi_y)}}{v} , \tag{17} \]
where \( \xi = \{\xi_x, \xi_y\} \) is the vertical projection of the reflection point to the surface, \( s = \{s_x, s_y\} = x_2 - h_2 \) is the source location, and \( r = \{r_x, r_y\} = x_2 + h_2 \) is the receiver location. According to Fermat’s principle, the reflection ray path between two fixed points must correspond to the extremum value of the traveltime. Hence, in the vicinity of a reflected ray,
\[ \frac{\partial t}{\partial \xi_x} = 0 ; \frac{\partial t}{\partial \xi_y} = 0 . \tag{18} \]
Solving the system of equations (A-4) for \( \xi_x \) and \( \xi_y \) allows us to find the reflection ray path for a given source-receiver pair on the surface. The solution is derived in Appendix B to be
\[ \xi_x = \frac{x_0 - \beta x_1}{1 - \beta} , \tag{19} \]
\[ \xi_y = y_1 + (x_0 - \xi_x) \cot \varphi - \frac{(y_2 - y_1) \left[ (x_0 - \xi_x)^2 - \beta (x_1 - \xi_x)^2 + R^2 \right]}{h_2^2 \sin^2 \varphi - (y_2 - y_1)^2} , \tag{20} \]
where \( x_0 \) has the same meaning as in the preceding section and is defined by (7).

Since the reflection point is contained inside the ellipsoid, its projection obeys the evident inequality
\[ (\xi_y - y_1)^2 \leq R^2 - \beta (\xi_x - x_1)^2 . \tag{21} \]
Figure 2: The AMO impulse response traveltime. Parameters: $|h_1| = 1000$ m, $|h_2| = 750$ m, $t_1 = 1$ sec. The top plots illustrate the case of an unrealistically low velocity ($v = 10$ m/s); on the bottom, $v = 2000$ m/s. On the left side the azimuth rotation $\varphi = 30^\circ$; on the right, $\varphi = 3^\circ$. 

[txamo/app/ amoapp]
It is inequality (21) that defines the aperture of the AMO operator.

The AMO operator’s contours for different azimuth rotation angles are shown in Figure 2. Comparing the results for the case of an unrealistically low velocity (the top two plots in Figure 2) and the case of a realistic velocity (the bottom two plots) clearly demonstrates the gain in the reduction of the aperture size achieved by the aperture limitation. The gain is especially spectacular for small azimuths. When the azimuth rotation approaches zero, the area of the 3-D aperture monotonously shrinks to a line, and the limit of the traveltime of the AMO impulse response (the inverse of (9)) approaches the offset continuation operator (14) (Figures 3). This means that taking into account the aperture limitations of AMO provides a consistent description valid for small azimuth rotations including zero (the offset continuation case). Obviously, the cost of an integral operator is proportional to its size. The size of the offset continuation operator cannot extend the difference between the offsets $|h_1| - |h_2|$. If we applied DMO and inverse DMO explicitly, the total size of the two operators would be about $|h_1| + |h_2|$, which is substantially greater. This fact proves that in the case of small azimuth rotations the AMO price is less than those of not only 3-D prestack migration, but also 3-D DMO and inverse DMO combined (Canning and Gardner, 1992). Figure 4 shows the saddle shape of the AMO operator impulse response in a 3-D AVS display.

CONCLUSIONS

We have applied two different theoretical approaches to AMO to find a complete definition of the integral operator (1). Biondi and Chemingui (1994) proposed cascading the DMO and inverse DMO operators to define AMO in the frequency domain. The same approach is repeated here in a simpler way by transferring the analysis to the natural time-space domain. A new contribution to the evaluation of the AMO operator follows from applying a different approach, which extends the geometric theory of DMO (Deregowski and Rocca, 1981) to the AMO case. Cascading prestack migration and modeling allows us to evaluate the AMO operator aperture. The compactness of the AMO aperture indicates that the integral operator can be performed at a low cost and therefore promises economic benefits for its practical implementation.
Figure 4: AMO impulse response traveltime in three dimensions (the AVS display). Parameters: $|h_1| = 1000$ m, $|h_2| = 750$ m, $t_1 = 1$ sec, $v = 2000$ m/s, $\varphi = 30^\circ$.

REFERENCES


APPENDIX A: AMO AMPLITUDE

The weighting function of the AMO operator can be determined from cascading the DMO and inverse DMO operators by means of equation (10). In the case of Hale’s
DMO (Hale 1984) and its adjoint (Ronen 1987),

\[ w_{10}(x_1; x_0, h_1, t_0) = \sqrt{\frac{t_0}{2\pi}} \frac{|h_1|}{h_2^2 - (x_1 - x_0)^2}, \quad (A-1) \]

\[ w_{02}(x_0; x_2, h_2, t_2) = \sqrt{\frac{t_2}{2\pi}} \frac{|h_2|}{h_2^2 - (x_0 - x_2)^2}. \quad (A-2) \]

As follows from (A-1), (A-2), and (10),

\[ w_{12}(x_1; x_2, h_2, t_2) = \frac{t_2}{2\pi} \times \frac{|h_1| |h_2| \sin \varphi}{(h_1^2 \sin^2 \varphi - ((x_2 - x_1) \sin \varphi - (y_2 - y_1) \cos \varphi)^2) (h_2^2 \sin^2 \varphi - (y_2 - y_1)^2)}. \quad (A-3) \]

In the case of the so-called true-amplitude DMO (Black et al., 1993) and its asymptotic inverse,

\[ w_{10}(x_1; x_0, h_1, t_0) = \sqrt{\frac{t_0}{2\pi}} \frac{h_1^2 + (x_1 - x_0)^2}{|h_1| (h_1^2 - (x_1 - x_0)^2)}, \quad (A-4) \]

\[ w_{02}(x_0; x_2, h_2, t_2) = \sqrt{\frac{t_2}{2\pi}} \frac{|h_2|}{h_2^2 - (x_0 - x_2)^2}. \quad (A-5) \]

Inserting (A-4) and (A-5) into (10) yields

\[ w_{12}(x_1; x_2, h_2, t_2) = \frac{t_2}{2\pi} \frac{|h_2|}{h_1^2 \sin^2 \varphi - ((x_2 - x_1) \sin \varphi - (y_2 - y_1) \cos \varphi)^2) (h_2^2 \sin^2 \varphi - (y_2 - y_1)^2). \quad (A-6) \]

**APPENDIX B: DERIVING THE AMO APERTURE**

This appendix describes the derivation of the main formulas for the aperture evaluation that follow from the Fermat principle (A-4). In order to avoid the algebraic complications of (A-4), we simplify the problem by taking into account the cylindrical symmetry of the ellipsoidal reflector (B-1).

Consider a plane drawn through the reflection point and the central line of the ellipsoid (the axis of the cylindrical symmetry). This plane has to contain the central (normally reflected) ray from the reflector. This conclusion follows from the fact that all the normal reflections emerge at the central line because of the cylindrical symmetry, as shown in Figure B-1. The intersection of the 3-D reflector and the plane is the 2-D ellipse

\[ \hat{z}(x) = \sqrt{R^2 - \beta (x - x_1)^2}. \quad (B-1) \]
The connection between the emergence point of the normal ray $x_0$ and the $x$ coordinate of the reflection point $\xi_x$ can be derived from the relationship evident in Figure B-1 as follows:

$$ x_0 = \xi_x - \hat{z} (\xi_x) \tan \alpha = \xi_x + \hat{z} (\xi_x) \hat{z}' (\xi_x) = \xi_x (1 - \beta) + \beta x_1. $$ (B-2)

Equation (B-2) allows us to evaluate $\xi_x$ in terms of $x_0$ and get (19). The emergence point of the normal ray $x_0$ corresponds to the midpoint on an imaginary zero-offset section (with a coincident source and receiver). Therefore, the location of this point is determined for given input and output midpoints in accordance with expression (7).

Obviously, the reflection point has to be inside the ellipse (B-1). Therefore, its projection obeys the inequality

$$ |\xi_x - x_1| \leq \frac{R}{\sqrt{\beta}}. $$ (B-3)

As follows from (B-3), (B-2), and (16),

$$ |x_0 - x_1| \leq \frac{R (1 - \beta)}{\sqrt{\beta}} = \frac{h_t^2}{\sqrt{\frac{v^2 t_2^2}{2} + h_t^2}}. $$ (B-4)

Inequality (B-4) is the known aperture limitation of the DMO operator (2) found by Deregowski and Rocca (1981). The equality in (B-4) is achieved when the reflection point is on the surface, where the reflector dip increases to 90 degrees.

Now the only unknown left in our problem is the $y$-coordinate of the reflection point $\xi_y$. To find this unknown, we substitute (19) into (17), choosing the convenient parameterization

$$ s = x_0 + h_s; \quad r = x_0 + h_r, $$ (B-5)
where \( h_r - h_s = 2h_2 \), and \( h_r + h_s = 2(x_1 - x_0) \) (Figure B-1). The two-point traveltime function in (17) transforms to the form

\[
t = \frac{\sqrt{(x_0 - \xi_x)^2 - \beta (x_1 - \xi_x)^2 + R^2 + h_2^2 + 2h_2^2 \cdot (x_0 - \xi)}}{v} + \frac{\sqrt{(x_0 - \xi_x)^2 - \beta (x_1 - \xi_x)^2 + R^2 + h_r^2 + 2h_r^2 \cdot (x_0 - \xi)}}{v}.
\]

Applying the second equation from (A-4), we get a simple linear equation for \( \xi_y \), which has the explicit solution (20). From (19) and (20) one can find the reflection point location for given midpoint and offset. To find the limits of possible output midpoint locations, we constrain the reflection point to be inside the ellipsoid (B-1) similarly to the way we did in two dimensions when deriving (B-4). First, let’s consider the case of \( y_2 = y_1 \) (the output midpoint \( x_2 \) is on the line drawn through \( x_1 \) in the direction of the input azimuth). In this case, combining expression (20) and inequality (21) produces

\[
|x_0 - x_1| \leq \frac{R (1 - \beta)}{\sqrt{\beta + \beta^2 \cot^2 \varphi}}.
\]

For any azimuth rotation angle \( \varphi \) less than 90 degrees, the limitation (B-7) is smaller than that of the DMO operator (B-4). The difference increases with the decrease of the azimuth rotation, since the AMO aperture section on the line \( y_2 = y_1 \) monotonously shrinks to a point \( x_2 = x_0 = x_1 \) when \( \varphi \) approaches zero. To extend this conclusion to the whole 3-D aperture, we can find the contour of the aperture by putting the reflection point at the edge of the ellipsoid (B-1), as follows:

\[
(\xi_y - y_1)^2 = R^2 - \beta (\xi_x - x_1)^2
\]

and solving (20) for \( y_2 \). The aperture contour can then be defined by the system of parametric expressions

\[
y_2 (\xi_x) = y_1 + d (\xi_x) \sin \varphi,
\]

\[
x_2 (\xi_x) = \xi_x (1 - \beta) + \beta x_1 + d (\xi_x) \cos \varphi,
\]

where

\[
d (\xi_x) = \frac{d_y^2 + d_x^2 - \sqrt{(d_y^2 + d_x^2)^2 + 4h_2^2 (d_y \sin \varphi + d_x \cos \varphi)^2}}{2 (d_y \sin \varphi + d_x \cos \varphi)},
\]

\[
d_x (\xi_x) = \xi_x - x_0 = \beta (\xi_x - x_1),\text{ and } d_y (\xi_x) = \xi_y - y_1 \text{ is defined by (B-8).}
Theory of differential offset continuation

Sergey Fomel

ABSTRACT

I introduce a partial differential equation to describe the process of prestack reflection data transformation in the offset, midpoint, and time coordinates. The equation is proved theoretically to provide correct kinematics and amplitudes on the transformed constant-offset sections. Solving an initial-value problem with the proposed equation leads to integral and frequency-domain offset continuation operators, which reduce to the known forms of dip moveout operators in the case of continuation to zero offset.

INTRODUCTION

The Earth subsurface is three-dimensional, while seismic reflection data from a multiplicity acquisition belong to a five-dimensional space (time, 2-D offset, and 2-D midpoint coordinates). This fact alone indicates the additional connection that exists in the data space. I show in this paper that it is possible, under certain assumptions, to express this connection in a concise mathematical form of a partial differential equation. The theoretical analysis of this equation allows us to explain and predict the data transformation between different offsets.

The partial differential equation, introduced in this paper, describes the process of offset continuation, which is a transformation of common-offset seismic gathers from one constant offset to another (Bolondi et al., 1982). Bagaini and Spagnolini (1996) identified offset continuation (OC) with a whole family of prestack continuation operators, such as shot continuation (Bagaini and Spagnolini, 1993), dip moveout as a continuation to zero offset (Hale, 1991), and three-dimensional azimuth moveout (Biondi et al., 1998). An intuitive introduction to the concept of offset continuation is presented by Hill et al. (2001). A general data mapping prospective is developed by Bleistein and Jaramillo (2000).

As early as in 1982, Bolondi et al. came up with the idea of describing offset continuation and dip moveout (DMO) as a continuous process by means of a partial differential equation (Bolondi et al., 1982). However, their approximate differential operator, built on the results of Deregowski and Rocca’s classic paper (Deregowski

*e-mail: sergey@sep.stanford.edu, biondo@sep.stanford.edu
†To my knowledge, the first derivation of the revised offset continuation equation was accomplished by Joseph Higginbotham of Texaco in 1989. Unfortunately, Higginbotham’s derivation never appeared in the open literature.
and Rocca (1981), failed in the cases of steep reflector dips or large offsets. Hale (1983) writes:

The differences between this algorithm [DMO by Fourier transform] and previously published finite-difference DMO algorithms are analogous to the differences between frequency-wavenumber (Stolt 1978; Gazdag 1978) and finite-difference (Claerbout 1976) algorithms for migration. For example, just as finite-difference migration algorithms require approximations that break down at steep dips, finite-difference DMO algorithms are inaccurate for large offsets and steep dips, even for constant velocity.

Continuing this analogy, we can observe that both finite-difference and frequency-domain migration algorithms share a common origin: the wave equation. The new OC equation, presented in this paper and valid for all offsets and dips, plays a role analogous to that of the wave equation for offset continuation and dip moveout algorithms. A multitude of seismic migration algorithms emerged from the fundamental wave-propagation theory that is embedded in the wave equation. Likewise, the fundamentals of DMO algorithms can be traced to the OC differential equation.

In the first part of the paper, I prove that the revised equation is, under certain assumptions, kinematically valid. This means that wavefronts of the offset continuation process correspond to the reflection wave traveltimes and correctly transform between different offsets. Moreover, the wave amplitudes are also propagated correctly according to the true-amplitude criterion (Black et al. 1993).

In the second part of the paper, I relate the offset continuation equation to different methods of dip moveout. Considering DMO as a continuation to zero offset, I show that DMO operators can be obtained by solving a special initial value problem for the OC equation. Different known forms of DMO (Hale 1991) appear as special cases of more general offset continuation operators.

The companion paper (Fomel, 2003b) demonstrates a practical application of differential offset continuation to seismic data interpolation.

**INTRODUCING THE OFFSET CONTINUATION EQUATION**

Most of the contents of this paper refer to the following linear partial differential equation:

\[
h \left( \frac{\partial^2 P}{\partial y^2} - \frac{\partial^2 P}{\partial h^2} \right) = t_n \frac{\partial^2 P}{\partial t_n \partial h}. \tag{1}\]

Equation (16) describes an artificial (non-physical) process of transforming reflection seismic data \(P(y, h, t_n)\) in the offset-midpoint-time domain. In equation (16), \(h\) stands for the half-offset \((h = (r - s)/2)\), where \(s\) and \(r\) are the source and the receiver surface
coordinates), \( y \) is the midpoint (\( y = (r + s)/2 \)), and \( t_n \) is the time coordinate after normal moveout correction is applied:

\[
t_n = \sqrt{t^2 - \frac{4h^2}{v^2}}.
\]  

(2)

The velocity \( v \) is assumed to be known a priori. Equation (16) belongs to the class of linear hyperbolic equations, with the offset \( h \) acting as a time-like variable. It describes a wave-like propagation in the offset direction.

**Proof of validity**

A simplified version of the ray method technique (Cerveny, 2001; Babich, 1991) can allow us to prove the theoretical validity of equation (16) for all offsets and reflector dips by deriving two equations that describe separately wavefront (traveltime) and amplitude transformation. According to the formal ray theory, the leading term of the high-frequency asymptotics for a reflected wave recorded on a seismogram takes the form

\[
P(y, h, t_n) \approx A_n(y, h) R_n(t_n - \tau_n(y, h)),
\]

(3)

where \( A_n \) stands for the amplitude, \( R_n \) is the wavelet shape of the leading high-frequency term, and \( \tau_n \) is the traveltime curve after normal moveout. Inserting (18) as a trial solution for (16), collecting terms that have the same asymptotic order (correspond to the same-order derivatives of the wavelet \( R_n \)), and neglecting low-order terms, we arrive at the set of two first-order partial differential equations:

\[
h \left[ \left( \frac{\partial \tau_n}{\partial y} \right)^2 - \left( \frac{\partial \tau_n}{\partial h} \right)^2 \right] = -\tau_n \frac{\partial \tau_n}{\partial h},
\]

(4)

\[
(\tau_n - 2h \frac{\partial \tau_n}{\partial h}) \frac{\partial A_n}{\partial h} + 2h \frac{\partial \tau_n}{\partial y} \frac{\partial A_n}{\partial y} + hA_n \left( \frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2} \right) = 0 .
\]

(5)

Equation (19) describes the transformation of traveltime curve geometry in the OC process analogously to how the eikonal equation describes the front propagation in the classic wave theory. What appear to be wavefronts of the wave motion described by equation (16) are traveltime curves of reflected waves recorded on seismic sections. The law of amplitude transformation for high-frequency wave components related to those wavefronts is given by equation (20). In terms of the theory of partial differential equations, equation (19) is the characteristic equation for (16).

**Proof of kinematic equivalence**

In order to prove the validity of equation (19), it is convenient to transform it to the coordinates of the initial shot gathers: \( s = y - h \), \( r = y + h \), and \( \tau = \sqrt{\tau_n^2 + \frac{4h^2}{v^2}} \). The
transformed equation takes the form
\[
\left( \tau^2 + \frac{(r - s)^2}{v^2} \right) \left( \frac{\partial \tau}{\partial r} - \frac{\partial \tau}{\partial s} \right) = 2 (r - s) \tau \left( \frac{1}{v^2} - \frac{\partial \tau}{\partial r} \frac{\partial \tau}{\partial s} \right) .
\] (6)

Now the goal is to prove that any reflection traveltime function \( \tau(r, s) \) in a constant velocity medium satisfies equation (21).

Let \( S \) and \( R \) be the source and the receiver locations, and \( O \) be a reflection point for that pair. Note that the incident ray \( SO \) and the reflected ray \( OR \) form a triangle with the basis on the offset \( SR \) (\( l = |SR| = |r - s| \)). Let \( \alpha_1 \) be the angle of \( SO \) from the vertical axis, and \( \alpha_2 \) be the analogous angle of \( RO \) (Figure 1). The law of sines gives us the following explicit relationships between the sides and the angles of the triangle \( SOR \):
\[
|SO| = |SR| \frac{\cos \alpha_2}{\sin (\alpha_2 - \alpha_1)} ,
\] (7)
\[
|RO| = |SR| \frac{\cos \alpha_1}{\sin (\alpha_2 - \alpha_1)} .
\] (8)

Hence, the total length of the reflected ray satisfies
\[
v \tau = |SO| + |RO| = |SR| \frac{\cos \alpha_1 + \cos \alpha_2}{\sin (\alpha_2 - \alpha_1)} = |r - s| \frac{\cos \alpha}{\sin \gamma} .
\] (9)

Here \( \gamma \) is the reflection angle (\( \gamma = (\alpha_2 - \alpha_1)/2 \)), and \( \alpha \) is the central ray angle (\( \alpha = (\alpha_2 + \alpha_1)/2 \)), which coincides with the local dip angle of the reflector at the reflection point. Recalling the well-known relationships between the ray angles and the first-order traveltine derivatives
\[
\frac{\partial \tau}{\partial s} = \frac{\sin \alpha_1}{v} ,
\] (10)
\[
\frac{\partial \tau}{\partial r} = \frac{\sin \alpha_2}{v} ,
\] (11)
we can substitute (9), (10), and (11) into (21), which leads to the simple trigonometric equality
\[
\cos^2 \left( \frac{\alpha_1 + \alpha_2}{2} \right) + \sin^2 \left( \frac{\alpha_1 - \alpha_2}{2} \right) = 1 - \sin \alpha_1 \sin \alpha_2 .
\] (12)

It is now easy to show that equality (12) is true for any \( \alpha_1 \) and \( \alpha_2 \), since
\[
\sin^2 a - \sin^2 b = \sin (a + b) \sin (a - b) .
\]

Thus we have proved that equation (21), equivalent to (19), is valid in constant velocity media independently of the reflector geometry and the offset. This means that high-frequency asymptotic components of the waves, described by the OC equation, are located on the true reflection traveltine curves.

The theory of characteristics can provide other ways to prove the kinematic validity of equation (19), as described by Fomel (1994) and Goldin (1994).
Comparison with Bolondi’s OC equation

Equation (16) and the previously published OC equation (Bolondi et al., 1982) differ only with respect to the single term \( \frac{\partial^2 P}{\partial y^2} \). However, this difference is substantial.

From the offset continuation characteristic equation (19), we can conclude that the first-order traveltime derivative with respect to offset decreases with decreasing offset. The derivative equals zero at the zero offset, as predicted by the principle of reciprocity (the reflection traveltime has to be an even function of offset). Neglecting \( \left( \frac{\partial \tau_n}{\partial h} \right)^2 \) in (19) leads to the characteristic equation

\[
h \left( \frac{\partial \tau_n}{\partial y} \right)^2 = -\tau_n \frac{\partial \tau_n}{\partial h},
\]

which corresponds to the approximate OC equation of Bolondi et al. (1982). The approximate equation has the form

\[
h \frac{\partial^2 P}{\partial y^2} = \tau_n \frac{\partial^2 P}{\partial t_n \partial h}.
\]

Comparing equations (13) and (19), we can note that approximation (13) is valid only if

\[
\left( \frac{\partial \tau_n}{\partial h} \right)^2 \ll \left( \frac{\partial \tau_n}{\partial y} \right)^2.
\]

To find the geometric constraints implied by inequality (15), we can express the traveltime derivatives in geometric terms. As follows from expressions (10) and (11),

\[
\frac{\partial \tau}{\partial y} = \frac{\partial \tau}{\partial r} + \frac{\partial \tau}{\partial s} = \frac{2 \sin \alpha \cos \gamma}{v},
\]

\[
\frac{\partial \tau}{\partial h} = \frac{\partial \tau}{\partial r} - \frac{\partial \tau}{\partial s} = \frac{2 \cos \alpha \sin \gamma}{v}.
\]
Expression (9) allows transforming equations (16) and (17) to the form

\[ \tau_n \frac{\partial \tau_n}{\partial y} = \tau \frac{\partial \tau}{\partial y} = 4h \sin \alpha \cos \alpha \cot \gamma ; \quad (18) \]

\[ \tau_n \frac{\partial \tau_n}{\partial h} = \tau \frac{\partial \tau}{\partial h} - \frac{4h}{v^2} = -4h \frac{\sin^2 \alpha}{v^2} . \quad (19) \]

Without loss of generality, we can assume \( \alpha \) to be positive. Consider a plane tangent to a true reflector at the reflection point (Figure 2). The traveltime of a wave, reflected from the plane, has the known explicit expression

\[ \tau = \frac{2}{v} \sqrt{L^2 + h^2 \cos^2 \alpha} , \quad (20) \]

where \( L \) is the length of the normal ray from the midpoint. As follows from combining (20) and (9),

\[ \cos \alpha \cot \gamma = \frac{L}{h} . \quad (21) \]

We can now combine equations (12), (18), and (19) to transform inequality (15) to the form

\[ h \ll \frac{L}{\sin \alpha} = z \cot \alpha , \quad (22) \]

where \( z \) is the depth of the plane reflector under the midpoint. For example, for a dip of 45 degrees, equation (14) will be satisfied only for offsets that are much smaller than the depth of the reflector.

Figure 2: Reflection rays and tangent to the reflector in a constant velocity medium (a scheme).

**Offset continuation geometry: time rays**

To study the laws of travelttime curve transformation in the OC process, it is convenient to apply the method of characteristics (Courant, 1962) to the eikonal-type equation (19). The characteristics of equation (19) \( \text{[b]-characteristics with respect to equation (16)} \) are the trajectories of the high-frequency energy propagation in the
imaginary OC process. Following the formal analogy with seismic rays, I call those trajectories \textit{time rays}, where the word \textit{time} refers to the fact that the trajectories describe the traveltime transformation \cite{Fomel94}. According to the theory of first-order partial differential equations, time rays are determined by a set of ordinary differential equations (characteristic equations) derived from equation (19):

\begin{align}
\frac{dy}{dt_n} &= -\frac{2hY}{t_n H} , \quad \frac{dY}{dt_n} = \frac{Y}{t_n} , \\
\frac{dh}{dt_n} &= -\frac{1}{H} + \frac{2h}{t_n} , \quad \frac{dH}{dt_n} = \frac{Y^2}{t_n H} ,
\end{align}

(23)

where \(Y\) corresponds to \(\frac{\partial \tau_n}{\partial y}\) along a ray and \(H\) corresponds to \(\frac{\partial \tau_n}{\partial h}\). In this notation, equation (19) takes the form

\[ h (Y^2 - H^2) = -t_n H \]

(24)

and serves as an additional constraint for the definition of time rays. System (23) can be solved by standard mathematical methods \cite{TenenbaumPollard85}. Its general solution takes the parametric form, where the time variable \(t_n\) is the parameter changing along a time ray:

\[ y(t_n) = C_1 - C_2 \frac{t^2_n}{t^2_0 (y)} ; \quad h(t_n) = t_n \sqrt{C_2^2 t^2_n + C_3} ; \]

(25)

\[ Y(t_n) = \frac{C_2 t_n}{C_3} ; \quad H(t_n) = \frac{h}{C_3 t_n} \]

(26)

and \(C_1, C_2,\) and \(C_3\) are independent coefficients, constant along each time ray. To find the values of these coefficients, we can pose an initial-value problem for the system of differential equations (23). The traveltime curve \(\tau_n(y; h)\) for a given common offset \(h\) and the first partial derivative \(\frac{\partial \tau_n}{\partial h}\) along the same constant offset section provide natural initial conditions. A particular case of those conditions is the zero-offset traveltime curve. If the first partial derivative of traveltime with respect to offset is continuous, it vanishes at zero offset according to the reciprocity principle (traveltime must be an even function of the offset):

\[ t_0 (y_0) = \tau_n(y; 0) , \quad \frac{\partial \tau_n}{\partial h} \bigg|_{h=0} = 0 . \]

Applying the initial-value conditions to the general solution (26) generates the following expressions for the ray invariants:

\[ C_1 = \frac{h + \frac{Y}{H}}{y} = y_0 - \frac{t_0 (y_0)}{t_0' (y_0)} ; \quad C_2 = \frac{h Y}{\tau^2_n H} = -\frac{1}{t_0 (y_0) t_0' (y_0)} ; \]

\[ C_3 = \frac{h}{t_0 (y_0) H} = -\frac{1}{(t_0' (y_0))^2} , \]

(27)

where \(t_0' (y_0)\) denotes the derivative \(\frac{dt_0}{dy}\). Finally, substituting equations (27) into (26), we obtain an explicit parametric form of the ray trajectories:

\[ y_1 (t_1) = y + \frac{h Y}{t_0^2 H} \left( t^2_n - t^2_1 \right) = y_0 + \frac{t^2_1 - t^2_0 (y_0)}{t_0 (y_0) t_0' (y_0)} ; \]

(28)

\[ h^2_1 (t_1) = \frac{h Y}{t^3_0 H} \left( t^2_n + t^2_1 \frac{h Y^2}{t_n H} \right) = t^2_1 \left( \frac{t^2_1 - t^2_0 (y_0)}{t_0 (y_0) t_0' (y_0)} \right)^2 . \]

(29)
Here \( y_1, h_1, \) and \( t_1 \) are the coordinates of the continued seismic section. Equations (28) indicates that the time ray projections to a common-offset section have a parabolic form. Time rays do not exist for \( t'_0(y_0) = 0 \) (a locally horizontal reflector) because in this case post-NMO offset continuation transform is not required.

The actual parameter that determines a particular time ray is the reflection point location. This important conclusion follows from the known parametric equations

\[
\begin{align*}
  t_0(x) &= t_v \sec \alpha = t_v(x) \sqrt{1 + u^2 \left( t'_v(x) \right)^2}, \\
  y_0(x) &= x + ut_v \tan \alpha = x + u^2 t_v(x) t'_v(x),
\end{align*}
\]

where \( x \) is the reflection point, \( u \) is half of the wave velocity (\( u = v/2 \)), \( t_v \) is the vertical time (reflector depth divided by \( u \)), and \( \alpha \) is the local reflector dip. Taking into account that the derivative of the zero-offset traveltime curve is

\[
\frac{dt_0}{dy_0} = \frac{t'_0(x)}{y'_0(x)} = \frac{\sin \alpha}{u} = \frac{t'_v(x)}{\sqrt{1 + u^2 \left( t'_v(x) \right)^2}}
\]

and substituting equations (30) and (31) into (28) and (29), we get

\[
\begin{align*}
  y_1(t_1) &= x + \frac{t_1^2 - t_v^2(x)}{t_v(x) t'_v(x)}; \\
  u^2 t^2(t_1) &= t_1^2 \frac{t_1^2 - t_v^2(x)}{(t_v(x) t'_v(x))^2},
\end{align*}
\]

where \( t^2(t_1) = t_1^2 + h_1^2(t_1)/u^2 \).

To visualize the concept of time rays, let us consider some simple analytic examples of its application to geometric analysis of the offset-continuation process.

**Example 1: plane reflector**

The simplest and most important example is the case of a plane dipping reflector. Putting the origin of the \( y \) axis at the intersection of the reflector plane with the surface, we can express the reflection traveltie after NMO in the form

\[
\tau_n(y, h) = p \sqrt{y^2 - h^2},
\]

where \( p = 2 \frac{\sin \alpha}{v} \), and \( \alpha \) is the dip angle. The zero-offset traveltime in this case is a straight line:

\[
t_0(y_0) = p y_0.
\]

According to equations (28, 29), the time rays in this case are defined by

\[
\begin{align*}
  y_1(t_1) &= \frac{t_1^2}{p^2 y_0}; \\
  h_1^2(t_1) &= \frac{t_1^2 - p^2 y_0^2}{p^4 y_0^2}; \\
  y_0 &= \frac{y^2 - h^2}{y}.
\end{align*}
\]

The geometry of the OC transformation is shown in Figure 3.
Figure 3: Transformation of the reflection traveltime curves in the OC process: the case of a plane dipping reflector. Left: Time coordinate before the NMO correction. Right: Time coordinate after NMO. The solid lines indicate traveltime curves at different common-offset sections; the dashed lines indicate time rays.

Example 2: point diffractor

The second example is the case of a point diffractor (the left side of Figure 4). Without loss of generality, the origin of the midpoint axis can be put above the diffraction point. In this case the zero-offset reflection traveltime curve has the well-known hyperbolic form

\[ t_0(y_0) = \frac{\sqrt{z^2 + y_0^2}}{u}, \]  

where \( z \) is the depth of the diffractor and \( u = v/2 \) is half of the wave velocity. Time rays are defined according to equations (28-29), as follows:

\[ y_1(t_1) = \frac{u^2 t_1^2 - z^2}{y_0}; \quad u^2 t_1^2 + h_1^2(t_1) = u^2 t_1^2 \frac{u^2 t_1^2 - z^2}{y_0^2}. \]  

Example 3: elliptic reflector

The third example (the right side of Figure 4) is the curious case of a focusing elliptic reflector. Let \( y \) be the center of the ellipse and \( h \) be half the distance between the foci of the ellipse. If both foci are on the surface, the zero-offset traveltime curve is defined by the so-called “DMO smile” \cite{Deregowski1981}:

\[ t_0(y_0) = \frac{t_\alpha}{h} \sqrt{h^2 - (y - y_0)^2}, \]  

where \( \alpha \) is the angle of the ellipse.
Figure 4: Transformation of the reflection traveltime curves in the OC process. Left: the case of a diffraction point. Right: the case of an elliptic reflector. Solid lines indicate traveltime curves at different common-offset sections, dashed lines indicate time rays.

where \( t_n = 2z/v \), and \( z \) is the small semi-axis of the ellipse. The time-ray equations are

\[
y_1(t_1) = y + \frac{h^2}{y - y_0} \left( t_1^2 - t_n^2 \right) ; \quad h_1(t_1) = h^2 \left( t_1^2 - t_n^2 \right) \left( 1 + \frac{h^2}{(y - y_0)^2} \right) \left( t_1^2 - t_n^2 \right).
\]

(41)

When \( y_1 \) coincides with \( y \), and \( h_1 \) coincides with \( h \), the source and the receiver are in the foci of the elliptic reflector, and the traveltime curve degenerates to a point \( t_1 = t_n \). This remarkable fact is the actual basis of the geometric theory of dip moveout (Deregowski and Rocca, 1981).

**Proof of amplitude equivalence**

Let us now consider the connection between the laws of traveltime transformation and the laws of the corresponding amplitude transformation. The change of the wave amplitudes in the OC process is described by the first-order partial differential transport equation (20). We can find the general solution of this equation by applying the method of characteristics. The solution takes the explicit integral form

\[
A_n(t_n) = A_0(t_0) \exp \left( \int_{t_0}^{t_n} \left[ h \left( \frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2} \right) \left( \tau_n \frac{\partial \tau_n}{\partial h} \right)^{-1} \right] d\tau_n \right).
\]

(42)

The integral in equation (42) is defined on a curved time ray, and \( A_n(t_n) \) stands for the amplitude transported along this ray. In the case of a plane dipping reflector, the ray
amplitude can be immediately evaluated by substituting the explicit traveltime and time ray equations from the preceding section into (42). The amplitude expression in this case takes the simple form

$$A_n(t_n) = A_0(t_0) \exp \left( - \int_{t_0}^{t_n} \frac{d\tau_n}{\tau_n} \right) = A_0(t_0) \frac{t_0}{t_n}. \quad (43)$$

In order to consider the more general case of a curvilinear reflector, we need to take into account the connection between the traveltime derivatives in (42) and the geometry of the reflector. As follows directly from the trigonometry of the incident and reflected rays triangle (Figure 1),

$$h = \frac{r - s}{2} = D \frac{\cos \alpha \sin \gamma \cos \gamma}{\cos^2 \alpha - \sin^2 \gamma}, \quad (44)$$

$$y = \frac{r + s}{2} = x + D \frac{\cos^2 \alpha \sin \alpha}{\cos^2 \alpha - \sin^2 \gamma}, \quad (45)$$

$$y_0 = x + D \sin \alpha, \quad (46)$$

where $D$ is the length of the normal ray. Let $\tau_0 = 2D/v$ be the zero-offset reflection traveltime. Combining equations (44) and (46) with (9), we can get the following relationship:

$$a = \frac{\tau_n}{\tau_0} = \frac{\cos \alpha \cos \gamma}{\left(\cos^2 \alpha - \sin^2 \gamma\right)^{1/2}} \left(1 + \frac{\sin^2 \alpha \sin^2 \gamma}{\cos^2 \alpha - \sin^2 \gamma}\right)^{1/2} = \frac{h}{\sqrt{h^2 - (y - y_0)^2}}, \quad (47)$$

which describes the “DMO smile” [1] found by [2] in geometric terms. Equation (47) allows for a convenient change of variables in equation (42). Let the reflection angle $\gamma$ be a parameter monotonically increasing along a time ray. In this case, each time ray is uniquely determined by the position of the reflection point, which in turn is defined by the values of $D$ and $\alpha$. According to this change of variables, we can differentiate (47) along a time ray to get

$$\frac{d\tau_n}{\tau_n} = -\frac{\sin^2 \alpha}{2 \cos^2 \gamma \left(\cos^2 \gamma - \sin^2 \alpha\right)} \frac{d}{d\left(\cos^2 \gamma\right)} \left(\cos^2 \gamma\right). \quad (48)$$

Note also that the quantity $h \left(\tau_n \frac{\partial \tau_n}{\partial h}\right)^{-1}$ in equation (42) coincides exactly with the time ray invariant $C_3$ found in equation (27). Therefore its value is constant along each time ray and equals

$$h \left(\tau_n \frac{\partial \tau_n}{\partial h}\right)^{-1} = -\frac{v^2}{4 \sin^2 \alpha}. \quad (49)$$

Finally, as shown in Appendix [3]

$$\tau_n \left(\frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2}\right) = 4 \frac{\cos^2 \gamma}{v^2} \left(\cos^2 \gamma + DK\right), \quad (50)$$
where $K$ is the reflector curvature at the reflection point. Substituting (48), (49), and (50) into (42) transforms the integral to the form

$$
\int_{t_0}^{t_n} \left[ h \left( \frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2} \right) \left( \tau_n \frac{\partial \tau_n}{\partial h} \right)^{-1} \right] d\tau_n =
$$

$$
= -\frac{1}{2} \int_{\cos^2 \gamma_0}^{\cos^2 \gamma} \left( \frac{1}{\cos^2 \gamma - \sin^2 \alpha} - \frac{1}{\cos^2 \gamma + DK} \right) d\left( \cos^2 \gamma \right)
$$

which we can evaluate analytically. The final equation for the amplitude transformation is

$$
A_n = A_0 \frac{\sqrt{\cos^2 \gamma - \sin^2 \alpha}}{\sqrt{\cos^2 \gamma_0 - \sin^2 \alpha}} \left( \frac{\cos^2 \gamma_0 + DK}{\cos^2 \gamma + DK} \right)^{1/2} =
$$

$$
= A_0 \frac{\tau_0 \cos \gamma}{\tau_n \cos \gamma_0} \left( \frac{\cos^2 \gamma_0 + DK}{\cos^2 \gamma + DK} \right)^{1/2}.
$$

(51)

In case of a plane reflector, the curvature $K$ is zero, and equation (52) coincides with (43). In the general case can be rewritten as

$$
A_n = \frac{c \cos \gamma}{\tau_n \sqrt{\cos^2 \gamma + DK}},
$$

(53)

where $c$ is constant along each time ray (it may vary with the reflection point location on the reflector but not with the offset). We should compare equation (53) with the known expression for the reflection wave amplitude of the leading ray series term in 2.5-D media (Bleistein et al., 2001):

$$
A = \frac{C_R(\gamma)\Psi}{G},
$$

(54)

where $C_R$ stands for the angle-dependent reflection coefficient, $G$ is the geometric spreading

$$
G = v\tau \sqrt{\frac{\cos^2 \gamma + DK}{\cos \gamma}},
$$

(55)

and $\Psi$ includes other possible factors (such as the source directivity) that we can either correct or neglect in the preliminary processing. It is evident that the curvature dependence of the amplitude transformation (53) coincides completely with the true geometric spreading factor (55) and that the angle dependence of the reflection coefficient is not accounted for the offset continuation process. If the wavelet shape of the reflected wave on seismic sections [$R_n$ in equation (18)] is described by the delta function, then, as follows from the known properties of this function,

$$
A \delta (t - \tau (y, h)) = \left| \frac{dt_n}{dt} \right| A \delta (t_n - \tau_n (y, h)) = \frac{t}{t_n} A \delta (t_n - \tau_n (y, h)),
$$

(56)
which leads to the equality
\[ A_n = A \frac{t}{t_n} . \] (57)
Combining equation (57) with equations (54) and (53) allows us to evaluate the amplitude after continuation from some initial offset \( h_0 \) to another offset \( h_1 \), as follows:
\[ A_1 = \frac{C_R(\gamma_0) \Psi_0}{G_1} . \] (58)
According to equation (58), the OC process described by equation (16) is amplitude-preserving in the sense that corresponds to the definition of Born DMO (Bleistein, 1990; Liner, 1991). This means that the geometric spreading factor from the initial amplitudes is transformed to the true geometric spreading on the continued section, while the reflection coefficient stays the same. This remarkable dynamic property allows AVO (amplitude versus offset) analysis to be performed by a dynamic comparison between true constant-offset sections and the sections transformed by OC from different offsets. With a simple trick, the offset coordinate is transferred to the reflection angles for the AVO analysis. As follows from (47) and (9),
\[ \frac{\tau^2}{\tau_0} = \cos \gamma . \] (59)
If we include the \( \frac{t^2}{t_{n0}} \) factor in the DMO operator (continuation to zero offset) and divide the result by the DMO section obtained without this factor, the resultant amplitude of the reflected events will be directly proportional to \( \cos \gamma \), where the reflection angle \( \gamma \) corresponds to the initial offset. Of course, this conclusion is rigorously valid for constant-velocity 2.5-D media only.

Black et al. (1993) suggest a definition of true-amplitude DMO different from that of Born DMO. The difference consists of two important components:

1. **True-amplitude DMO addresses preserving the peak amplitude of the image wavelet instead of preserving its spectral density.** In the terms of this paper, the peak amplitude corresponds to the pre-NMO amplitude \( A \) from formula (54) instead of corresponding to the spectral density amplitude \( A_n \). A simple correction factor \( \frac{t}{t_n} \) would help us take the difference between the two amplitudes into account. Multiplication by \( \frac{t}{t_n} \) can be easily done at the NMO stage.

2. **Seismic sections are multiplied by time to correct for the geometric spreading factor prior to DMO (or, in our case, offset continuation) processing.**

As follows from (55), multiplication by \( t \) is a valid geometric spreading correction for plane reflectors only. It is the amplitude-preserving offset continuation based on the OC equation (16) that is able to correct for the curvature-dependent factor in the amplitude. To take into account the second aspect of Black’s definition, we can consider the modified field \( \hat{P} \) such that
\[ \hat{P}(y, h, t_n) = t P(y, h, t_n) . \] (60)
Substituting (60) into the OC equation (16) transforms the latter to the form

$$h \left( \frac{\partial^2 \hat{P}}{\partial y^2} - \frac{\partial^2 \hat{P}}{\partial h^2} \right) = t_n \frac{\partial^2 \hat{P}}{\partial t_n \partial h} - \frac{\partial \hat{P}}{\partial h}. \quad (61)$$

Equations (61) and (16) differ only with respect to the first-order damping term $\frac{\partial \hat{P}}{\partial h}$. This term affects the amplitude behavior but not the traveltimes, since the eikonal-type equation (19) depends on the second-order terms only. Offset continuation operators based on (61) conform to Black’s definition of true-amplitude processing.

Fomel and Bleistein (2001) describe an alternative approach to confirming the kinematic and amplitude validity of the offset continuation equation. Applying equation (16) directly on the Kirchhoff model of prestack seismic data shows that the equation is satisfied to the same asymptotic order of accuracy as the Kirchhoff modeling approximation (Haddon and Buchen, 1981; Bleistein, 1984).

**INTEGRAL OFFSET CONTINUATION OPERATOR**

Equation (16) describes a continuous process of reflected wavefield continuation in the time-offset-midpoint domain. In order to find an integral-type operator that performs the one-step offset continuation, I consider the following initial-value problem for equation (16):

**Given a post-NMO constant-offset section at half-offset $h_1$**

$$P(t_n, h, y)|_{h=\frac{h_1}{2}} = P^{(0)}_1(t_n, y) \quad (62)$$

and its first-order derivative with respect to offset

$$\frac{\partial P(t_n, h, y)}{\partial h}|_{h=\frac{h_1}{2}} = P^{(1)}_1(t_n, y), \quad (63)$$

**find the corresponding section $P^{(0)}(t_n, y)$ at offset $h$.**

Equation (16) belongs to the hyperbolic type, with the offset coordinate $h$ being a “time-like” variable and the midpoint coordinate $y$ and the time $t_n$ being “space-like” variables. The last condition (63) is required for the initial value problem to be well-posed (Courant, 1962). From a physical point of view, its role is to separate the two different wave-like processes embedded in equation (16), which are analogous to inward and outward wave propagation. We will associate the first process with continuation to a larger offset and the second one with continuation to a smaller offset. Though the offset derivatives of data are not measured in practice, they can be estimated from the data at neighboring offsets by a finite-difference approximation. Selecting a propagation branch explicitly, for example by considering the high-frequency asymptotics of the continuation operators, can allow us to eliminate the need for condition (63). In this section, I discuss the exact integral solution of the OC equation and analyze its asymptotics.
The integral solution of problem \[62,63\] for equation \[16\] is obtained with the help of the classic methods of mathematical physics \[Fomel, 1994, 2001\]. It takes the explicit form

\[
P(t_n, h, y) = \int \int P_1^{(0)}(t_1, y_1) G_0(t_1, h_1, y_1; t_n, h, y) dt_1 dy_1 
+ \int \int P_1^{(1)}(t_1, y_1) G_1(t_1, h_1, y_1; t_n, h, y) dt_1 dy_1 ,
\]

where the Green’s functions \(G_0\) and \(G_1\) are expressed as

\[
G_0(t_1, h_1, y_1; t_n, h, y) = \text{sign}(h - h_1) \frac{H(t_n)}{\pi} \frac{\partial}{\partial t_n} \left\{ \frac{H(\Theta)}{\sqrt{\Theta}} \right\} ,
\]

\[
G_1(t_1, h_1, y_1; t_n, h, y) = \text{sign}(h - h_1) \frac{H(t_n)}{\pi} h \frac{t_n}{t_1^2} \left\{ \frac{H(\Theta)}{\sqrt{\Theta}} \right\} ,
\]

and the parameter \(\Theta\) is

\[
\Theta(t_1, h_1, y_1; t_n, h, y) = \left( h_1^2/t_1^2 - h^2/t_n^2 \right) \left( t_1^2 - t_n^2 \right) - (y_1 - y)^2 .
\]

\(H\) stands for the Heaviside step-function.

From equations \[65\] and \[66\] one can see that the impulse response of the offset continuation operator is discontinuous in the time-offset-midpoint space on a surface defined by the equality

\[
\Theta(t_1, h_1, y_1; t_n, h, y) = 0 ,
\]

which describes the “wavefronts” of the offset continuation process. In terms of the theory of characteristics \[Courant, 1962\], the surface \(\Theta = 0\) corresponds to the characteristic conoid formed by the bi-characteristics of equation \[16\] - time rays emerging from the point \(\{t_n, h, y\} = \{t_1, h_1, y_1\}\). The common-offset slices of the characteristic conoid are shown in the left plot of Figure \[5\].

As a second-order differential equation of the hyperbolic type, equation \[16\] describes two different processes. The first process is “forward” continuation from smaller to larger offsets, the second one is “reverse” continuation in the opposite direction. These two processes are clearly separated in the high-frequency asymptotics of operator \[64\]. To obtain the asymptotic representation, it is sufficient to note that \(\frac{1}{\sqrt{\pi}} \frac{H(t)}{\sqrt{\sqrt{t^2-a^2}}} = \frac{H(t-a)}{\sqrt{2a\sqrt{t-a}}}(t, a > 0)\). Thus, the asymptotical form of the integral offset-continuation operator becomes

\[
P^{(\pm)}(t_n, h, y) = D_{\pm t_n}^{1/2} \int w^{(\pm)}(\xi; h_1, h, t_n) P_1^{(0)}(\theta^{(\pm)}(\xi; h_1, h, t_n), y_1 - \xi) d\xi 
\pm I_{\pm t_n}^{1/2} \int w^{(\pm)}(\xi; h_1, h, t_n) P_1^{(1)}(\theta^{(\pm)}(\xi; h_1, h, t_n), y_1 - \xi) d\xi .
\]

Here the signs “+” and “−” correspond to the type of continuation (the sign of \(h - h_1\)), \(D_{\pm t_n}^{1/2}\) and \(I_{\pm t_n}^{1/2}\) stand for the operators of causal and anticausal half-order
Figure 5: Constant-offset sections of the characteristic conoid - “offset continuation fronts” (left), and branches of the conoid used in the integral OC operator (right). The upper part of the plots (small times) corresponds to continuation to smaller offsets; the lower part (large times) corresponds to larger offsets.

differentiation and integration applied with respect to the time variable $t_n$, the summation paths $\theta^{(\pm)}(\xi; h_1, h, t_n)$ correspond to the two non-negative sections of the characteristic conoid (68) (Figure 5):

$$t_1 = \theta^{(\pm)}(\xi; h_1, h, t_n) = \frac{t_n}{h} \sqrt{\frac{U \pm V}{2}},$$

(70)

where $U = h^2 + h_1^2 - \xi^2$, and $V = \sqrt{U^2 - 4 h^2 h_1^2}$. $\xi$ is the midpoint separation (the integration parameter), and $w_0^{(\pm)}$ and $w_1^{(\pm)}$ are the following weighting functions:

$$w_0^{(\pm)} = \frac{1}{\sqrt{2\pi}} \frac{\theta^{(\pm)}(\xi; h_1, h, t_n)}{\sqrt{t_n V}},$$

(71)

$$w_1^{(\pm)} = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{t_n h_1}}{\sqrt{V} \theta^{(\pm)}(\xi; h_1, h, t_n)}.$$

(72)

Expression (70) for the summation path of the OC operator was obtained previously by Stovas and Fomel (1996) and Biondi and Chemingui (1994). A somewhat different form of it is proposed by Bagaini and Spagnolini (1996). I describe the kinematic interpretation of formula (70) in Appendix B.

In the high-frequency asymptotics, it is possible to replace the two terms in equation (69) with a single term (Fomel et al., 2003a). The single-term expression is

$$P^{(\pm)}(t_n, h, y) = D_{\pm t_n}^{1/2} \int w^{(\pm)}(\xi; h_1, h, t_n) P_1^{(0)}(\theta^{(\pm)}(\xi; h_1, h, t_n), y_1 - \xi) d\xi,$$

(73)
where

\[
\begin{align*}
w^{(+)} &= \sqrt{\frac{2\pi}{V^3/2}} \frac{\theta^{(+)}(\xi; h_1, h, t_n) h^2 - h_1^2 - \xi^2}{V^3/2}, \\
w^{(-)} &= \sqrt{2\pi t_n} \frac{\theta^{(-)}(\xi; h_1, h, t_n) h_1^2 - h^2 + \xi^2}{V^3/2}.
\end{align*}
\]

(74)  (75)

A more general approach to true-amplitude asymptotic offset continuation is developed by ?.

The limit of expression (70) for the output offset \( h \) approaching zero can be evaluated by L'Hospital's rule. As one would expect, it coincides with the well-known expression for the summation path of the integral DMO operator (Deregowski and Rocca, 1981)

\[
t_1 = \theta^{(-)}(\xi; h_1, 0, t_n) = \lim_{h \to 0} \frac{t_n}{h} \sqrt{\frac{U - V}{2}} = \frac{t_n h_1}{\sqrt{h_1^2 - \xi^2}}.
\]

(76)

I discuss the connection between offset continuation and DMO in the next section.

OFFSET CONTINUATION AND DMO

Dip moveout represents a particular case of offset continuation for the output offset equal to zero. In this section, I consider the DMO case separately in order to compare the solutions of equation (16) with the Fourier-domain DMO operators, which have been the standard for DMO processing since Hale’s outstanding work (Hale 1983, 1984).

Equation (64) transforms to the time-wavenumber domain with the help of integral tables:

\[
\tilde{P}(t_n, h, k) = H(t_n) \left( \tilde{P}_0(t_n, h, k) + t_n \tilde{P}_1(t_n, h, k) \right),
\]

(77)

where

\[
\begin{align*}
\tilde{P}_0 &= \frac{\partial}{\partial t_n} \int_{(h_1/h) t_n}^{t_n} \tilde{P}_1^{(0)}(|t_1|, k) \int_0^\infty \frac{1}{t_1^2} \left( \sqrt{\frac{2}{t_1^2} \left( \frac{h^2}{t_1^2} - \frac{h_1^2}{t_1^2} \right) \left( t_n^2 - t_1^2 \right)} \right) dt_1, \\
\tilde{P}_1 &= \int_{(h_1/h) t_n}^{t_n} h_1 \tilde{P}_1^{(1)}(|t_1|, k) \int_0^\infty \frac{1}{t_1^2} \left( \sqrt{\frac{2}{t_1^2} \left( \frac{h^2}{t_1^2} - \frac{h_1^2}{t_1^2} \right) \left( t_n^2 - t_1^2 \right)} \right) dt_1, \\
\tilde{P}_1^{(j)}(t_1, k) &= \int P_1^{(j)}(t_1, y_1) \exp(-i k y_1) dy_1 \ (j = 0, 1), \\
\tilde{P}(t_n, h, k) &= \int P(t_n, h, y) \exp(-i k y) dy \ (j = 0, 1).
\end{align*}
\]

(78)  (79)  (80)  (81)
Setting the output offset to zero, we obtain the following DMO-like integral operators in the $t$–$k$ domain:

$$\tilde{P}(t_0, 0, k) = H(t_0) \left( \tilde{P}_0(t_0, k) + t_0 \tilde{P}_1(t_0, k) \right) ,$$  \hspace{1cm} (82)

where

$$\tilde{P}_0(t_0, k) = -\frac{\partial}{\partial t_0} \int_{t_0}^{\infty} \tilde{P}_0^{(0)}(|t_1|, k) J_0 \left( \frac{k h_1}{t_1} \sqrt{t_1^2 - t_0^2} \right) dt_1 ,$$  \hspace{1cm} (83)

$$\tilde{P}_1(t_0, k) = -\int_{t_0}^{\infty} h_1 \tilde{P}_1^{(1)}(|t_1|, k) J_0 \left( \frac{k h_1}{t_1} \sqrt{t_1^2 - t_0^2} \right) dt_1 \frac{dt_1}{t_1^2} ,$$  \hspace{1cm} (84)

the wavenumber $k$ corresponds to the midpoint axis $y$, and $J_0$ is the zeroth-order Bessel function. The Fourier transform of (83) and (84) with respect to the time variable $t_0$ reduces to known integrals (Gradshtein and Ryzhik, 1994) and creates explicit DMO-type operators in the frequency-wavenumber domain, as follows:

$$\tilde{\tilde{P}}_0(\omega_0, k) = i \int_{-\infty}^{\infty} \tilde{\tilde{P}}_0^{(0)}(|t_1|, k) \frac{\sin (\omega_0 |t_1| A)}{A} \frac{dt_1}{t_1^2} ,$$  \hspace{1cm} (85)

$$\tilde{\tilde{P}}_1(\omega_0, k) = i \int_{-\infty}^{\infty} h_1 \tilde{\tilde{P}}_1^{(1)}(|t_1|, k) \frac{\sin (\omega_0 |t_1| A)}{A} \frac{dt_1}{t_1^2} ,$$  \hspace{1cm} (86)

where

$$A = \sqrt{1 + \frac{(k h_1)^2}{(\omega_0 t_1)^2}} ,$$  \hspace{1cm} (87)

$$\tilde{\tilde{P}}_j(\omega_0, k) = \int \tilde{P}_j(t_0, k) \exp(i\omega_0 t_0) dt_0 .$$  \hspace{1cm} (88)

It is interesting to note that the first term of the continuation to zero offset (85) coincides exactly with the imaginary part of Hale’s DMO operator (Hale, 1984). However, unlike Hale’s, operator (82) is causal, which means that its impulse response does not continue to negative times. The non-causality of Hale’s DMO and related issues are discussed in more detail by Stovas and Fomel (1996).

Though Hale’s DMO is known to provide correct reconstruction of the geometry of zero-offset reflections, it does not account properly for the amplitude changes (Black et al., 1993). The preceding section of this paper shows that the additional contribution to the amplitude is contained in the second term of the OC operator (64), which transforms to the second term in the DMO operator (82). Note that this term vanishes at the input offset equal to zero, which represents the case of the inverse DMO operator.

Considering the inverse DMO operator as the continuation from zero offset to a non-zero offset, we can obtain its representation in the $t$-$k$ domain from equations (77–79) as

$$\tilde{P}(t_n, h, k) = H(t_n) \frac{\partial}{\partial t_n} \int_{0}^{t_n} \tilde{P}_0(|t_0|, k) J_0 \left( \frac{k h}{t_n} \sqrt{t_n^2 - t_0^2} \right) dt_0 ,$$  \hspace{1cm} (89)
Fourier transforming equation \( (89) \) with respect to the time variable \( t_0 \) according to equation \( (88) \), we get the Fourier-domain version of the “amplitude-preserving” inverse DMO:

\[
\tilde{P}(t_n, h, k) = \frac{H(t_n)}{2\pi} \frac{\partial}{\partial t_n} \int_{-\infty}^{\infty} \tilde{P}_0(\omega_0, k) \frac{\sin(\omega_0 |t_n| A)}{\omega_0 A} d\omega_0 , \tag{90}
\]

\[
A = \sqrt{1 + \frac{(kh)^2}{(\omega_0 t_n)^2}} . \tag{91}
\]

Comparing operator \( (90) \) with Ronen’s version of inverse DMO \( \text{(Ronen 1987)} \), one can see that if Hale’s DMO is denoted by \( D_{t_0} H \), then Ronen’s inverse DMO is \( H^T D_{-t_0} \), while the amplitude-preserving inverse \( (90) \) is \( D_{t_n} H^T \). Here \( D_t \) is the derivative operator \( \left( \frac{\partial}{\partial t} \right) \), and \( H^T \) stands for the adjoint operator defined by the dot-product test

\[
(\text{Hm, d}) = (m, H^T d), \tag{92}
\]

where the parentheses denote the dot product:

\[
(m_1, m_2) = \int \int m_1(t_n, y) m_2(t_n, y) dt_n dy .
\]

In high-frequency asymptotics, the difference between the amplitudes of the two inverses is simply the Jacobian term \( \frac{dt_0}{dt_n} \), asymptotically equal to \( \frac{t_0}{t_n} \). This difference corresponds exactly to the difference between Black’s definition of amplitude preservation \( \text{(Black et al. 1993)} \) and the definition used in Born DMO \( \text{(Bleistein 1990; Liner 1991)} \), as discussed above. While operator \( (90) \) preserves amplitudes in the Born DMO sense, Ronen’s inverse satisfies Black’s amplitude preservation criteria. This means Ronen’s operator implies that the “geometric spreading” correction (multiplication by time) has been performed on the data prior to DMO.

To construct a one-term DMO operator, thus avoiding the estimation of the offset derivative in \( (72) \), let us consider the problem of inverting the inverse DMO operator \( (90) \). One of the possible approaches to this problem is the least-squares iterative inversion, as proposed by \( \text{Ronen 1987} \). This requires constructing the adjoint operator, which is Hale’s DMO (or its analog) in the case of Ronen’s method. The iterative least-squares approach can account for irregularities in the data geometry \( \text{(Ronen et al. 1991; Ronen 1994)} \) and boundary effects, but it is computationally expensive because of the multiple application of the operators. An alternative approach is the asymptotic inversion, which can be viewed as a special case of preconditioning the adjoint operator \( \text{(Liner and Cohen 1988; Chemingui and Biondi 1996)} \). The goal of the asymptotic inverse is to reconstruct the geometry and the amplitudes of the reflection events in the high-frequency asymptotic limit.

According to Beylkin’s theory of asymptotic inversion, also known as the generalized Radon transform \( \text{(Beylkin 1985)} \), two operators of the form

\[
D(\omega) = \int X(t, \omega) M(t) \exp[i\omega\phi(t, \omega)] dt \tag{93}
\]
and
\[
\tilde{M}(t) = \int Y(t, \omega) D(\omega) \exp [-i \omega \phi(t, \omega)] \, d\omega
\]  
(94)

constitute a pair of asymptotically inverse operators (\(\tilde{M}(t)\) matching \(M(t)\) in the high-frequency asymptotics) if
\[
X(t, \omega) Y(t, \omega) = \frac{Z(t, \omega)}{2 \pi},
\]  
(95)

where \(Z\) is the “Beylkin determinant”
\[
Z(t, \omega) = \left| \frac{\partial \omega}{\partial \tilde{\omega}} \right| \quad \text{for} \quad \tilde{\omega} = \omega \frac{\partial \phi(t, \omega)}{\partial t}.
\]  
(96)

With respect to the high-frequency asymptotic representation, we can recast (90) in the equivalent form by moving the time derivative under the integral sign:
\[
\tilde{P}(t_n, k) \approx H(t_n) \frac{2 \pi}{\Re} \left[ \int_{-\infty}^{\infty} A^{-2} \tilde{P}_0(\omega_0, k) \exp \left( -i \omega_0 |t_n| A \right) d\omega_0 \right]
\]  
(97)

Now the asymptotic inverse of (97) is evaluated by means of Beylkin’s method (93)-(94), which leads to an amplitude-preserving one-term DMO operator of the form
\[
\tilde{P}_0(\omega_0, k) = \Im \left[ \int_{-\infty}^{\infty} B \tilde{P}_1^{(0)}(|t_1|, k) \exp \left( i \omega_0 |t_1| A \right) dt_1 \right],
\]  
(98)

where
\[
B = A^2 \frac{\partial}{\partial \omega_0} \left( \omega_0 \frac{\partial (t_n A)}{\partial t_n} \right) = A^{-1} \left( 2 A^2 - 1 \right).
\]  
(99)

The amplitude factor (99) corresponds exactly to that of Born DMO (Bleistein, 1990) in full accordance with the conclusions of the asymptotic analysis of the offset-continuation amplitudes. An analogous result can be obtained with the different definition of amplitude preservation proposed by Black et al. (1993). In the time-and-space domain, the operator asymptotically analogous to (A-7) is found by applying either the stationary phase technique (Liner, 1990; Black et al., 1993) or Goldin’s method of discontinuities (Goldin, 1988, 1990), which is the time-and-space analog of Beylkin’s asymptotic inverse theory (Stovas and Fomel, 1996). The time-and-space asymptotic DMO operator takes the form
\[
P_0(t_0, y) = D^{1/2} \int_{\xi_0}^{\infty} w_0(\xi; h_1, t_0) P_1^{(0)}(\theta^{(\cdot)}(\xi; h_1, 0, t_0), y_1 - \xi) \, d\xi,
\]  
(100)

where the weighting function \(w_0\) is defined as
\[
w_0(\xi; h_1, t_0) = \sqrt{\frac{t_0}{2 \pi}} \frac{h_1 (h_1^2 + \xi^2)}{(h_1^2 - \xi^2)^2}.
\]  
(101)
OFFSET CONTINUATION IN THE LOG-STRETCH DOMAIN

The log-stretch transform, proposed by Bolondi et al. (1982) and further developed by many other researchers, is a useful tool in DMO and OC processing. Applying a log-stretch transform of the form

$$\sigma = \ln \left| \frac{t_n}{t_*} \right| ,$$

(102)

where $t_*$ is an arbitrarily chosen time constant, eliminates the time dependence of the coefficients in equation (16) and therefore makes this equation invariant to time shifts. After the double Fourier transform with respect to the midpoint coordinate $y$ and to the transformed (log-stretched) time coordinate $\sigma$, the partial differential equation (16) takes the form of an ordinary differential equation,

$$h \left( \frac{d^2 \hat{P}}{dh^2} + k^2 \hat{P} \right) = i\Omega \frac{d\hat{P}}{dh} ,$$

(103)

where

$$\hat{P}(h) = \int \int P(t_n = t_* \exp(\sigma), h, y) \exp(i\Omega \sigma - iky) d\sigma dy .$$

(104)

Equation (103) has the known general solution, expressed in terms of cylinder functions of complex order $\lambda = \frac{1+i\Omega}{2}$ (Watson, 1952)

$$\hat{P}(h) = C_1(\lambda) (kh)^\lambda J_{-\lambda}(kh) + C_2(\lambda) (kh)^\lambda J_\lambda(kh) ,$$

(105)

where $J_{-\lambda}$ and $J_\lambda$ are Bessel functions, and $C_1$ and $C_2$ stand for some arbitrary functions of $\lambda$ that do not depend on $k$ and $h$.

In the general case of offset continuation, $C_1$ and $C_2$ are constrained by the two initial conditions (62) and (63). In the special case of continuation from zero offset, we can neglect the second term in (105) as vanishing at the zero offset. The remaining term defines the following operator of inverse DMO in the $\Omega, k$ domain:

$$\hat{P}(h) = \hat{P}(0) Z_\lambda(kh) ,$$

(106)

where $Z_\lambda$ is the analytic function

$$Z_\lambda(x) = \frac{\Gamma(1 - \lambda)}{\Gamma(n + 1 - \lambda)} \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \left( \frac{x^2}{4} \right)^n ,$$

(107)

$\Gamma$ is the gamma function and $\mathbf{0}F_1$ is the confluent hypergeometric limit function (Petkovsek et al., 1996).
The DMO operator now can be derived as the inversion of operator (106), which is a simple multiplication by $1/Z_\lambda(\lambda)$. Therefore, offset continuation becomes a multiplication by $Z_\lambda(\lambda kh_2)/Z_\lambda(\lambda kh_1)$ (the cascade of two operators). This fact demonstrates an important advantage of moving to the log-stretch domain: both offset continuation and DMO are simple filter multiplications in the Fourier domain of the log-stretched time coordinate.

In order to compare operator (106) with the known versions of log-stretch DMO, we need to derive its asymptotic representation for high frequency $\Omega$. The required asymptotic expression follows directly from the definition of function $Z_\lambda$ in equation (107) and the known asymptotic representation for a Bessel function of high order (Watson, 1952):

$$J_\lambda(\lambda z) \xrightarrow{\lambda \to \infty} \frac{(\lambda z)^\lambda \exp(\lambda \sqrt{1-z^2})}{e^\lambda \Gamma(\lambda+1)(1-z^2)^{1/4} \{1+\sqrt{1-z^2}\} \sqrt{1-z^2}}.$$  \hspace{1cm} (108)

Substituting approximation (108) into (107) and considering the high-frequency limit of the resultant expression yields

$$Z_\lambda(\lambda kh) \approx \left\{1 + \sqrt{1 - \left(\frac{kh}{2}\right)^2}\right\}^\lambda \exp \left(\lambda \left[1 - \sqrt{1 - \left(\frac{kh}{2}\right)^2}\right]\right) \left(1 - \left(\frac{kh}{2}\right)^2\right)^{1/4} \approx F(\epsilon) e^{i\Omega \psi(\epsilon)},$$ \hspace{1cm} (109)

where $\epsilon$ denotes the ratio $\frac{2kh}{\Omega}$,  

$$F(\epsilon) = \sqrt{\frac{1+\sqrt{1+\epsilon^2}}{2\sqrt{1+\epsilon^2}}} \exp\left(\frac{1-\sqrt{1+\epsilon^2}}{2}\right),$$ \hspace{1cm} (110)

and

$$\psi(\epsilon) = \frac{1}{2} \left(1 - \sqrt{1 + \epsilon^2} + \ln\left(\frac{1+\sqrt{1+\epsilon^2}}{2}\right)\right).$$ \hspace{1cm} (111)

The asymptotic representation (109) is valid for high frequency $\Omega$ and $|\epsilon| \leq 1$. The phase function $\psi$ defined in (111) coincides precisely with the analogous term in Liner’s exact log DMO (Liner, 1990), which provides the correct geometric properties of DMO. Similar expressions for the log-stretch phase factor $\psi$ were derived in different ways by Zhou et al. (1996) and Canning and Gardner (1996). However, the amplitude term $F(\epsilon)$ differs from the previously published ones because of the difference in the amplitude preservation properties.

A number of approximate log DMO operators have been proposed in the literature. As shown by Liner (1990), all of them but exact log DMO distort the geometry of reflection effects at large offsets. The distortion is caused by the implied approximations of the true phase function $\psi$. Bolondi’s OC operator (Bolondi
et al., 1982) implies \( \psi(\epsilon) \approx -\frac{\epsilon^2}{8} \), Notfors’ DMO (Notfors and Godfrey, 1987) implies \( \psi(\epsilon) \approx 1 - \sqrt{1 + (\epsilon/2)^2} \), and the “full DMO” (Bale and Jakubowicz, 1987) has \( \psi(\epsilon) \approx \frac{1}{2} \ln \left[ 1 - (\epsilon/2)^2 \right] \). All these approximations are valid for small \( \epsilon \) (small offsets or small reflector dips) and have errors of the order of \( \epsilon^4 \) (Figure 6). The range of validity of Bolondi’s operator is defined in equation (22).

Figure 6: Phase functions of the log DMO operators. Solid line: exact log DMO; dashed line: Bolondi’s OC; dashed-dotted line: Bale’s full DMO; dotted line: Notfors’ DMO.

In practice, seismic data are often irregularly sampled in space but regularly sampled in time. This makes it attractive to apply offset continuation and DMO operators in the \( \{\Omega, y\} \) domain, where the frequency \( \Omega \) corresponds to the log-stretched time and \( y \) is the midpoint coordinate. Performing the inverse Fourier transform on the spatial frequency transforms the inverse DMO operator (106) to the \( \{\Omega, y\} \) domain, where the filter multiplication becomes a convolutional operator:

\[
\hat{P}(\Omega, h, y) = \frac{\hat{F}(\Omega)}{\sqrt{2\pi}} \int_{|\xi|<h} \frac{h}{h^2 - \xi^2} \hat{P}_0(\Omega, y - \xi) \exp \left( -i\Omega \ln \left[ 1 - \left( \frac{\xi^2}{h^2} \right) \right] \right) d\xi. \tag{112}
\]

Here \( \hat{F}(\Omega) \) is a high-pass frequency filter:

\[
\hat{F}(\Omega) = \frac{\Gamma(1/2 - i\Omega/2)}{\sqrt{1/2 \Gamma(-i\Omega/2)}}. \tag{113}
\]

At high frequencies \( \hat{F}(\Omega) \) is approximately equal to \( (-i\Omega)^{1/2} \), which corresponds to the half-derivative operator \( \left( \frac{\partial}{\partial \sigma} \right)^{1/2} \), which, in turn, is equal to the \( \left( t_n \frac{\partial}{\partial t_n} \right)^{1/2} \) term of the asymptotic OC operator (69). The difference between the exact filter \( \hat{F} \) and its approximation by the half-order derivative operator is shown in Figure 7. This difference is a measure of the validity of asymptotic OC operators.

Inverting operator (112), we can obtain the DMO operator in the \( \{\Omega, y\} \) domain.

**DISCUSSION**

The differential model for offset continuation is based on several assumptions. It is important to fully realize them in order to understand the practical limitations of this model.
Figure 7: Amplitude (left) and phase (right) of the time filter in the log-stretch domain. The solid line is for the exact filter; the dashed line for its approximation by the half-order derivative filter. The horizontal axis corresponds to the dimensionless log-stretch frequency $\Omega$.

- The *constant velocity* assumption is essential for theoretical derivations. In practice, this limitation is not too critical, because the operators act locally. DMO and offset continuation algorithms based on the constant-velocity assumptions are widely used in practice ([Hale](1995)).

- The *single-mode* assumption does not include multiple reflections in the model. If multiple events (with different apparent velocities) are present in the data, they might require extending the model. Convolving two (or more) differential offset continuation operators, corresponding to different velocities, we can obtain a higher-order differential operator for predicting multiple events.

- The *continuous AVO* assumption implies that the reflectivity variation with offset is continuous and can be neglected in a local neighborhood of a particular offset. While the offset continuation model correctly predicts the geometric spreading effects in the reflected wave amplitudes, it does not account for the variation of the reflection coefficient with offset.

- The *2.5-D* assumption was implicit in the derivation of the offset continuation equation. According to this assumption, the reflector does not change in the cross-line direction, and we can always consider the reflection plane in two dimensions.
CONCLUSIONS

I have introduced a partial differential equation (16) and proved that the process described by it provides for a kinematically and dynamically equivalent offset continuation transform. Kinematic equivalence means that in constant velocity media the reflection traveltimes are transformed to their true locations on different offsets. Dynamic equivalence means that, in the OC process, the geometric spreading term in the amplitudes of reflected waves transforms in accordance with the laws of geometric seismics, while the angle-dependent reflection coefficient stays the same.

The offset continuation equation can be applied directly to design OC operators of the finite-difference type. To construct integral OC operators, an initial value problem is solved for the offset continuation equation (16). For the special cases of continuation to zero offset (DMO) and continuation from zero offset (inverse DMO), the OC operators are related to the known forms of DMO operators: Hale’s Fourier DMO, Born DMO, and Liner’s “exact log DMO.” The discovery of these relations sheds additional light on the problem of amplitude preservation in DMO.

ACKNOWLEDGMENTS

Sergey Goldin drew my attention to the role of curvature dependence in reflected wave amplitudes (Goldin and Fomel, 1995). I also acknowledge the support of Alexey Stovas, who introduced me to the world of DMO and collaborated with me on developing the asymptotic theory of offset continuation (Stovas and Fomel, 1996). Norman Bleistein collaborated with me on confirming the amplitude validity of offset continuation (Fomel et al., 1996; Fomel and Bleistein, 2001). Shuki Ronen and Richard Bale pointed out the importance of the log-stretch domain in DMO and related problems. I appreciate the encouragement of Jon Claerbout and Fabio Rocca regarding the analysis of the offset-continuation partial differential equation.

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

REFERENCES


Goldin, S., 1990, A geometric approach to seismic processing: the method of discon-
Goldin, S. V., 1988, Transformation and recovery of discontinuities in problems of
tomographic type: Institute of Geology and Geophysics.
——, 1994, Superposition and continuation of tranformations used in seismic migra-
tion: Russian Geology and Geophysics, 35, 131–145.
Russian Geology and Geophysics, 36, 103–115.
Gradshtein, I. S., and I. M. Ryzhik, 1994, Table of integrals, series, and products:
Haddon, R. A. W., and P. W. Buchen, 1981, Use of Kirchhoff’s formula for body
Edge, 20, 210–213.
Liner, C., 1990, General theory and comparative anatomy of dip moveout: Geo-
physics, 55, 595–607.
189.
Liner, C. L., and J. K. Cohen, 1988, An amplitude-preserving inverse of hale’s DmO:
Notfors, C. D., and R. J. Godfrey, 1987, Dip moveout in the frequency-wavenumber
Ronen, S., 1994, Handling irregular geometry: Equalized DmO and beyond: 64th
Ronen, S., V. Sorin, and R. Bale, 1991, Spatial dealiasing of 3-D seismic reflection
Stolt, R. H., 1978, Migration by Fourier transform: Geophysics, 43, 23–48. (Discuss-
ion and reply in GEO-60-5-1583).
Stovas, A. M., and S. B. Fomel, 1996, Kinematically equivalent integral DMO oper-
ators: Russian Geology and Geophysics, 37, 102–113.
Tenenbaum, M., and H. Pollard, 1985, Ordinary differential equations : an elemen-
tary textbook for students of mathematics, engineering, and the sciences: Dover
Publications.
University Press.
Zhou, B., I. M. Mason, and S. A. Greenhalgh, 1996, An accurate formulation of
log-stretch dip moveout in the frequency-wavenumber domain: Geophysics, 61,
815–820.
APPENDIX A
SECOND-ORDER REFLECTION TRAVELTIME DERIVATIVES

This appendix contains a derivation of equations connecting second-order partial derivatives of the reflection traveltime with the geometric properties of the reflector in a constant velocity medium. These equations are used in the main text of this paper to describe the amplitude behavior of offset continuation. Let $\tau(s, r)$ be the reflection traveltime from the source $s$ to the receiver $r$. Consider a formal equality

$$\tau(s, r) = \tau_1 (s, x(s, r)) + \tau_2 (x(s, r), r),$$  \hfill (A-1)

where $x$ is the reflection point parameter, $\tau_1$ corresponds to the incident ray, and $\tau_2$ corresponds to the reflected ray. Differentiating (A-1) with respect to $s$ and $r$ yields

$$\frac{\partial \tau}{\partial s} = \frac{\partial \tau_1}{\partial s} + \frac{\partial \tau}{\partial x} \frac{\partial x}{\partial s},$$ \hfill (A-2)

$$\frac{\partial \tau}{\partial r} = \frac{\partial \tau_2}{\partial r} + \frac{\partial \tau}{\partial x} \frac{\partial x}{\partial r}.$$ \hfill (A-3)

According to Fermat’s principle, the two-point reflection ray path must correspond to the traveltime stationary point. Therefore

$$\frac{\partial \tau}{\partial x} \equiv 0$$ \hfill (A-4)

for any $s$ and $r$. Taking into account (A-4) while differentiating (A-2) and (A-3), we get

$$\frac{\partial^2 \tau}{\partial s^2} = \frac{\partial^2 \tau_1}{\partial s^2} + B_1 \frac{\partial x}{\partial s},$$ \hfill (A-5)

$$\frac{\partial^2 \tau}{\partial r^2} = \frac{\partial^2 \tau_2}{\partial r^2} + B_2 \frac{\partial x}{\partial r},$$ \hfill (A-6)

$$\frac{\partial^2 \tau}{\partial s \partial r} = B_1 \frac{\partial x}{\partial r} = B_2 \frac{\partial x}{\partial s},$$ \hfill (A-7)

where

$$B_1 = \frac{\partial^2 \tau_1}{\partial s \partial x} ; B_2 = \frac{\partial^2 \tau_2}{\partial r \partial x}.$$  

Differentiating equation (A-4) gives us the additional pair of equations

$$C \frac{\partial x}{\partial s} + B_1 = 0,$$ \hfill (A-8)

$$C \frac{\partial x}{\partial r} + B_2 = 0,$$ \hfill (A-9)

where

$$C = \frac{\partial^2 \tau}{\partial x^2} = \frac{\partial^2 \tau_1}{\partial x^2} + \frac{\partial^2 \tau_2}{\partial x^2}.$$
Solving the system (A-8) - (A-9) for $\frac{\partial x}{\partial s}$ and $\frac{\partial x}{\partial r}$ and substituting the result into (A-5) - (A-7) produces the following set of expressions:

\begin{align*}
\frac{\partial^2 \tau}{\partial s^2} &= \frac{\partial^2 \tau_1}{\partial s^2} - C^{-1} B_1^2 ; \\
\frac{\partial^2 \tau}{\partial r^2} &= \frac{\partial^2 \tau_2}{\partial r^2} - C^{-1} B_2^2 ; \\
\frac{\partial^2 \tau}{\partial s \partial r} &= -C^{-1} B_1 B_2 .
\end{align*}

(A-10)

(A-11)

(A-12)

In the case of a constant velocity medium, expressions (A-10) to (A-12) can be applied directly to the explicit equation for the two-point eikonal

\[ \tau_1(y, x) = \tau_2(x, y) = \sqrt{(x-y)^2 + z^2(x)} / v . \]

(A-13)

Differentiating (A-13) and taking into account the trigonometric relationships for the incident and reflected rays (Figure II), one can evaluate all the quantities in (A-10) to (A-12) explicitly. After some heavy algebra, the resultant expressions for the traveltime derivatives take the form

\begin{align*}
\frac{\partial \tau}{\partial s} &= \frac{\tau_1}{v} = \sin \alpha_1 \\
\frac{\partial \tau_1}{\partial x} &= \frac{\sin \gamma}{v \cos \alpha} \\
\frac{\partial \tau}{\partial r} &= \frac{\tau_2}{v} = \sin \alpha_2 \\
\frac{\partial \tau_2}{\partial x} &= -\frac{\sin \gamma}{v \cos \alpha} ;
\end{align*}

(A-14)

(A-15)

\[ B_1 = \frac{\partial^2 \tau_1}{\partial s \partial x} = \frac{\cos \alpha_1}{v D \cos \alpha} \left( -1 - \frac{\sin \gamma}{\cos \alpha} \sin \alpha_1 \right) ; \]  

(A-16)

\[ B_2 = \frac{\partial^2 \tau_2}{\partial r \partial x} = \frac{\cos \alpha_2}{v D \cos \alpha} \left( -1 + \frac{\sin \gamma}{\cos \alpha} \sin \alpha_2 \right) ; \]

(A-17)

\[ B_1 B_2 = \frac{\cos^6 \gamma}{v^2 D^2 a^4} ; B_1 + B_2 = -2 \frac{\cos^3 \gamma}{v D a^2} (2 a^2 - 1) ; \]

(A-18)

\[ \frac{\partial^2 \tau_1}{\partial x^2} = \frac{\cos^2 \gamma + D K}{v D \cos^3 \alpha} \cos \alpha_1 ; \frac{\partial^2 \tau_2}{\partial x^2} = \frac{\cos^2 \gamma + D K}{v D \cos^3 \alpha} \cos \alpha_2 ; \]

(A-19)

\[ C = \frac{\partial^2 \tau_1}{\partial x^2} + \frac{\partial^2 \tau_2}{\partial x^2} = 2 \cos \gamma \frac{\cos^2 \gamma + D K}{v D \cos^3 \alpha} . \]

(A-20)

Here $D$ is the length of the normal (central) ray, $\alpha$ is its dip angle ($\alpha = \alpha_1 + \alpha_2$, $\tan \alpha = z'(x)$), $\gamma$ is the reflection angle ($\gamma = \frac{\alpha_2 - \alpha_1}{2}$), $K$ is the reflector curvature at the reflection point ($K = z''(x) \cos \alpha$), and $a$ is the dimensionless function of $\alpha$ and $\gamma$ defined in (17).

The equations derived in this appendix were used to obtain the equation

\[ \tau_n \left( \frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2} \right) = 4 \left( \tau \frac{\partial^2 \tau}{\partial s \partial r} + \frac{\cos^2 \gamma}{v^2} \right) = 4 \frac{\cos^2 \gamma}{v^2} \left( \frac{\sin^2 \alpha + DK}{\cos^2 \gamma + DK} \right) , \]

(A-21)

which coincides with (50) in the main text.
APPENDIX B

THE KINEMATICS OF OFFSET CONTINUATION

This Appendix presents an alternative method to derive equation (70), which describes the summation path of the integral offset continuation operator. The method is based on the following considerations.

The summation path of an integral (stacking) operator coincides with the phase function of the impulse response of the inverse operator. Impulse response is by definition the operator reaction to an impulse in the input data. For the case of offset continuation, the input is a reflection common-offset gather. From the physical point of view, an impulse in this type of data corresponds to the special focusing reflector (elliptical isochrone) at the depth. Therefore, reflection from this reflector at a different constant offset corresponds to the impulse response of the OC operator. In other words, we can view offset continuation as the result of cascading prestack common-offset migration, which produces the elliptic surface, and common-offset modeling (inverse migration) for different offsets. This approach resembles that of Deregowski and Rocca (1981). It was also applied to a more general case of azimuth moveout (AMO) by Fomel and Biondi (1995) and fully generalized by Bleistein and Jaramillo (2000). The geometric approach implies that in order to find the summation pass of the OC operator, one should solve the kinematic problem of reflection from an elliptic reflector whose focuses are in the shot and receiver locations of the output seismic gather.

In order to solve this problem, let us consider an elliptic surface of the general form

\[ h(x) = \sqrt{d^2 - \beta (x - x')^2}, \tag{B-1} \]

where \(0 < \beta < 1\). In a constant velocity medium, the reflection ray path for a given source-receiver pair on the surface is controlled by the position of the reflection point \(x\). Fermat’s principle provides a required constraint for finding this position. According to Fermat’s principle, the reflection ray path corresponds to a stationary value of the travel-time. Therefore, in the neighborhood of this path,

\[ \frac{\partial \tau(s, r, x)}{\partial x} = 0, \tag{B-2} \]

where \(s\) and \(r\) stand for the source and receiver locations on the surface, and \(\tau\) is the reflection traveltime

\[ \tau(s, r, x) = \frac{\sqrt{h^2(x) + (s - x)^2}}{v} + \frac{\sqrt{h^2(x) + (r - x)^2}}{v}. \tag{B-3} \]

Substituting equations (B-3) and (B-1) into (B-2) leads to a quadratic algebraic equation on the reflection point parameter \(x\). This equation has the explicit solution

\[ x(s, r) = x' + \frac{\xi^2 + H^2 - h^2 + \text{sign}(h^2 - H^2) \sqrt{(\xi^2 - H^2 - h^2)^2 - 4 H^2 h^2}}{2 \xi (1 - \beta)}, \tag{B-4} \]
where $h = (r - s)/2$, $\xi = y - x'$, $y = (s + r)/2$, and $H^2 = d^2 \left( \frac{1}{\beta} - 1 \right)$. Replacing $x$ in equation (B-3) with its expression (B-4) solves the kinematic part of the problem, producing the explicit traveltime expression

$$
\tau(s, r) = \begin{cases} 
\frac{1}{v} \sqrt{\frac{4h^2 - \beta (f + g)^2}{1 - \beta}} & \text{for } h^2 > H^2 \\
\frac{1}{v} \sqrt{\frac{4h^2 + \beta (F + G)^2}{1 - \beta}} & \text{for } h^2 < H^2
\end{cases}
$$

(B-5)

where

$$f = \sqrt{(r - x')^2 - H^2}, \quad g = \sqrt{(s - x')^2 - H^2},
F = \sqrt{H^2 - (r - x')^2}, \quad G = \sqrt{H^2 - (s - x')^2}.
$$

The two branches of equation (B-5) correspond to the difference in the geometry of the reflected rays in two different situations. When a source-and-receiver pair is inside the focuses of the elliptic reflector, the midpoint $y$ and the reflection point $x$ are on the same side of the ellipse with respect to its small semi-axis. They are on different sides in the opposite case (Figure B-1).

![Figure B-1](image_url)

Reflections from an ellipse. The three pairs of reflected rays correspond to a common midpoint (at 0.1) and different offsets. The focuses of the ellipse are at 1 and -1.

If we apply the NMO correction, equation (B-5) is transformed to

$$
\tau_n(s, r) = \begin{cases} 
\frac{1}{v} \sqrt{\frac{\beta}{1 - \beta} \sqrt{4h^2 - (f + g)^2}} & \text{for } h^2 > H^2 \\
\frac{1}{v} \sqrt{\frac{\beta}{1 - \beta} \sqrt{4h^2 + (F + G)^2}} & \text{for } h^2 < H^2
\end{cases}
$$

(B-6)

Then, recalling the relationships between the parameters of the focusing ellipse $r$, $x'$ and $\beta$ and the parameters of the output seismic gather (Deregowski and Rocca, 1981)

$$
r = \frac{vt_n}{2}, \quad x' = y, \quad \beta = \frac{t_n^2}{t_n^2 + 4h^2 + \frac{4h^2}{v^2}}, \quad H = h,
$$

(B-7)
and substituting expressions (B-7) into equation (B-6) yields the expression

\[ t_1(s_1, r_1; s, r, t_n) = \begin{cases} 
\frac{t_n}{2h} \sqrt{4h_1^2 - (f + g)^2} & \text{for } h_1^2 > h^2 \\
\frac{t_2}{2h} \sqrt{4h_1^2 + (F + G)^2} & \text{for } h_1^2 < h^2 
\end{cases} \]  
(B-8)

where

\[ f = \sqrt{(r_1 - r)(r_1 - s)}, \quad g = \sqrt{(s_1 - r)(s_1 - s)}, \]
\[ F = \sqrt{(r - r_1)(r_1 - s)}, \quad G = \sqrt{(s_1 - r)(s - s_1)}. \]

It is easy to verify algebraically the mathematical equivalence of equation (26) and equation (70) in the main text. The kinematic approach described in this appendix applies equally well to different acquisition configurations of the input and output data. The source-receiver parameterization used in (26) is the actual definition for the summation path of the integral shot continuation operator (Bagaini and Spagnolini, 1993, 1996). A family of these summation curves is shown in Figure B-2.

Figure B-2: .

Summation paths of the integral shot continuation. The output source is at -0.5 km. The output receiver is at 0.5 km. The indexes of the curves correspond to the input source location.
Amplitude preservation for offset continuation: Confirmation for Kirchhoff data

Sergey Fomel* and Norman Bleistein†

ABSTRACT

Offset continuation (OC) is the operator that transforms common-offset seismic reflection data from one offset to another. Earlier papers by the first author presented a partial differential equation in midpoint and offset to achieve this transformation. The equation was derived from the kinematics of the continuation process with no reference to amplitudes. We present here a proof that the solution of the OC partial differential equation does propagate amplitude properly at all offsets, at least to the same order of accuracy as the Kirchhoff approximation. That is, the OC equation provides a solution with the correct traveltime and correct leading-order amplitude. “Correct amplitude” in this case means that the transformed amplitude exhibits the right geometrical spreading and reflection-surface-curvature effects for the new offset. The reflection coefficient of the original offset is preserved in this transformation. This result is more general than the earlier results in that it does not rely on the two-and-one-half dimensional assumption.

INTRODUCTION

Offset continuation (OC) is the operator that transforms common-offset seismic reflection data to data with a different offset. Following the classic results of Deregowski and Rocca (1981), Bolondi et al. (1982, 1984) described OC as a continuous process of gradual change of the offset by means of a partial differential equation. Because it is based on the small-offset small-dip approximation, Bolondi’s equation failed at large offsets or steep reflector dips. Nevertheless, the OC concept inspired a flood of research on dip moveout (DMO) correction (Hale, 1991). Since one can view DMO as a particular case of OC (continuation to zero offset), the offset continuation theory can serve as a natural basis for DMO theory. Its immediate application is in interpolating data undersampled in the offset dimension.

Fomel (1994, 2003) introduced a revised version of the OC differential equation and proved that it provides the correct kinematics of the continued wavefield for any offset.

*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50A-1148, Berkeley, CA 94720, USA
†Center for Wave Phenomena, Colorado School of Mines, Golden, CO 80401, USA
*e-mail: sergey@sep.stanford.edu, biondo@sep.stanford.edu
and reflector dip under the assumption of constant effective velocity. The equation is interpreted as an "image wave equation" by Hubral et al. (1996). Studying the laws of amplitude transformation shows that in 2.5-D media the amplitudes of continued seismic gathers transform according to the rules of geometric seismics, except for the reflection coefficient, which remains unchanged (Goldin and Fomel, 1995; Fomel, 2003). The solution of the boundary problem on the OC equation for the DMO case (Fomel, 2003) coincides in high-frequency asymptotics with the amplitude-preserving DMO, also known as Born DMO (Liner, 1991; Bleistein, 1990). However, for the purposes of verifying that the amplitude is correct for any offset, this derivation is incomplete.

In this paper, we perform a direct test on the amplitude properties of the OC equation. We describe the input common-offset data by the Kirchhoff modeling integral, which represents the high-frequency approximation of a reflected (scattered) wavefield, recorded at the surface at nonzero offset (Bleistein, 1984). For reflected waves, the Kirchhoff approximation is accurate up to the two orders in the high-frequency series (the ray series) for the differential operator applied to the solution, with the first order describing the phase function alone and the second order describing the amplitude. We prove that both orders of accuracy are satisfied when the offset continuation equation is applied to Kirchhoff data. Thus, this differential equation is the "right" equation to two orders, producing the correct amplitude as well as the correct phase for offset continuation. That is, the geometric spreading effects and curvature effects of the reflected data are properly transformed. The angularly dependent reflection coefficient of the original offset is preserved.

This proof relates the OC equation with "wave-equation" processing. It also provides additional confirmation of the fact that the true-amplitude OC and DMO operators (Black et al., 1993; Goldin and Fomel, 1995; Santos et al., 1997; Tygel et al., 1998) do not depend on the reflector curvature and can properly transform reflections from arbitrarily shaped reflectors (Goldin, 1990; Tygel et al., 1996; Bleistein et al., 2001). The latter result was specifically a 2.5-D result, whereas the result of this paper does not depend on the 2.5-D assumption. That is, the result presented here remains valid when the reflector has out-of-plane variation.

Our method of proof is indirect. We first write the Kirchhoff representation for the reflected wave in a form that can be easily matched to the solution of the OC differential equation. We then present the analogues of the eikonal and transport equations for the OC equation and show that the amplitude and phase of the Kirchhoff representation satisfy those two equations.

THE KIRCHHOFF MODELING APPROXIMATION

In this section, we introduce the Kirchhoff approximate integral representation of the upward propagating response to a single reflector, with separated source and receiver points. We then show how the amplitude of this integrand is related to the zero-
offset amplitude at the source receiver point on the ray, making equal angles at the
scattering point with the rays from the separated source and receiver. The Kirch-
off integral representation (Haddon and Buchen, 1981; Bleistein, 1984) describes
the wavefield scattered from a single reflector. This representation is applicable in
situations where the high-frequency assumption is valid (the wavelength is smaller
than the characteristic dimensions of the model) and corresponds in accuracy to the
WKBJ approximation for reflected waves, including phase shifts through buried foci.

The general form of the Kirchhoff modeling integral is

$$U_S(r, s, \omega) = \int_\Sigma R(x; r, s) \frac{\partial}{\partial n} [U_I(s, x, \omega) G(x, r, \omega)] d\Sigma,$$

where $s = (s, 0, 0)$ and $r = (r, 0, 0)$ stand for the source and the receiver location
vectors at the surface of observation; $x$ denotes a point on the reflector surface $\Sigma$;
$R$ is the reflection coefficient at $\Sigma$; $n$ is the upward normal to the reflector at the
point $x$; and $U_I$ and $G$ are the incident wavefield and Green’s function, respectively
represented by their WKBJ approximation,

$$U_I(s, x, \omega) = F(\omega) A_s(s, x) e^{i\omega \tau_s(s, x)},$$

$$G(x, r, \omega) = A_r(x, r) e^{i\omega \tau_r(x, r)}.$$  

In this equation, $\tau_s(s, x)$ and $A_s(s, x)$ are the traveltime and the amplitude of
the wave propagating from $s$ to $x$; $\tau_r(x, r)$ and $A_r(x, r)$ are the corresponding quantities
for the wave propagating from $x$ to $r$; and $F(\omega)$ is the spectrum of the input signal,
assumed to be the transform of a bandlimited impulsive source. In the time domain,
the Kirchhoff modeling integral transforms to

$$u_S(r, s, t) = \int_\Sigma R(x; r, s) \frac{\partial}{\partial n} [A_s(s, x) A_r(x, r) f(t - \tau_s(s, x) - \tau_r(x, r))] dx,$$

with $f$ denoting the inverse temporal transform of $F$. The reflection traveltime $\tau_{sr}$
corresponds physically to the diffraction from a point diffractor located at the point
$x$ on the surface $\Sigma$, and the amplitudes $A_s$ and $A_r$ are point diffractor amplitudes, as well.

The main goal of this paper is to test the compliance of representation (4) with the
offset continuation differential equation. The OC equation contains the derivatives of
the wavefield with respect to the parameters of observation $(s, r, \text{and } t)$. According to
the rules of classic calculus, these derivatives can be taken under the sign of integration
in formula (4). Furthermore, since we do not assume that the true-amplitude OC
operator affects the reflection coefficient $R$, the offset-dependence of this coefficient
is outside the scope of consideration. Therefore, the only term to be considered as
a trial solution to the OC equation is the kernel of the Kirchhoff integral, which is
contained in the square brackets in equations (1) and (4) and has the form

$$k(s, r, x, t) = A_{sr}(s, r, x) f(t - \tau_{sr}(s, r, x)),$$
where

\[ \tau_{sr}(s, r, x) = \tau_s(s, x) + \tau_r(x, r) , \]

(6)

\[ A_{sr}(s, r, x) = A_s(s, x) A_r(x, r) . \]

(7)

In a 3-D medium with a constant velocity \( v \), the traveltimes and amplitudes have the simple explicit expressions

\[ \tau_s(s, x) = \frac{\rho_s(s, x)}{v} , \quad \tau_r(x, r) = \frac{\rho_r(x, r)}{v} , \]

(8)

\[ A_s(s, x) = \frac{1}{4\pi \rho_s(s, x)} , \quad A_r(x, r) = \frac{1}{4\pi \rho_r(x, r)} , \]

(9)

where \( \rho_s \) and \( \rho_r \) are the lengths of the incident and reflected rays, respectively (Figure 1). If the reflector surface \( \Sigma \) is explicitly defined by some function \( z = z(x) \), then

\[ \rho_s(s, x) = \sqrt{(x - s)^2 + z^2(x)} , \quad \rho_r(x, r) = \sqrt{(r - x)^2 + z^2(x)} . \]

(10)

Figure 1: Geometry of diffraction in a constant velocity medium: view in the reflection plane.

We then introduce a particular zero-offset amplitude, namely the amplitude along the zero offset ray that bisects the angle between the incident and reflected ray in this plane, as shown in Figure 1. We denote the square of this amplitude as \( A_0 \). That is,

\[ A_0 = \frac{1}{(4\pi \rho_0)^2} . \]

(11)

\( A_0 \) is the amplitude factor that appears in the Kirchhoff integral set up for a zero-offset reflection along the ray \( \rho_0 \). It is, thus, the desired output factor inside the Kirchhoff
integral after DMO. As follows from formulas (7) and (9), the amplitude transformation in DMO (continuation to zero offset) is characterized by the dimensionless ratio

$$\frac{A_{sr}}{A_0} = \frac{\rho_0^2}{\rho_s \rho_r},$$

(12)

where $\rho_0$ is the length of the zero-offset ray (Figure 1).

As follows from the simple trigonometry of the triangles, formed by the incident and reflected rays (the law of cosines),

$$\sqrt{\rho_s^2 + \rho_0^2 - 2 \rho_s \rho_0 \cos \gamma} + \sqrt{\rho_r^2 + \rho_0^2 - 2 \rho_r \rho_0 \cos \gamma} =$$

$$= \sqrt{\rho_s^2 + \rho_r^2 - 2 \rho_s \rho_r \cos 2\gamma},$$

(13)

where $\gamma$ is the reflection angle, as shown in the figure. After straightforward algebraic transformations of equation (13), we arrive at the explicit relationship between the ray lengths:

$$\frac{(\rho_s + \rho_r) \rho_0}{2 \rho_s \rho_r} = \cos \gamma.$$

(14)

Substituting (14) into (12) yields

$$\frac{A_{sr}}{A_0} = \frac{\tau_0}{\tau_{sr}} \cos \gamma,$$

(15)

where $\tau_0$ is the zero-offset two-way travelt ime ($\tau_0 = 2 \rho_0/v$).

What we have done is rewrite the finite-offset amplitude in the Kirchhoff integral in terms of a particular zero-offset amplitude. That zero-offset amplitude would arise as the geometric spreading effect if there were a reflector whose dip was such that the finite-offset pair would be specular at the scattering point. Of course, the zero-offset ray would also be specular in this case.

**THE OFFSET CONTINUATION EQUATION**

In this section, we introduce the offset continuation partial differential equation. We then develop its WKBJ, or ray theoretic, solution for phase and leading-order amplitude. We explain how we verify that the traveltime and amplitude of the integrand of the Kirchhoff representation (4) satisfy the “eikonal” and “transport” equations of the OC partial differential equation. To do so, we make use of relationship (15), derived from the Kirchhoff integral.

The offset continuation differential equation derived in earlier papers (Fomel, 1994, 2003) is

$$h \left( \frac{\partial^2 P}{\partial y^2} - \frac{\partial^2 P}{\partial h^2} \right) = t_n \frac{\partial^2 P}{\partial t_n \partial h},$$

(16)

†To our knowledge, the first derivation of the revised offset continuation equation was accomplished by Joseph Higginbotham of Texaco in 1989. Unfortunately, Higginbotham’s derivation never appeared in the open literature.
In this equation, $h$ is the half-offset ($h = l/2$), $y$ is the midpoint ($y = (s + r)/2$) [hence, $y = (r + s)/2$], and $t_n$ is the NMO-corrected traveltme

$$t_n = \sqrt{t^2 - \frac{l^2}{v^2}}. \tag{17}$$

Equation (16) describes the process of seismogram transformation in the time-midpoint-offset domain. One can obtain the high-frequency asymptotics of its solution by standard methods, as follows. We introduce a trial asymptotic solution of the form

$$P(y, h, t_n) = A_n(y, h) f(t_n - \tau_n(y, h)). \tag{18}$$

It is important to remember the assumption that $f$ is a “rapidly varying function,” for example, a bandlimited delta function. We substitute this solution into equation (16) and collect the terms in order of derivatives of $f$. This is the direct counterpart of collecting terms in powers of frequency when applying WKBJ in the frequency domain. From the leading asymptotic order (the second derivative of the function $f$), we obtain the eikonal equation describing the kinematics of the OC transformation:

$$h \left[ \left( \frac{\partial \tau_n}{\partial y} \right)^2 - \left( \frac{\partial \tau_n}{\partial h} \right)^2 \right] = -\tau_n \frac{\partial \tau_n}{\partial h}. \tag{19}$$

In this equation, we have replaced a multiplier of $t_n$ by $\tau_n$ on the right side of the equation. This is consistent with our assumption that $f$ is a bandlimited delta function or some equivalent impulsive source. Analogously, collecting the terms containing the first derivative of $f$ leads to the transport equation describing the transformation of the amplitudes:

$$\left( \tau_n - 2h \frac{\partial \tau_n}{\partial h} \right) \frac{\partial A_n}{\partial h} + 2h \frac{\partial \tau_n}{\partial y} \frac{\partial A_n}{\partial y} + hA_n \left( \frac{\partial^2 \tau_n}{\partial y^2} - \frac{\partial^2 \tau_n}{\partial h^2} \right) = 0. \tag{20}$$

We then rewrite the eikonal equation (19) in the time-source-receiver coordinate system, as follows:

$$\left( \tau^2_{sr} + \frac{l^2}{v^2} \right) \left( \frac{\partial \tau_{sr}}{\partial r} - \frac{\partial \tau_{sr}}{\partial s} \right) = 2l \tau_{sr} \left( \frac{1}{v^2} - \frac{\partial \tau_{sr}}{\partial r} \frac{\partial \tau_{sr}}{\partial s} \right), \tag{21}$$

which makes it easy (using Mathematica) to verify that the explicit expression for the phase of the Kirchhoff integral kernel (6) satisfies the eikonal equation for any scattering point $\mathbf{x} = (x_1, x_2, z)$. Here, $\tau_{sr}$ is related to $\tau_n$ as $t$ is related to $t_n$ in equation (17). 

\(^1\)Note that the scattering point $\mathbf{x}$ plays the role of a set of parameters in the partial differential equation for $\tau_{sr}$. To pass from a two-dimensional in-plane traveltme to a three-dimensional traveltme, one need only replace $z^2$ with $x_2^2 + z^2$. The role of $x = x_1$ remains unchanged in the solution.
The general solution of the amplitude equation (20) has the form (Fomel, 2003)

\[ A_n = A_0 \frac{\tau_0 \cos \gamma}{\tau_n} \left( \frac{1 + \rho_0 K}{\cos^2 \gamma + \rho_0 K} \right)^{1/2}, \]  

(22)

where \( K \) is the reflector curvature at the reflection point. The kernel \( 5 \) of the Kirchhoff integral \( 4 \) corresponds to the reflection from a point diffractor: the integral realizes the superposition of Huygens secondary source contributions. We can obtain the solution of the amplitude equation for this case by formally setting the curvature \( K \) to infinity (setting the radius of curvature to zero). The infinite curvature transforms formula (22) to the relationship

\[ \frac{A_n}{A_0} = \frac{\tau_0}{\tau_n} \cos \gamma. \]  

(23)

Again, we exploit the assumption that the signal \( f \) has the form of the delta function. In this case, the amplitudes before and after the NMO correction are connected according to the known properties of the delta function, as follows:

\[ A_{sr} \delta (t - \tau_{sr}(s, r, x)) = \left. \frac{\partial t_n}{\partial t} \right|_{t=\tau_{sr}} A_{sr} \delta (t_n - \tau_n(s, r, x)) = A_n \delta (t_n - \tau_n(s, r, x)). \]  

(24)

with

\[ A_n = \frac{\tau_{sr}}{\tau_n} A_{sr}. \]  

(25)

Combining equations (25) and (23) yields

\[ \frac{A_{sr}}{A_0} = \frac{\tau_0}{\tau_{sr}} \cos \gamma, \]  

(26)

which coincides exactly with the previously found formula (15). As with the solution of the eikonal equation, we pass from an in-plane solution in two dimensions to a solution for a scattering point in three dimensions by replacing \( z^2 \) with \( x_2^2 + z^2 \).

Although the presented equations pertain to the case of offset continuation that starts from \( h = 0 \), i.e., inverse DMO, this is sufficient, since every other continuation can be obtained as a chain of DMO and inverse DMO.

Thus, it is apparent that the OC differential equation (16) relates to the Kirchhoff representation of reflection data. We see that the amplitude and phase of the Kirchhoff representation for arbitrary offset is the point diffractor WKBJ solution of the offset continuation differential equation. Hence, the Kirchhoff approximation is a solution of the OC differential equation when we hold the reflection coefficient constant. This means that the solution of the OC differential equation has all the features of amplitude preservation, as does the Kirchhoff representation, including geometrical spreading, curvature effects, and phase shift effects. Furthermore, in the Kirchhoff representation and the solution of the OC partial differential equation by WKBJ, we have not used the 2.5-D assumption. Therefore the preservation of amplitude is not restricted to cylindrical surfaces as it is in the true-amplitude proof for DMO (Bleistein et al., 2001). This is what we sought to confirm.
DISCUSSION

We have proved that the offset continuation equation correctly transforms common-offset seismic data modeled by the Kirchhoff integral approximation. The kinematic and dynamic equivalence of the OC equation has been proved previously by different methods (Fomel, 2003). However, connecting this equation with Kirchhoff modeling opens new insights into the theoretical basis of DMO and offset continuation:

1. The Kirchhoff integral can serve as a link between the wave-equation theory, conventionally used in seismic data processing, and the kinematically derived OC equation. Though the analysis in this paper follows the constant-velocity model, this link can be extended in principle to handle the case of a variable background velocity.

2. The OC equation operates on the kernel of the Kirchhoff integral, which is independent of the local dip and curvature of the reflector. This proves that the true-amplitude OC and DMO operators can properly transform reflections from curved reflectors. Moreover, this result does not imply any special orientation of the reflector curvature matrix. Therefore, it does not require a commonly made 2.5-D assumption (Bleistein et al., 2001). Implicitly, this fact proves the amplitude preservation property of the three-dimensional azimuth moveout (AMO) operator (Biondi et al., 1998), based on cascading the true-amplitude DMO and inverse DMO operators.

ACKNOWLEDGMENTS

We thank the Center for Wave Phenomena Consortium Project at the Colorado School of Mines and the Stanford Exploration Project at Stanford University. We appreciate the encouragement of Fabio Rocca regarding the analysis of the offset continuation partial differential equation. Finally, we thank the anonymous reviewer for the helpful suggestions.

REFERENCES

Effective AMO implementation in the log-stretch, frequency-wavenumber domain

Ioan Vlad and Biondo Biondi

INTRODUCTION

Azimuth moveout (AMO), introduced by Biondi et al. (1998), is used as part of the styling goal (in conjunction with a derivative as a roughener) in Biondi and Vlad (2001). This paper describes the implementation of AMO for the above-stated purpose, with a historical background, proof, and discussion of pitfalls and practical steps.

THE AZIMUTH MOVEOUT

AMO is conceived as a cascade of forward and reverse dip moveout (DMO) operators. Thus, the accuracy and speed of the DMO operator used is highly important. Computing the DMO in the frequency domain is accurate and simple, but computationally expensive because the DMO operator is temporally nonstationary. The technique of logarithmic time-stretching, introduced by Bolondi et al. (1982) increases the computational efficiency because the DMO operator is stationary in the log-stretch domain, and Fast Fourier Transforms can be used instead of slow Discrete Fourier Transforms. Gardner (1991), Black et al. (1993) and Zhou et al. (1996) derived equivalent and accurate log-stretch, frequency-wavenumber DMO operators. The implementation of the AMO presented in this paper is based on the derivation and algorithm in Zhou et al. (1996).

THE LOG-STRETCH, FREQUENCY-WAVENUMBER AMO IN 3D

Starting from the parametric DMO relations of Black et al. (1993), Zhou et al. (1996) derives an expression for a DMO applicable on 2D NMO-ed data. In order to extend the expression to 3D, we only have to replace the product $kh$ between the wavenumber and half offset with the dot product of the same quantities, which are vectors in the case of 3D data. In order to perform AMO from the offset $\vec{h}_1$ to the offset $\vec{h}_2$, we need to cascade one forward DMO from offset $\vec{h}_1$ to zero offset with a reverse DMO from zero offset to offset $\vec{h}_2$. Thus, applying log-stretch, frequency-wavenumber AMO on

*e-mail: nick@sep.stanford.edu, biondo@sep.stanford.edu
a 3D cube of data \( P(t, m_x, m_y) \) in order to obtain \( P(t, m_x, m_y) \) involves the following sequence of operations:

1. Apply log-stretch along the time axis on the \( P(t, m_x, m_y) \) cube, with the formula:

\[
\tau = \ln \left( \frac{t}{t_c} \right),
\]

where \( t_c \) is the minimum cutoff time introduced to avoid taking the logarithm of zero. All samples from times smaller than \( t_c \) are simply left untouched, the rest of the procedure will be applied to the cube \( P(t > t_c, m_x, m_y) \).

2. 3D forward FFT of the \( P(\tau, m_x, m_y) \) cube. The 3D forward Fourier Transform is defined as follows:

\[
P(\Omega, k_x, k_y) = \int \int \int P(\tau, m_x, m_y) e^{i(\Omega \tau - k_x m_x - k_y m_y)} d\tau dm_x dm_y
\]

It can be seen that the sign of the transform along the \( \tau \)-axis is opposite to that over the midpoint axes.

3. For each element of the cube, perform the AMO shift:

\[
P(\Omega, k_x, k_y) = e^{i(\Phi_1 - \Phi_2)} P(\Omega, k_x, k_y), \text{ where}
\]

\[
\Phi_j = \begin{cases} 
0, & \text{for } \vec{k} \cdot \vec{h} = 0 \\
\vec{k} \cdot \vec{h}, & \text{for } \Omega = 0 \\
\frac{\Omega}{2} \left\{ \sqrt{1 + \left( \frac{2 \vec{k} \cdot \vec{h}}{\Omega} \right)^2} - 1 - \ln \left[ \sqrt{1 + \left( \frac{2 \vec{k} \cdot \vec{h}}{\Omega} \right)^2 + 1} \right] \right\}, & \text{otherwise}
\end{cases},
\]

where \( \vec{k} \cdot \vec{h} = k_x h_x + k_y h_y \)

and \( j \) can take the values 1 or 2. The frequency domain variables must have incorporated in their value a \( 2\pi \) constant (they are defined according to equation (2)).

4. Do reverse 3D FFT in order to obtain the \( P(\tau, m_x, m_y) \) cube.

5. Do reverse log stretch along the time axis and affix to the top of the cube the slices from times smaller than \( t_c \). The final result is a \( P(t, m_x, m_y) \) cube.

Figure 1 shows the impulse response of the above described AMO.
STRETCHING AND ALIASING

For the purpose of this discussion we define stretching of a single-dimension space as any transformation from one space to another that has the following property: at least an arbitrarily chosen sequence of two consecutive, equal in length, intervals in the input space is transformed into a sequence of two consecutive, not equal in length, intervals in the output space. Stretching an x-space to a y-space will be denoted as

\[ y = f(x) \]  

(6)

Two obvious examples of stretching are

\[ NMO : \ y = \sqrt{x^2 + \alpha}, \quad \text{and} \]

\[ \log - \text{stretch} : \ y = \log \left( \frac{x}{\alpha} \right), \]

where \( \alpha \) is a positive real number whose value does not matter for the purpose of this discussion. As it can be seen in Fig. 2 if we keep the same sampling rate (\( \Delta y = \Delta x \)), aliasing can occur when doing the reverse transformation, from x to y. In order to avoid aliasing, we need to compute \( \Delta y_{\max} \), the largest acceptable sampling rate in the y domain. This can sometimes lead to a larger number of samples in the y domain, and thus to larger computational expense. This can be limited to some extent if the signal in the x-space has been bandpassed, as is often the case with seismic data, with
the largest frequency present in the data ($f_{\text{max}}$) smaller than the Nyquist frequency given by the sampling rate ($f_{\text{Ny}}$). Thus, we can replace in our calculations $\Delta x$ with

$$\Delta x_{\text{max}} = \frac{1}{2f_{\text{max}}}$$

which will result in a $\Delta y_{\text{max}}$ larger than that computed using $\Delta x$, the sampling rate in the $x$ space.

\[\text{Figure 2: Illustration of how aliasing can occur while stretching: if the same sampling rate is used for the } y\text{-space (lower plot) as for the } x\text{-space (upper plot), serious aliasing will occur when transforming back to } x\text{-space. This will not happen if the sampling rate in the } y\text{-space is smaller than or equal to } \Delta y_{\text{max}}.}\]

In order to compute $\Delta y_{\text{max}}$, we will consider two points in the $x$ space, as seen in Fig. 2 such as

$$x_b = x_a + \Delta x_{\text{max}}$$

(7)

and $y_a$ and $y_b$, the images of $x_a$ and $x_b$ in the $y$ space. Thus,

$$\Delta y = y_b - y_a = f(x_a + \Delta x_{\text{max}}) - f(x_a)$$
Figure 3: Illustration of how aliasing can occur while stretching: if the same sampling rate is used for the $y$-space (lower plot) as for the $x$-space (upper plot), serious aliasing will occur when transforming back to $x$-space. This will not happen if the sampling rate in the $y$-space is smaller than or equal to $\Delta y_{\text{max}}$. 
The largest sampling rate in the $y$-space that will not result in aliasing is $\Delta y_{\text{max}}$, the minimum possible value of $\Delta y$. Suppose there is a value $x_m$ that minimizes $\Delta y$. Then,

$$\Delta y_{\text{max}} = \left[ f(x + \Delta x_{\text{max}}) - f(x) \right]_{x_m}$$

In particular, in the case of log-stretch, given by equation (1), if $t_m$ plays the role of $x_m$ from the equation above, then

$$\Delta \tau_{\text{max}} = \left[ \log \left( \frac{t + \Delta t_{\text{max}}}{t_c} \right) - \log \left( \frac{t}{t_c} \right) \right]_{t_m} = \log \left( 1 + \frac{\Delta t_{\text{max}}}{t_m} \right)$$

$\tau_{\text{max}}$ will be minimum when $t_m$ is as large as possible, thus minimizing the expression under the logarithm. How large can $t_m$ get? Since the length of the seismic trace is limited to a value $t_{\text{max}}$,

$$t_m = t_{\text{max}} - \Delta t_{\text{max}}$$

because $t_m$ is the equivalent of $x_a$ from eq. (7) and Fig. 2. Thus, we get

$$\Delta \tau_{\text{max}} = \log \left( \frac{t_{\text{max}}}{t_{\text{max}} - \Delta t_{\text{max}}} \right)$$

**F-K FILTERING**

As it can be seen in Fig. 4, the impulse response of the AMO computed in the log-stretch, frequency-wavenumber domain has some artifacts: high amplitude, large saddle corners. Low temporal frequencies and high spatial slopes are also present. These artifacts can be eliminated easily using a f-k filter, which is described below.

Suppose we want to attenuate all spatial frequencies $k$ that are larger than a certain threshold $k_{\text{max}}$, where

$$k = \sqrt{k_x^2 + k_y^2} \quad \text{and} \quad k_{\text{max}} = \frac{2|\omega|}{v_{\text{min}}}$$

with $\omega$, $k_x$ and $k_y$ being the coordinates in the frequency-wavenumber domain (without logstretch), and $v$ being the minimum apparent velocity of the events that we want the filtered data cube to contain. Thus, the data cube will become:

$$P_{\text{filtered}}(\omega, k_x, k_y) = \begin{cases} P(\omega, k_x, k_y) & \text{if} \quad k \leq k_{\text{max}} \\ e^{-\varepsilon(k - k_{\text{max}})^2} P(\omega, k_x, k_y) & \text{if} \quad k > k_{\text{max}} \end{cases}$$

Too small an $\varepsilon$ will result in an abrupt transition in the f-k domain, and thus ringing artifacts in the t-x domain. An $\varepsilon$ which is too big will result in no visible filtering of the targeted artifacts. Moreover, $\varepsilon$ depends on the choice of units and the number of samples for the $m_x$ and $m_y$ axes: since the exponential needs to be dimensionless, we have

$$\varepsilon = \frac{\varepsilon_0}{dk_x dk_y}$$
where

\[ dk_x = \frac{1}{n_x d_x} \quad \text{and} \quad dk_y = \frac{1}{n_y d_y}. \]

Thus, the final expression of \( \varepsilon \) is

\[ \varepsilon = \varepsilon_0 n_x d_x n_y d_y, \quad \text{(12)} \]

where \( \varepsilon_0 \) is a value that is hand-picked only once, and embedded in the code. This way, we will not have to change anything at all in the code or in the parameters in order to set \( \varepsilon_0 \), no matter what the units of the data cube may be.

The result of the filtering can be seen in Fig. 5: the slices through the cube are taken at exactly the same locations as those in Fig. 4, but now the artefacts are gone.

**COST-CUTTING AVENUES**

The largest computational savings come from the use of FFTs for AMO, instead of slow Fourier integration necessary in the absence of log-stretch. Standard means of minimizing the CPU time and the amount of memory used to compute the AMO have also been employed. They include computing the AMO shift for only half of the
elements of the cube in the complex domain, since the Fourier transform $F$ of a real function is Hermitian:

$$F(s) = F^*(-s)$$

(where $s$ denotes the frequency domain variable and the star symbol denotes the complex conjugate). Another way of reducing computational expenses was through the use of RFFTW and FFTW type Fourier Transforms [Frigo and Johnson [1998]], adaptive to hardware architecture, and taking advantage of the property stated in (13). Also, the code was divided into subroutines in such a way that some quantities were not computed unnecessarily several times when AMO was applied to more than one cube of data. Finally, shared memory parallelization with the OpenMP standard was applied to all the computationally intensive do loops in the code.

**CONCLUSIONS**

Azimuth moveout can be successfully implemented in the log-stretch, frequency-wavenumber domain. It is accurate, fast, and furthermore it does not have any characteristics that can result in coding difficulties.
REFERENCES


Earthquake stacks at constant offset

Jon F. Claerbout

ABSTRACT
I show Shearer’s earthquake stacks over all source-receiver locations at constant offset and compare them to exploration seismic data. This electronic document simply reads the stacks and plots them.

INTRODUCTION
The figures in this paper show Professor Peter Shearer’s stacks over many earthquakes at constant epicentral distance (offset angle). Notice that the time scale is minutes and the offset is degrees of angle on the earth’s surface. Otherwise, there are remarkable similarities to conventional exploration seismic data. For details of how the stacks were made, see the references.

SHEARER’S EARTHQUAKE STACKS
Professor Shearer has found a way to choose the polarity for each seismogram so that the SH waves do not randomly cancel in the stack. Since SH waves have no conversions, the SH-wave stack in Figure 1 is easier to understand than the P-SV stack shown later. The first big event is direct S (shear wave). It comes tangent to ScS (reflection from the earth’s core) hitting grazing angle around 90 degrees at about 23 minutes of travel time. The first multiple reflection of ScS at grazing angle is at double the epicentral distance and double the travel time. It is easily seen at the opposite side of the earth near 180 degrees offset. Figure 1 also shows six bounces of direct S bouncing from the earth’s surface before the plot ends at one hour travel time. Incidentally, many fewer earthquakes are observed near 180 degrees than near 90 degrees for the simple geometrical reason that ten degrees surrounding the equator is a much bigger area than ten degrees surrounding the pole. Thus the quality of the stacks degrades rapidly toward the poles.

Waves analogous to familiar water-bottom pegleg multiples are readily seen tracking S about 4 minutes later than S. The data shows these two events, one being a reflection from a layer at 410 km and the other from 660 km. As with familiar water-bottom peglegs, each event is a superposition of two events, one with a short path near the source and the other with a short path near the receiver.
Figure 1: Transverse component. SH waves.
Of particular interest are waves identified by Professor Shearer that track SS, the first surface multiple of S, but arrive about 4 minutes earlier. Were it not for the confusing similarity between “shear” and “Shearer”, I would call these “Shearer” waves. They arrive earlier than the first multiple of direct S because instead of an earth-surface bounce at the middle of the path, they bounce a little sooner from the underside of the layers at 660 km and 410 km. The nice thing about Shearer’s waves is that they are uniquely associated with the middle of the path, rather than having two bounce locations, as do familiar first-order peglegs. We might like to search our files of exploration data to see if we can find analogous events.

In the stack in Figure 2, Professor Shearer has chosen the polarities of each seismogram so that P-SV waves stack in phase. Because of conversions between P and SV, there are a huge number of new events which I leave to you to pick out. Waves that propagate further than 180 degrees fold back (like discrete Fourier transforms). I cannot explain why such folding was not visible on the SH section.

Figure 3 is a plot of signal amplitudes. These were the first stacks that Professor Shearer made because they require no knowledge of polarity.

A challenging goal is to redo these stacks from the original data trying to boost the spectral bandwidth of the final result. This involves statics corrections and corrections for source depth.

ACKNOWLEDGMENT

I received this data from Peter Shearer and I received his permission to redistribute it to friends and colleagues provided I request you to reference its origin, should you have occasion to copy it. Examples of earlier versions of these stacks are found in the references. (The stacks here were done before February 1993.) Peter may be willing to supply newer and better stacks. His electronic mail address is shearer@mahi.ucsd.edu.

REFERENCES

Shearer, P. M., 1991a, Constraints on upper mantle discontinuities from observations of long period reflected and converted phases: J. Geophy. Res., 96, 18,147–18,182.
Figure 2: Vertical component showing P and SV waves.
Figure 3: Transverse and vertical component of signal magnitude. See if you believe in whole-earth anisotropy.
INTRODUCTION

Wouldn’t it be great if I could take signals of 10-30 Hz bandwidth from 100 different offsets and construct a zero-offset trace with 5-100 Hz bandwidth? This would not violate Shannon’s sampling theorem which theoretically allows us to have a transform from 100 signals of 20 Hz bandwidth to one signal at 2000 Hz bandwidth. The trouble is that simple NMO is not such a transformation. Never-the-less, if the different offsets really did give us any extra information, we should be able to put the information into extra bandwidth. Let us consider noise free synthetic data and see if we can come up with a model where this could happen.

FITTING FRAMEWORK

The operator of interest is the one that creates many offsets of seismic data from a zero-offset model space.

\[ \mathbf{z} \] is a white seismic trace (model) at zero-offset

\[ \mathbf{d}_j \] is a red seismic trace (data) at nonzero-offset \( x_j \)

\( \mathbf{L} \) is a seismic band pass filter

\( H_j \) sprays along hyperbola using a known, rough \( v(z) \)

\( \overline{H}_j \) sprays using a known, smooth \( \overline{v}(z) \)

The operator of interest is the one that transforms \( \mathbf{z} \) to all the data \( \mathbf{d}_j \) at all of the offsets \( x_j \).

\[ \mathbf{d}_j = \mathbf{LH}_j \mathbf{z} \quad (1) \]

Here is a trivial idea: Estimates \( \hat{\mathbf{z}}_j \) of \( \mathbf{z} \) from data \( \mathbf{d}_j \) at different offsets \( x_j \) have different spectral bands because of NMO stretch. Wide offsets create low frequency. Trouble is, these low frequencies add little spectral bandwidth. We want extra high frequencies too.

We know a simple two-step process where one offset can be obtained from another: First moveout for one offset. Then inverse moveout for the other offset. Whenever such offset continuation works, extra offsets cannot bring us extra information. Extra traces give only redundancy.

Inversion theory says if the transformation has no null space we should be able to solve for everything. Since in practice we cannot seem to obtain that extra bandwidth, it seems that the operator \( \mathbf{LH}_j \) has a large null space, about equal in size to the trace length times (the number of offsets minus one).
ROUGH VELOCITY(Z)

Taking velocity to be a rough (bumpy) function of depth, different offset traces might be fundamentally different thus providing different information (i.e. more information hence potentially more bandwidth). The bumpy velocity model seems artificial since it requires to be known a rough velocity as a function depth. Never-the-less, the idea could be helpful because we sometimes have well logs, or we might later learn how to bootstrap our velocity estimate from a smooth velocity to a rougher one.

Many people think about rough impedance. Here we consider a rough interval velocity.

ROUGH V(Z) MAKES TAU(T) MULTIVALUED

According to the Dix approximation, travel time \( t(\tau) \) is a unique function of vertical travel time \( \tau \) because

\[
\begin{align*}
    t^2 &= \tau^2 + x^2/v(\tau)^2
\end{align*}
\]

(2)

The reverse is not true, however, \( \tau(t) \) can be a multivalued function of \( t \), and is especially likely to be so where \( v(\tau) \) is a rough function of \( \tau \). When \( \tau(t) \) is a multivalued function of \( t \) the process of offset continuation breaks down. Then extra offsets are providing extra information. We don’t yet know if the extra information is a small amount or a large amount or whether that extra information is uniformly or locally distributed. Figure 1 shows an example.

Figure 1 shows two kinds of multivaluedness in the transformation. First is the familiar kind that arises whenever \( dv/dz > 0 \) where travel times of shallow waves cross those of deep waves. Let us place a line through the broad maxima in \( t(\tau) \) at about \( t = 2.5\tau \) for all \( x \). In a constant velocity earth, the ratio \( t/\tau = 2.5 \) corresponds to a propagation angle \( \cos \theta = \tau/t \) or about \( \theta = 66^\circ \). Thus, a wave with average angle greater than \( \theta = 66^\circ \) generally arrives at the same time and offset as another wave with an average angle less than \( \theta = 66^\circ \).

The second way of being multivalued is less familiar and hence more interesting, the roughness in the \( t(\tau) \) transformation. We see this roughness does give rise to multivaluedness. Disappointingly, the multivaluedness is not found everywhere but mainly along the \( \theta = 66^\circ \) trend. We have not yet answered how much extra information we can obtain from this. Clearly though, if multivaluedness is what makes different offsets give us different information, it is along this “mute-line” \( \theta = 66^\circ \) trend where we must look.

Let us find the high frequency. Where does an observable (low) frequency on the \( t \) axis map to a high frequency on the \( \tau \) axis? It happens where a long region on the \( t \) axis maps to a short region on the \( \tau \) axis, in other words, where the slope \( dt/d\tau \) is greatest. This is the opposite of usual NMO in the neighborhood of the diagonal asymptote in Figure 1 where \( dt/d\tau < 1 \). From the figure, we see the possibility for
Figure 1: Right shows $t(\tau)$ for many offsets.
frequency boosting does not arise from the roughness in velocity but just beneath the water bottom at any offset, i.e., at the greatest angles. Since $dt/d\tau$ is negative there, it gives a kind of upside-down image. To understand this image, think of head waves where the deepest layer is fastest and hence has the earliest arrival with slarrow layer arrivals coming later.

It is possible the Dix approximation is breaking down here, a concern that requires further study. Accurate reflection seismograms in this region are easy to make with the phase shift method. Getting correct head waves is more complicated.

**FURTHER STEPS**

Each offset $x_j$ allows us to make a different estimate of the earth model $z_j$. There are two possibilities:

i.e. $z_j \neq z_{j+1}$.

\[
\hat{z}_j = H_j' L'd_j
\]

(3)

\[
\hat{z}_j = \overline{H}_j' L'd_j
\]

(4)

We should plot $\hat{z}_j$ as a function of $j$. We should also plot $\hat{z}_j - \hat{z}_{j-1}$ as a function of $j$ and see if we can find any higher temporal frequencies.
INTRODUCTION

Elliptical anisotropy has found wide use as a simple approximation to transverse isotropy because of a unique symmetry property (an elliptical dispersion relation corresponds to an elliptical impulse response) and a simple relationship to standard geophysical techniques (hyperbolic moveout corresponds to elliptical wavefronts; NMO measures horizontal velocity, and time-to-depth conversion depends on vertical velocity). However, elliptical anisotropy is only useful as an approximation in certain restricted cases, such as when the underlying true anisotropy does not depart too far from ellipticity or the observed angular aperture is small. This limitation is fundamental, because there are only two parameters needed to define an ellipse: the horizontal and vertical velocities. (Sometimes the orientation of the principle axes is also included as a free parameter, but usually not.)

In a previous SEP report [Muir (1990)] showed how to extend the standard elliptical approximation to a so-called double-elliptic form. (The relation between the elastic constants of a TI medium and the coefficients of the corresponding double-elliptic approximation is developed in a companion paper, [Muir (1991)].) The aim of this new approximation is to preserve the useful properties of elliptical anisotropy while doubling the number of free parameters, thus allowing a much wider range of transversely isotropic media to be adequately fit. At first glance this goal seems unattainable: elliptical anisotropy is the most complex form of anisotropy possible with a simple analytical form in both the dispersion relation and impulse response domains. Muir’s approximation is useful because it nearly satisfies both incompatible goals at once: it has a simple relationship to NMO and true vertical and horizontal velocity, and to a good approximation it has the same simple analytical form in both domains of interest.

The purpose of this short note is to test by example how well the double-elliptic approximation comes to meeting these goals:

1. Simple relationships to NMO and true velocities on principle axes.
2. Simple analytical form for both the dispersion relation and impulse response.
3. Approximates general transversely isotropic media well.

The results indicate that the method should work well in practice.

REVIEW OF SYMMETRIC PROPERTIES OF ELLIPSES

Figure [I] shows a transversely isotropic medium and its vertical paraxial elliptic approximation represented four different ways. Note that the dashed elliptical-approximation curve plots as an ellipse in both the group-velocity and phase-slowness domains. The vertical and horizontal velocities of the elliptical impulse response define its principle
axes; the corresponding vertical and horizontal slownesses give the principle axes of the elliptical dispersion relation.

\[
\begin{array}{ccc}
\text{Velocity} & \text{Slowness} \\
\text{Group} & & 1/r \\
\text{Phase} & & 1/r
\end{array}
\]

Figure 1: Group velocity (= impulse response), group slowness, phase velocity, and phase slowness (= dispersion relation) plots for the qSV mode of Greenhorn Shale (thick solid line) and an elliptically anisotropic paraxial approximation to it (thin dashed line). The “R” and “1/r” symbols indicate the recurring transformations linking the four representations.

These vertical and horizontal velocities have simple geophysical interpretations. The vertical velocity of the paraxial elliptic approximation (equals the true vertical velocity) is what we need to do time-to-depth conversion. The horizontal velocity of the paraxial elliptic approximation (not the same thing as the true horizontal velocity) is what is required for NMO. (In a surface survey the sources and receivers are laid out along a horizontal line; if the vertical velocity were changed, it would effectively change the vertical scale of the survey, but the traveltime field recorded along the horizontal surface would remain unchanged. So it is horizontal velocity that matters. But it isn’t the true horizontal velocity, because we’re only considering NMO for near-vertical propagation. It is the horizontal velocity of the vertical paraxial elliptic approximation.)

THE DOUBLE ELLIPTIC APPROXIMATION...

in the impulse-response domain

Muir’s trick is to turn the problem on its side and also look at the horizontal paraxial approximating ellipse. This has all the useful properties of the vertical ellipse, with
Figure 2: Three different approximations (dashed curves) to the $q_{SV}$ impulse-response surface of Greenhorn Shale (bold curves). On the left is the standard vertical paraxial elliptic approximation. In the center is the horizontal paraxial elliptic approximation. On the right is Muir’s double-elliptic approximation.

Figure 3: Three different approximations (dashed curves) to the $q_{SV}$ dispersion-relation surface of Greenhorn Shale (bold curves). On the left is the standard vertical paraxial elliptic approximation. In the center is the horizontal paraxial elliptic approximation. On the right is Muir’s double-elliptic approximation; the fit is so close the dashed curve is hard to see.
the difference that it fits the true horizontal velocity and the vertical NMO velocity (not the same thing as the true vertical velocity). Figure 2 shows the vertical and horizontal paraxial approximating ellipses, along with Muir’s double-elliptic approximation. Muir’s approximation fits four parameters, the vertical and horizontal true and NMO velocities. Since it is single-valued, it cannot follow the qSV triplication, but fits well elsewhere.

in the dispersion-relation domain

There is no such problem in the dispersion-relation domain, since triplications are impossible there. Figure 3 shows how the three approximations shown in Figure 2 work in the dispersion-relation domain. Just like elliptical anisotropy, Muir’s four-parameter approximation is calculated in the dispersion-relation domain by simply using slownesses instead of velocities in the same approximation formula. Note in Figure 3 that the double-elliptic approximation in the dispersion-relation domain is nearly exact for this example.

but are they consistent?

Of course there is one technicality: the elliptical approximations in the previous two figures were perfectly consistent, but the double-elliptic approximations were not. Muir’s approximation did a much better job in the dispersion-relation domain than it did in the impulse-response domain. Figure 4 shows how the double-elliptic approximations fit in the two different domains compare in the impulse-response domain. Should we be concerned by this discrepancy? Probably not, because this is a rather extreme example. If there are no triplications the approximations as fit in the two domains are much more nearly consistent, as is demonstrated in Figure 5.

CONCLUSIONS

The double-elliptic approximation introduced by Muir (1990) can approximate the kinematics of TI media quite accurately, although it is forced to cut off triplications when fit in the impulse-response domain. In the dispersion-relation domain it remains accurate even for triplicating media.

Our analysis in this paper was limited to simple kinematics. Before pronouncing the double-elliptic approximation a success, we need to also demonstrate how it works when used as the basis for an imaging technique; i.e., how accurately does it model the dynamics of the wave equation? Karrenbach (1991) examines this question in a companion paper.
Figure 4: Two different double-elliptic approximations (dashed curves) fit to the qSV mode of Greenhorn Shale (solid curves). Left: the approximation is fit in the impulse-response domain, and so the dashed curve has a simple analytic form. Right: the approximation is fit in the dispersion-relation domain, and so is able to closely follow the triplication. This approximating curve can only be calculated parametrically, however, and so is less useful. elliptic2/vels/ compare

Figure 5: Two different double-elliptic approximations (dashed curves) corresponding to those in Figure 4 but this time fit to the qP mode of Greenhorn Shale (solid curves). (The size of the ‘*’ in the middle shows the relative scales.) The discrepancy is much less since there are no troublesome triplications. elliptic2/vels/ compare2
REFERENCES


Velocity continuation and the anatomy of residual prestack time migration

Sergey Fomel

ABSTRACT

Velocity continuation is an imaginary continuous process of seismic image transformation in the post-migration domain. It generalizes the concepts of residual and cascaded migrations. Understanding the laws of velocity continuation is crucially important for a successful application of time migration velocity analysis. These laws predict the changes in the geometry and intensity of reflection events on migrated images with the change of the migration velocity. In this paper, I derive kinematic and dynamic laws for the case of prestack residual migration from simple geometric principles. The main theoretical result is a decomposition of prestack velocity continuation into three different components corresponding to residual normal moveout, residual dip moveout, and residual zero-offset migration. I analyze the contribution and properties of each of the three components separately. This theory forms the basis for constructing efficient finite-difference and spectral algorithms for time migration velocity analysis.

INTRODUCTION

The conventional approach to seismic migration theory (Claerbout, 1985; Berkhout, 1985) employs the downward continuation concept. According to this concept, migration extrapolates upgoing reflected waves, recorded on the surface, to the place of their reflection to form an image of subsurface structures. Post-stack time migration possesses peculiar properties, which can lead to a different viewpoint on migration. One of the most interesting properties is an ability to decompose the time migration procedure into a cascade of two or more migrations with smaller migration velocities. This remarkable property is described by Rothman et al. (1985) as residual migration. Larner and Beasley (1987) generalized the method of residual migration to one of cascaded migration. Cascading finite-difference migrations overcomes the dip limitations of conventional finite-difference algorithms (Larner and Beasley, 1987); cascading Stolt-type f-k migrations expands their range of validity to the case of a vertically varying velocity (Beasley et al., 1988). Further theoretical generalization sets the number of migrations in a cascade to infinity, making each step in the velocity space infinitesimally small. This leads to a partial differential equation in the time-midpoint-velocity space, discovered by Claerbout (1986). Claerbout’s equation
describes the process of velocity continuation, which fills the velocity space in the same manner as a set of constant-velocity migrations. Slicing in the migration velocity space can serve as a method of velocity analysis for migration with nonconstant velocity (Shurtleff, 1984; Fowler, 1984, 1988; Mikulich and Hale, 1992).

The concept of velocity continuation was introduced in the earlier publications (Fomel, 1994, 1997; Hubral et al., 1996) and Schleicher et al. (1997) use the term image waves to describe a similar idea. Adler (2002) generalizes it to the case of variable background velocities under the name Kirchhoff image propagation. The importance of this concept lies in its ability to predict changes in the geometry and intensity of reflection events on seismic images with the change of migration velocity. While conventional approaches to migration velocity analysis methods take into account only vertical movement of reflectors (Deregowski, 1990; Liu and Bleistein, 1995), velocity continuation attempts to describe both vertical and lateral movements, thus providing for optimal focusing in velocity analysis applications (Fomel, 2001, 2003b).

In this paper, I describe the velocity continuation theory for the case of prestack time migration, connecting it with the theory of prestack residual migration (Al-Yahya and Fowler, 1986; Etgen, 1990; Stolt, 1996). By exploiting the mathematical theory of characteristics, a simplified kinematic derivation of the velocity continuation equation leads to a differential equation with correct dynamic properties. In practice, one can accomplish dynamic velocity continuation by integral, finite-difference, or spectral methods. The accompanying paper (Fomel, 2003b) introduces one of the possible numerical implementations and demonstrates its application on a field data example.

The paper is organized into two main sections. First, I derive the kinematics of velocity continuation from the first geometric principles. I identify three distinctive terms, corresponding to zero-offset residual migration, residual normal moveout, and residual dip moveout. Each term is analyzed separately to derive an analytical prediction for the changes in the geometry of travelt ime curves (reflection events on migrated images) with the change of migration velocity. Second, the dynamic behavior of seismic images is described with the help of partial differential equations and their solutions. Reconstruction of the dynamical counterparts for kinematic equations is not unique. However, I show that, with an appropriate selection of additional terms, the image waves corresponding to the velocity continuation process have the correct dynamic behavior. In particular, a special boundary value problem with the zero-offset velocity continuation equation produces the solution identical to the conventional Kirchhoff time migration.

**KINEMATICS OF VELOCITY CONTINUATION**

From the kinematic point of view, it is convenient to describe the reflector as a locally smooth surface $z = z(x)$, where $z$ is the depth, and $x$ is the point on the surface ($x$ is a two-dimensional vector in the 3-D problem). The image of the reflector obtained after
a common-offset prestack migration with a half-offset \( h \) and a constant velocity \( v \) is the surface \( z = z(x; h, v) \). Appendix A provides the derivations of the partial differential equation describing the image surface in the depth-midpoint-offset-velocity space. The purpose of this section is to discuss the laws of kinematic transformations implied by the velocity continuation equation. Later in this paper, I obtain dynamic analogs of the kinematic relationships in order to describe the continuation of migrated sections in the velocity space.

The kinematic equation for prestack velocity continuation, derived in Appendix A, takes the following form:

\[
\frac{\partial \tau}{\partial v} = v \tau \left( \frac{\partial \tau}{\partial x} \right)^2 + \frac{h^2}{v^3 \tau^2} \left( \frac{\partial \tau}{\partial x} \right)^2 \left( \frac{\partial \tau}{\partial h} \right)^2.
\]  

(1)

Here \( \tau \) denotes the one-way vertical traveltime \( \left( \tau = \frac{z}{v} \right) \). The right-hand side of equation (1) consists of three distinctive terms. Each has its own geophysical meaning. The first term is the only one remaining when the half-offset \( h \) equals zero. This term corresponds to the procedure of zero-offset residual migration. Setting the traveltime dip to zero eliminates the first and third terms, leaving the second, dip-independent one. One can associate the second term with the process of residual normal moveout. The third term is both dip- and offset-dependent. The process that it describes is residual dip moveout. It is convenient to analyze each of the three processes separately, evaluating their contributions to the cumulative process of prestack velocity continuation.

**Kinematics of Zero-Offset Velocity Continuation**

The kinematic equation for zero-offset velocity continuation is

\[
\frac{\partial \tau}{\partial v} = v \tau \left( \frac{\partial \tau}{\partial x} \right)^2.
\]  

(2)

The typical boundary-value problem associated with it is to find the traveltime surface \( \tau_2(x_2) \) for a constant velocity \( v_2 \), given the traveltime surface \( \tau_1(x_1) \) at some other velocity \( v_1 \). Both surfaces correspond to the reflector images obtained by time migrations with the specified velocities. When the migration velocity approaches zero, post-stack time migration approaches the identity operator. Therefore, the case of \( v_1 = 0 \) corresponds kinematically to the zero-offset (post-stack) migration, and the case of \( v_2 = 0 \) corresponds to the zero-offset modeling (demigration). The variable \( x \) in equation (2) describes both the surface midpoint coordinate and the subsurface image coordinate. One of them is continuously transformed into the other in the velocity continuation process.

The appropriate mathematical method of solving the kinematic problem posed above is the method of characteristics (Courant and Hilbert [1989]). The characteristics of equation (2) are the trajectories followed by individual points of the reflector
image in the velocity continuation process. These trajectories are called velocity rays (Fomel 1994; Liptow and Hubral 1995; Adler 2002). Velocity rays are defined by the system of ordinary differential equations derived from (2) according to the Hamilton-Jacobi theory:

\[
\frac{dx}{dv} = -2v\tau_x, \quad \frac{d\tau}{dv} = -\tau_v, \quad (3)
\]

\[
\frac{d\tau_x}{dv} = v\tau_x^3, \quad \frac{d\tau_v}{dv} = (\tau + v\tau_v)\tau_x^2, \quad (4)
\]

where \(\tau_x\) and \(\tau_v\) are the phase-space parameters. An additional constraint for \(\tau_x\) and \(\tau_v\) follows from equation (2), rewritten in the form

\[
\tau_v = v\tau_x^2. \quad (5)
\]

The general solution of the system of equations (3-4) takes the parametric form

\[
x(v) = A - Cv^2, \quad \tau^2(v) = B - C^2v^2, \quad (6)
\]

\[
\tau_x(v) = \frac{C}{\tau(v)}, \quad \tau_v(v) = \frac{C^2v}{\tau(v)}, \quad (7)
\]

where \(A\), \(B\), and \(C\) are constant along each individual velocity ray. These three constants are determined from the boundary conditions as

\[
A = x_1 + v_1^2\tau_1 \frac{\partial \tau_1}{\partial x_1} = x_0, \quad (8)
\]

\[
B = \tau_1^2 \left(1 + v_1^2 \left(\frac{\partial \tau_1}{\partial x_1}\right)^2\right) = \tau_0^2, \quad (9)
\]

\[
C = \tau_1 \frac{\partial \tau_1}{\partial x_1} = \tau_0 \frac{\partial \tau_0}{\partial x_0}, \quad (10)
\]

where \(\tau_0\) and \(x_0\) correspond to the zero velocity (unmigrated section), while \(\tau_1\) and \(x_1\) correspond to the velocity \(v_1\). The simple relationship between the midpoint derivative of the vertical traveltime and the local dip angle \(\alpha\) (appendix A),

\[
\frac{\partial \tau}{\partial x} = \tan \frac{\alpha}{v}, \quad (11)
\]

shows that equations (8) and (9) are precisely equivalent to the evident geometric relationships (Figure 1)

\[
x_1 + v_1\tau_1 \tan \alpha = x_0, \quad \frac{\tau_1}{\cos \alpha} = \tau_0. \quad (12)
\]

Equation (10) states that the points on a velocity ray correspond to a single reflection point, constrained by the values of \(\tau_1\), \(v_1\), and \(\alpha\). As follows from equations (6), the projection of a velocity ray to the time-midpoint plane has the parabolic shape.
Figure 1: Zero-offset reflection in a constant velocity medium (a scheme).

\[ x(\tau) = A + (\tau^2 - B)/C, \]
which has been noticed by Chun and Jacewitz (1981). On the depth-midpoint plane, the velocity rays have the circular shape \( z^2(x) = (A - x) B/C - (A - x)^2 \), described by Liptow and Hubral (1995) as "Thales circles."

For an example of kinematic continuation by velocity rays, let us consider the case of a point diffractor. If the diffractor location in the subsurface is the point \( x_d, z_d \), then the reflection traveltime at zero offset is defined from Pythagoras's theorem as the hyperbolic curve

\[ \tau_0(x_0) = \sqrt{z_d^2 + (x_0 - x_d)^2} \frac{v_d}{v}, \]

where \( v_d \) is half of the actual velocity. Applying equations (6) produces the following mathematical expressions for the velocity rays:

\[ x(v) = x_d \frac{v^2}{v_d^2} + x_0 \left( 1 - \frac{v^2}{v_d^2} \right), \]

\[ \tau^2(v) = \tau_d^2 + \frac{(x_0 - x_d)^2}{v_d^2} \left( 1 - \frac{v^2}{v_d^2} \right), \]

where \( \tau_d = \frac{z_d}{v_d} \). Eliminating \( x_0 \) from the system of equations (14) and (15) leads to the expression for the velocity continuation "wavefront":

\[ \tau(x) = \sqrt{\tau_d^2 + \frac{(x - x_d)^2}{v_d^2 - v^2}}. \]

For the case of a point diffractor, the wavefront corresponds precisely to the summation path of the residual migration operator (Rothman et al., 1985). It has a
hyperbolic shape when \( v_d > v \) (undermigration) and an elliptic shape when \( v_d < v \) (overmigration). The wavefront collapses to a point when the velocity \( v \) approaches the actual effective velocity \( v_d \). At zero velocity, \( v = 0 \), the wavefront takes the familiar form of the post-stack migration hyperbolic summation path. The form of the velocity rays and wavefronts is illustrated in the left plot of Figure 2.

Figure 2: Kinematic velocity continuation in the post-stack migration domain. Solid lines denote wavefronts: reflector images for different migration velocities; dashed lines denote velocity rays. a: the case of a point diffractor. b: the case of a dipping plane reflector.

Another important example is the case of a dipping plane reflector. For simplicity, let us put the origin of the midpoint coordinate \( x \) at the point of the plane intersection with the surface of observations. In this case, the depth of the plane reflector corresponding to the surface point \( x \) has the simple expression

\[
z_p(x) = x \tan \alpha ,
\]

where \( \alpha \) is the dip angle. The zero-offset reflection traveltime \( \tau_0(x_0) \) is the plane with a changed angle. It can be expressed as

\[
\tau_0(x_0) = p x_0 ,
\]

where \( p = \frac{\sin \alpha}{v_p} \), and \( v_p \) is half of the actual velocity. Applying formulas (6) leads to the following parametric expression for the velocity rays:

\[
x(v) = x_0 (1 - p^2 v^2) ,
\]

\[
\tau(v) = p x_0 \sqrt{1 - p^2 v^2} .
\]

Eliminating \( x_0 \) from the system of equations (19) and (20) shows that the velocity continuation wavefronts are planes with a modified angle:

\[
\tau(x) = \frac{p x}{\sqrt{1 - p^2 v^2}} .
\]

The right plot of Figure 2 shows the geometry of the kinematic velocity continuation for the case of a plane reflector.
Kinematics of Residual NMO

The residual NMO differential equation is the second term in equation (1):

\[
\frac{\partial \tau}{\partial v} = \frac{h^2}{v^3 \tau}.
\]  \hspace{1cm} (22)

Equation (22) does not depend on the midpoint \(x\). This fact indicates the one-dimensional nature of normal moveout. The general solution of equation (22) is obtained by simple integration. It takes the form

\[
\tau^2(v) = C - \frac{h^2}{v^2} = \tau^2_1 + h^2 \left( \frac{1}{v^2_1} - \frac{1}{v^2} \right),
\]  \hspace{1cm} (23)

where \(C\) is an arbitrary velocity-independent constant, and I have chosen the constants \(\tau_1\) and \(v_1\) so that \(\tau(v_1) = \tau_1\). Equation (23) is applicable only for \(v\) different from zero.

For the case of a point diffractor, equation (23) easily combines with the zero-offset solution (16). The result is a simplified approximate version of the prestack residual migration summation path:

\[
\tau(x) = \sqrt{\tau^2_d + \left( \frac{x - x_d}{v^2_d - v^2} \right)^2 + h^2 \left( \frac{1}{v^2_d} - \frac{1}{v^2} \right)^2}.
\]  \hspace{1cm} (24)

Summation paths of the form (24) for a set of diffractors with different depths are plotted in Figures 3 and 4. The parameters chosen in these plots allow a direct comparison with Etgen’s Figures 2.4 and 2.5 (Etgen, 1990), based on the exact solution and reproduced in Figures 8 and 9. The comparison shows that the approximate solution (24) captures the main features of the prestack residual migration operator, except for the residual DMO cusps appearing in the exact solution when the diffractor depth is smaller than the offset.

![Figure 3: Summation paths of the simplified prestack residual migration for a series of depth diffractors. Residual slowness \(v/v_d\) is 1.2; half-offset \(h\) is 1 km. This figure is to be compared with Etgen’s Figure 2.4, reproduced in Figure 8.](image)

Neglecting the residual DMO term in residual migration is approximately equivalent in accuracy to neglecting the DMO step in conventional processing. Indeed,
as follows from the geometric analog of equation (1) derived in Appendix A [equation (A-17)], dropping the residual DMO term corresponds to the condition

\[ \tan^2 \alpha \tan^2 \theta \ll 1, \tag{25} \]

where \( \alpha \) is the dip angle, and \( \theta \) is the reflection angle. As shown by Yilmaz and Claerbout (1980), the conventional processing sequence without the DMO step corresponds to the separable approximation of the double-square-root equation (A-4):

\[ \sqrt{1 - v^2 \left( \frac{\partial t}{\partial s} \right)^2} + \sqrt{1 - v^2 \left( \frac{\partial t}{\partial r} \right)^2} \approx 2 \sqrt{1 - v^2 \left( \frac{\partial t}{\partial x} \right)^2} + 2 \sqrt{1 - v^2 \left( \frac{\partial t}{\partial h} \right)^2} - 2, \tag{26} \]

where \( t \) is the reflection traveltime, and \( s \) and \( r \) are the source and receiver coordinates: \( s = x - h, r = x + h \). In geometric terms, approximation (26) transforms to

\[ \cos \alpha \cos \theta \approx \sqrt{1 - \sin^2 \alpha \cos^2 \theta} + \sqrt{1 - \sin^2 \theta \cos^2 \alpha} - 1. \tag{27} \]

Taking the difference of the two sides of equation (27), one can estimate its accuracy by the first term of the Taylor series for small \( \alpha \) and \( \theta \). The estimate is \( \frac{3}{4} \tan^2 \alpha \tan^2 \theta \) (Yilmaz and Claerbout 1980), which agrees qualitatively with (25). Although approximation (24) fails in situations where the dip moveout correction is necessary, it is significantly more accurate than the 15-degree approximation of the double-square-root equation, implied in the migration velocity analysis method of Yilmaz and Chambers (1984) and MacKay and Abma (1992). The 15-degree approximation

\[ \sqrt{1 - v^2 \left( \frac{\partial t}{\partial s} \right)^2} + \sqrt{1 - v^2 \left( \frac{\partial t}{\partial r} \right)^2} \approx 2 - \frac{v^2}{2} \left( \frac{\partial t}{\partial s} \right)^2 + \left( \frac{\partial t}{\partial r} \right)^2 \tag{28} \]

corresponds geometrically to the equation

\[ 2 \cos \alpha \cos \theta \approx \frac{3 + \cos 2\alpha \cos 2\theta}{2}. \tag{29} \]
Its estimated accuracy (from the first term of the Taylor series) is $\frac{1}{8} \tan^2 \alpha + \frac{1}{8} \tan^2 \theta$. Unlike the separable approximation, which is accurate separately for zero offset and zero dip, the 15-degree approximation fails at zero offset in the case of a steep dip and at zero dip in the case of a large offset.

**Kinematics of Residual DMO**

The partial differential equation for kinematic residual DMO is the third term in equation (1):

$$\frac{\partial \tau}{\partial v} = -\frac{h^2 v}{\tau} \left( \frac{\partial \tau}{\partial x} \right)^2 \left( \frac{\partial \tau}{\partial h} \right)^2.$$  \hspace{1cm} (30)

It is more convenient to consider the residual dip-moveout process coupled with residual normal moveout. Etgen (1990) describes this procedure as the cascade of inverse DMO with the initial velocity $v_0$, residual NMO, and DMO with the updated velocity $v_1$. The kinematic equation for residual NMO+DMO is the sum of the two terms in (1):

$$\frac{\partial \tau}{\partial v} = \frac{h^2}{v^3 \tau} \left( 1 - v^4 \left( \frac{\partial \tau}{\partial x} \right)^2 \left( \frac{\partial \tau}{\partial h} \right)^2 \right).$$  \hspace{1cm} (31)

The derivation of the residual DMO+NMO kinematics is detailed in Appendix B. Figure 5 illustrates it with the theoretical impulse response curves. Figure 6 compares the theoretical curves with the result of an actual cascade of the inverse DMO, residual NMO, and DMO operators.

Figure 7 illustrates the residual NMO+DMO velocity continuation for two particularly interesting cases. The left plot shows the continuation for a point diffractor. One can see that when the velocity error is large, focusing of the velocity rays forms a distinctive loop on the zero-offset hyperbola. The right plot illustrates the case of a plane dipping reflector. The image of the reflector shifts both vertically and laterally with the change in NMO velocity.

The full residual migration operator is the chain of residual zero-offset migration and residual NMO+DMO. I illustrate the kinematics of this operator in Figures 8 and 9, which are designed to match Etgen’s Figures 2.4 and 2.5 (Etgen, 1990). A comparison with Figures 3 and 4 shows that including the residual DMO term affects the images of objects with the depth smaller than the half-offset $h$. This term complicates the residual migration operator with cusps.
Figure 5: Theoretical kinematics of the residual NMO+DMO impulse responses for three impulses. Left plot: the velocity ratio $v_1/v_0$ is 1.333. Right plot: the velocity ratio $v_1/v_0$ is 0.833. In both cases the half-offset $h$ is 1 km.

Figure 6: The result of residual NMO+DMO (cascading inverse DMO, residual NMO, and DMO) for three impulses. Left plot: the velocity ratio $v_1/v_0$ is 1.333. Right plot: the velocity ratio $v_1/v_0$ is 0.833. In both cases the half-offset $h$ is 1 km.
Figure 7: Kinematic velocity continuation for residual NMO+DMO. Solid lines denote wavefronts: zero-offset traveltime curves; dashed lines denote velocity rays. a: the case of a point diffractor; the velocity ratio $v_1/v_0$ changes from 0.9 to 1.1. b: the case of a dipping plane reflector; the velocity ratio $v_1/v_0$ changes from 0.8 to 1.2. In both cases, the half-offset $h$ is 2 km.

Figure 8: Summation paths of prestack residual migration for a series of depth diffractors. Residual slowness $v/v_d$ is 1.2; half-offset $h$ is 1 km. This figure reproduces Etgen’s Figure 2.4.

Figure 9: Summation paths of prestack residual migration for a series of depth diffractors. Residual slowness $v/v_d$ is 0.8; half-offset $h$ is 1 km. This figure reproduces Etgen’s Figure 2.5.
FROM KINEMATICS TO DYNAMICS

The theory of characteristics (Courant and Hilbert, 1989) states that if a partial differential equation has the form

$$\sum_{i,j=1}^{n} \Lambda_{ij}(\xi_1, \ldots, \xi_n) \frac{\partial^2 P}{\partial \xi_i \partial \xi_j} + F\left(\xi_1, \ldots, \xi_n, P, \frac{\partial P}{\partial \xi_1}, \ldots, \frac{\partial P}{\partial \xi_n}\right) = 0,$$  \hspace{1cm} (32)

where $F$ is some arbitrary function, and if the eigenvalues of the matrix $\Lambda$ are nonzero, and one of them is different in sign from the others, then equation (32) describes a wave-type process, and its kinematic counterpart is the characteristic equation

$$\sum_{i,j=1}^{n} \Lambda_{ij}(\xi_1, \ldots, \xi_n) \frac{\partial \psi}{\partial \xi_i} \frac{\partial \psi}{\partial \xi_j} = 0$$ \hspace{1cm} (33)

with the characteristic surface

$$\psi(\xi_1, \ldots, \xi_n) = 0$$ \hspace{1cm} (34)

corresponding to the wavefront. In velocity continuation problems, it is appropriate to choose the variable $\xi_1$ to denote the time $t$, $\xi_2$ to denote the velocity $v$, and the rest of the $\xi$-variables to denote one or two lateral coordinates $x$. Without loss of generality, let us set the characteristic surface to be

$$\psi = t - \tau(x; v) = 0,$$ \hspace{1cm} (35)

and use the theory of characteristics to reconstruct the main (second-order) part of the dynamic differential equation from the corresponding kinematic equations. As in the preceding section, it is convenient to consider separately the three different components of the prestack velocity continuation process.

**Dynamics of Zero-Offset Velocity Continuation**

In the case of zero-offset velocity continuation, the characteristic equation is reconstructed from equation (2) to have the form

$$\frac{\partial \psi}{\partial v} \frac{\partial \psi}{\partial t} + v t \left( \frac{\partial \psi}{\partial x} \right)^2 = 0,$$ \hspace{1cm} (36)

where $\tau$ is replaced by $t$ according to equation (35). According to equation (32), the corresponding dynamic equation is

$$\frac{\partial^2 P}{\partial v \partial t} + v t \frac{\partial^2 P}{\partial x^2} + F\left(x, t, v, P, \frac{\partial P}{\partial t}, \frac{\partial P}{\partial v}, \frac{\partial P}{\partial x}\right) = 0,$$ \hspace{1cm} (37)

where the function $F$ remains to be defined. The simplest case of $F$ equal to zero corresponds to Claerbout’s velocity continuation equation (Claerbout, 1986), derived
in a different way. Levin (1986a) provides the dispersion-relation derivation, conceptually analogous to applying the method of characteristics.

In high-frequency asymptotics, the wavefield $P$ can be represented by the ray-theoretical (WKBJ) approximation,

$$ P(t, x, v) \approx A(x, v) f(t - \tau(x, v)) , $$

where $A$ is the amplitude, $f$ is the short (high-frequency) wavelet, and the function $\tau$ satisfies the kinematic equation (2). Substituting approximation (38) into the dynamic velocity continuation equation (37), collecting the leading-order terms, and neglecting the $F$ function leads to the partial differential equation for amplitude transport:

$$ \frac{\partial A}{\partial v} = v \tau \left( 2 \frac{\partial A}{\partial x} \frac{\partial \tau}{\partial x} + A \frac{\partial^2 \tau}{\partial x^2} \right) . $$

(39)

The general solution of equation (39) follows from the theory of characteristics. It takes the form

$$ A(x, v) = A(x_0, 0) \exp \left( \int_0^v u \tau(x, u) \frac{\partial^2 \tau(x, u)}{\partial x^2} \, du \right) , $$

(40)

where the integral corresponds to the curvilinear integration along the corresponding velocity ray, and $x_0$ corresponds to the starting point of the ray. In the case of a plane dipping reflector, the image of the reflector remains plane in the velocity continuation process. Therefore, the second traveltime derivative $\frac{\partial^2 \tau(x, u)}{\partial x^2}$ in (40) equals zero, and the exponential is equal to one. This means that the amplitude of the image does not change with the velocity along the velocity rays. This fact does not agree with the theory of conventional post-stack migration, which suggests downscaling the image by the “cosine” factor $\frac{\tau_0}{\tau}$ (Chun and Jacewitz, 1981; Levin, 1986b). The simplest way to include the cosine factor in the velocity continuation equation is to set the function $F$ to be $\frac{1}{t} \frac{\partial P}{\partial v}$. The resulting differential equation

$$ \frac{\partial^2 P}{\partial v \partial t} + v t \frac{\partial^2 P}{\partial x^2} + \frac{1}{t} \frac{\partial P}{\partial v} = 0 $$

(41)

has the amplitude transport

$$ A(x, v) = \frac{\tau_0}{\tau} A(x_0, 0) \exp \left( \int_0^v u \tau(x, u) \frac{\partial^2 \tau(x, u)}{\partial x^2} \, du \right) , $$

(42)

corresponding to the differential equation

$$ \frac{\partial A}{\partial v} = v \tau \left( 2 \frac{\partial A}{\partial x} \frac{\partial \tau}{\partial x} + A \frac{\partial^2 \tau}{\partial x^2} \right) - A \frac{1}{\tau} \frac{\partial \tau}{\partial v} . $$

(43)

Appendix C proves that the time-and-space solution of the dynamic velocity continuation equation (41) coincides with the conventional Kirchhoff migration operator.
Dynamics of Residual NMO

According to the theory of characteristics, described in the beginning of this section, the kinematic residual NMO equation (22) corresponds to the dynamic equation of the form

$$\frac{\partial P}{\partial v} + \frac{h^2}{v^3 t} \frac{\partial P}{\partial t} + F(h, t, v, P) = 0$$

(44)

with the undetermined function $F$. In the case of $F = 0$, the general solution is easily found to be

$$P(t, h, v) = \phi \left( t^2 + \frac{h^2}{v^2} \right).$$

(45)

where $\phi$ is an arbitrary smooth function. The combination of dynamic equations (44) and (41) leads to an approximate prestack velocity continuation with the residual DMO effect neglected. To accomplish the combination, one can simply add the term $\frac{h^2}{v^3 t^2} \frac{\partial^2 P}{\partial t^2}$ from equation (44) to the left-hand side of equation (41). This addition changes the kinematics of velocity continuation, but does not change the amplitude properties embedded in the transport equation (42).

Dunkin and Levin (1973) and Hale (1983) advocate using an amplitude correction term in the NMO step. This term can be easily added by selecting an appropriate function $F$ in equation (44). The choice $F = \frac{h^2}{v^3 t^2} P$ results in the equation

$$\frac{\partial P}{\partial v} + \frac{h^2}{v^3 t^2} \left( t \frac{\partial P}{\partial t} + P \right) = 0$$

(46)

with the general solution

$$P(t, h, v) = \frac{1}{t} \phi \left( t^2 + \frac{h^2}{v^2} \right),$$

(47)

which has the Dunkin-Levin amplitude correction term.

Dynamics of Residual DMO

The case of residual DMO complicates the building of a dynamic equation because of the essential nonlinearity of the kinematic equation (30). One possible way to linearize the problem is to increase the order of the equation. In this case, the resultant dynamic equation would include a term that has the second-order derivative with respect to velocity $v$. Such an equation describes two different modes of wave propagation and requires additional initial conditions to separate them. Another possible way to linearize equation (30) is to approximate it at small dip angles. In this case, the dynamic equation would contain only the first-order derivative with respect to the velocity and high-order derivatives with respect to the other parameters. The third, and probably the most attractive, method is to change the domain of consideration. For example, one could switch from the common-offset domain to the
domain of offset dip. This method implies a transformation similar to slant stacking
of common-midpoint gathers in the post-migration domain in order to obtain the
local offset dip information. Equation (30) transforms, with the help of the results
from Appendix A, to the form

$$\frac{v^3 \partial \tau}{\partial x} = \frac{\tau \sin^2 \theta}{\cos^2 \alpha - \sin^2 \theta},$$

(48)

with

$$\cos^2 \alpha = \left(1 + v^2 \left(\frac{\partial \tau}{\partial h}\right)^2\right)^{-1},$$

(49)

and

$$\sin^2 \alpha = v^2 \left(\frac{\partial \tau}{\partial h}\right)^2 \left(1 + v^2 \left(\frac{\partial \tau}{\partial h}\right)^2\right)^{-1}.$$  

(50)

For a constant offset dip $\tan \theta = v \frac{\partial \tau}{\partial h}$, the dynamic analog of equation (48) is the
third-order partial differential equation

$$v \cot^2 \theta \frac{\partial^3 P}{\partial t^2 \partial v} - v^3 \frac{\partial^3 P}{\partial x^2 \partial v} + t \frac{\partial^3 P}{\partial t^2 \partial v} + v^2 t \frac{\partial^3 P}{\partial x^2 \partial t} = 0.$$  

(51)

Equation (51) does not strictly comply with the theory of second-order linear differential
equations. Its properties and practical applicability require further research.

**CONCLUSIONS**

I have derived kinematic and dynamic equations for residual time migration in the
form of a continuous velocity continuation process. This derivation explicitly decomposes prestack velocity continuation into three parts corresponding to zero-offset continuation, residual NMO, and residual DMO. These three parts can be treated separately both for simplicity of theoretical analysis and for practical purposes. It is important to note that in the case of a three-dimensional migration, all three components of velocity continuation have different dimensionality. Zero-offset continuation is fully 3-D. It can be split into two 2-D continuations in the in- and cross-line directions. Residual DMO is a two-dimensional common-azimuth process. Residual NMO is a 1-D single-trace procedure.

The dynamic properties of zero-offset velocity continuation are precisely equivalent to those of conventional post-stack migration methods such as Kirchhoff migration. Moreover, the Kirchhoff migration operator coincides with the integral solution of the velocity continuation differential equation for continuation from the zero velocity plane.

This rigorous theory of velocity continuation gives us new insights into the methods of prestack migration velocity analysis. Extensions to the case of depth migration in a variable velocity background are developed by [Liu and McMechan (1996)] and
A practical application of velocity continuation to migration velocity analysis is demonstrated in the companion paper (Fomel, 2003b), where the general theory is used to design efficient and practical algorithms.

ACKNOWLEDGMENTS

This work was completed when the author was a member of the Stanford Exploration Project (SEP) at Stanford University. The financial support was provided by the SEP sponsors.

I thank Bee Bednar, Biondo Biondi, Jon Claerbout, Sergey Goldin, Bill Harlan, David Lumley, and Bill Symes for useful and stimulating discussions. Paul Fowler, Hugh Geiger, Samuel Gray, and one anonymous reviewer provided valuable suggestions that improved the quality of the paper.

REFERENCES


APPENDIX A

DERIVING THE KINEMATIC EQUATIONS

The main goal of this appendix is to derive the partial differential equation describing the image surface in a depth-midpoint-offset-velocity space.

Figure A-1: Reflection rays in a constant velocity medium (a scheme).

The derivation starts with observing a simple geometry of reflection in a constant-velocity medium, shown in Figure A-1. The well-known equations for the apparent slowness

\[
\frac{\partial t}{\partial s} = \frac{\sin \alpha_1}{v}, \quad (A-1)
\]

\[
\frac{\partial t}{\partial r} = \frac{\sin \alpha_2}{v}, \quad (A-2)
\]

relate the first-order traveltine derivatives for the reflected waves to the emergence angles of the incident and reflected rays. Here s stands for the source location at
the surface, \( r \) is the receiver location, \( t \) is the reflection traveltime, \( v \) is the constant velocity, and \( \alpha_1 \) and \( \alpha_2 \) are the angles shown in Figure A-1. Considering the traveltime derivative with respect to the depth of the observation surface \( z \) shows that the contributions of the two branches of the reflected ray, added together, form the equation

\[
-\frac{\partial t}{\partial z} = \frac{\cos \alpha_1}{v} + \frac{\cos \alpha_2}{v} .
\]  

(A-3)

It is worth mentioning that the elimination of angles from equations (A-1), (A-2), and (A-3) leads to the famous double-square-root equation,

\[
-v \frac{\partial t}{\partial z} = \sqrt{1 - v^2 \left( \frac{\partial t}{\partial s} \right)^2} + \sqrt{1 - v^2 \left( \frac{\partial t}{\partial r} \right)^2} ,
\]  

(A-4)

published in the Russian literature by Belonosova and Alekseev (1967) and commonly used in the form of a pseudo-differential dispersion relation (Clayton, 1978; Claerbout, 1985) for prestack migration (Yilmaz, 1979; Popovici, 1995). Considered locally, equation (A-4) is independent of the constant velocity assumption and enables recursive prestack downward continuation of reflected waves in heterogeneous isotropic media.

Introducing the midpoint coordinate \( x = \frac{s + r}{2} \) and half-offset \( h = \frac{r - s}{2} \), one can apply the chain rule and elementary trigonometric equalities to formulas (A-1) and (A-2) and transform these formulas to

\[
\frac{\partial t}{\partial x} = \frac{\partial t}{\partial s} + \frac{\partial t}{\partial r} = \frac{2 \sin \alpha \cos \theta}{v} ,
\]  

(A-5)

\[
\frac{\partial t}{\partial h} = \frac{\partial t}{\partial r} - \frac{\partial t}{\partial s} = \frac{2 \cos \alpha \sin \theta}{v} ,
\]  

(A-6)

where \( \alpha = \frac{\alpha_1 + \alpha_2}{2} \) is the dip angle, and \( \theta = \frac{\alpha_2 - \alpha_1}{2} \) is the reflection angle (Clayton, 1978; Claerbout, 1985). Equation (A-3) transforms analogously to

\[
-\frac{\partial t}{\partial z} = \frac{2 \cos \alpha \cos \theta}{v} .
\]  

(A-7)

This form of equation (A-3) is used to describe the stretching factor of the waveform distortion in depth migration (Tygel et al., 1994).

Dividing (A-5) and (A-6) by (A-7) leads to

\[
\frac{\partial z}{\partial x} = -\tan \alpha ,
\]  

(A-8)

\[
\frac{\partial z}{\partial h} = -\tan \theta .
\]  

(A-9)
Equation (A-9) is the basis of the angle-gather construction of Sava and Fomel (2003). Substituting formulas (A-8) and (A-9) into equation (A-7) yields yet another form of the double-square-root equation:

$$-\frac{\partial t}{\partial z} = \frac{2}{v} \left[ \sqrt{1 + \left( \frac{\partial z}{\partial x} \right)^2} \right] \left[ \sqrt{1 + \left( \frac{\partial z}{\partial h} \right)^2} \right]^{-1}, \quad (A-10)$$

which is analogous to the dispersion relationship of Stolt prestack migration (Stolt, 1978).

The law of sines in the triangle formed by the incident and reflected ray leads to the explicit relationship between the traveltime and the offset:

$$v t = 2 h \frac{\cos \alpha_1 + \cos \alpha_2}{\sin (\alpha_2 - \alpha_1)} = 2 h \frac{\cos \alpha}{\sin \theta}. \quad (A-11)$$

An algebraic combination of formulas (A-11), (A-5), and (A-6) forms the basic kinematic equation of the offset continuation theory (Fomel, 2003a):

$$\frac{\partial t}{\partial h} \left( t^2 + \frac{4 h^2}{v^2} \right) = h t \left( \frac{4}{v^2} + \left( \frac{\partial t}{\partial h} \right)^2 - \left( \frac{\partial t}{\partial x} \right)^2 \right). \quad (A-12)$$

Differentiating (A-11) with respect to the velocity $v$ yields

$$-v^2 \frac{\partial t}{\partial v} = 2 h \frac{\cos \alpha}{\sin \theta}. \quad (A-13)$$

Finally, dividing (A-13) by (A-7) produces

$$v \frac{\partial z}{\partial v} = \frac{h}{\cos \theta \sin \theta}. \quad (A-14)$$

Equation (A-14) can be written in a variety of ways with the help of an explicit geometric relationship between the half-offset $h$ and the depth $z$,

$$h = z \frac{\sin \theta \cos \theta}{\cos^2 \alpha - \sin^2 \theta}, \quad (A-15)$$

which follows directly from the trigonometry of the triangle in Figure A-1 (Fomel, 2003a). For example, equation (A-14) can be transformed to the form obtained by Liu and Bleistein (1995):

$$v \frac{\partial z}{\partial v} = \frac{z}{\cos^2 \alpha - \sin^2 \theta} = \frac{z}{\cos \alpha_1 \cos \alpha_2}. \quad (A-16)$$

In order to separate different factors contributing to the velocity continuation process, one can transform this equation to the form

$$v \frac{\partial z}{\partial v} = \frac{z}{\cos^2 \alpha} + \frac{h^2}{z} \left( 1 - \tan^2 \alpha \tan^2 \theta \right)$$

$$= z \left( 1 + \left( \frac{\partial z}{\partial x} \right)^2 \right) + \frac{h^2}{z} \left( 1 - \left( \frac{\partial z}{\partial x} \right)^2 \left( \frac{\partial z}{\partial h} \right)^2 \right). \quad (A-17)$$
Rewritten in terms of the vertical traveltime $\tau = z/v$, it further transforms to equation

$$\frac{\partial \tau}{\partial v} = v \tau \left( \frac{\partial \tau}{\partial x} \right)^2 + \frac{h^2}{v^3 \tau} \left( 1 - v^4 \left( \frac{\partial \tau}{\partial x} \right)^2 \left( \frac{\partial \tau}{\partial h} \right)^2 \right), \quad (A-18)$$

equivalent to equation (1) in the main text. Yet another form of the kinematic velocity continuation equation follows from eliminating the reflection angle $\theta$ from equations (A-14) and (A-15). The resultant expression takes the following form:

$$v \frac{\partial z}{\partial v} = \frac{2 (z^2 + h^2)}{\sqrt{z^2 + h^2 \sin^2 2\alpha + z \cos 2\alpha}} = \frac{z \cos^2 \alpha + \frac{2 h^2}{\sqrt{z^2 + h^2 \sin^2 2\alpha + z}}} . \quad (A-19)$$

**APPENDIX B**

**DERIVATION OF THE RESIDUAL DMO KINEMATICS**

This appendix derives the kinematical laws for the residual NMO+DMO transformation in the prestack offset continuation process.

The direct solution of equation (31) is nontrivial. A simpler way to obtain this solution is to decompose residual NMO+DMO into three steps and to evaluate their contributions separately. Let the initial data be the zero-offset reflection event $\tau_0(x_0)$. The first step of the residual NMO+DMO is the inverse DMO operator. One can evaluate the effect of this operator by means of the offset continuation concept (Fomel, 2003a). According to this concept, each point of the input traveltime curve $\tau_0(x_0)$ travels with the change of the offset from zero to $h$ along a special trajectory, which I call a *time ray*. Time rays are parabolic curves of the form

$$x(\tau) = x_0 + \frac{\tau^2 - \tau_0^2(x_0)}{\tau_0(x_0) \tau_0'(x_0)}, \quad (B-1)$$

with the final points constrained by the equation

$$h^2 = \frac{\tau^2 - \tau_0^2(x_0)}{(\tau_0(x_0) \tau_0'(x_0))^2}, \quad (B-2)$$

where $\tau_0'(x_0)$ is the derivative of $\tau_0(x_0)$. The second step of the cumulative residual NMO+DMO process is the residual normal moveout. According to equation (23), residual NMO is a one-trace operation transforming the traveltime $\tau$ to $\tau_1$ as follows:

$$\tau_1^2 = \tau^2 + h^2 d, \quad (B-3)$$

where

$$d = \left( \frac{1}{v_0^2} - \frac{1}{v_1^2} \right). \quad (B-4)$$
The third step is dip moveout corresponding to the new velocity $v_1$. DMO is the offset continuation from $h$ to zero offset along the redefined time rays (Fomel, 2003a)

$$x_2(\tau_2) = x + \frac{hX}{\tau_1^2 H} \left( \tau_1^2 - \tau_2^2 \right), \quad (B-5)$$

where $H = \frac{\partial \tau_1}{\partial h}$, and $X = \frac{\partial \tau_1}{\partial x}$. The end points of the time rays (B-5) are defined by the equation

$$\tau_2^2 = -\tau_1^2 \frac{H}{hX^2} \cdot \quad (B-6)$$

The partial derivatives of the common-offset traveltimes are constrained by the offset continuation kinematic equation

$$h(H^2 - X^2) = \tau_1 H, \quad (B-7)$$

which is equivalent to equation (A-12) in Appendix A. Additionally, as follows from equations (B-3) and the ray invariant equations from (Fomel, 2003a),

$$\tau_1 X = \tau \frac{\partial \tau}{\partial x} = \frac{\tau^2 \tau_0'(x_0)}{\tau_0(x_0)}, \quad (B-8)$$

Substituting (B-1-B-4) and (B-7-B-8) into equations (B-5) and (B-6) and performing the algebraic simplifications yields the parametric expressions for velocity rays of the residual NMO+DMO process:

\[
\begin{align*}
    x_2(d) &= x_0 + \frac{h^2 \tau_0'(x_0)}{T} \left( 1 - \frac{T^2}{T_2(d)} \right), \\
    \tau(d) &= \frac{\tau_1^2(d)}{T_2(d)},
\end{align*}
\]

where the function $T(h, \tau_0(x_0), \tau_0'(x_0))$ is defined by

$$T(h, \tau, \tau_x) = \tau + \sqrt{\tau^2 + 4h^2 \tau_x^2}, \quad (B-10)$$

$$T_2(d) = \sqrt{T(h, \tau_1^2(d), \tau_0'(x_0)) T(h, \tau_0(x_0), \tau_0'(x_0)))}, \quad (B-11)$$

and

$$\tau_1^2(d) = \tau_0 T + dh^2. \quad (B-12)$$

The last step of the cascade of inverse DMO, residual NMO, and DMO is illustrated in Figure B-1. The three plots in the figure show the offset continuation to zero offset of the inverse DMO impulse response shifted by the residual NMO operator. The middle plot corresponds to zero NMO shift, for which the DMO step collapses the wavefront back to a point. Both positive (top plot) and negative (bottom plot) NMO shifts result in the formation of the specific triangular impulse response of the residual NMO+DMO operator. As noticed by Etgen (1990), the size of the triangular operators dramatically decreases with the time increase. For large times (pseudo-depths) of the initial impulses, the operator collapses to a point corresponding to the pure NMO shift.
Figure B-1: Kinematic residual NMO+DMO operators constructed by the cascade of inverse DMO, residual NMO, and DMO. The impulse response of inverse DMO is shifted by the residual NMO procedure. Offset continuation back to zero offset forms the impulse response of the residual NMO+DMO operator. Solid lines denote traveltine curves; dashed lines denote the offset continuation trajectories (time rays). Top plot: $v_1/v_0 = 1.2$. Middle plot: $v_1/v_0 = 1$; the inverse DMO impulse response collapses back to the initial impulse. Bottom plot: $v_1/v_0 = 0.8$. The half-offset $h$ in all three plots is 1 km.
APPENDIX C

INTEGRAL VELOCITY CONTINUATION AND KIRCHHOFF MIGRATION

The main goal of this appendix is to prove the equivalence between the result of zero-offset velocity continuation from zero velocity and conventional post-stack migration. After solving the velocity continuation problem in the frequency domain, I transform the solution back to the time-and-space domain and compare it with the conventional Kirchhoff migration operator (Schneider, 1978). The frequency-domain solution has its own value, because it forms the basis for an efficient spectral algorithm for velocity continuation (Fomel, 2003b).

Zero-offset migration based on velocity continuation is the solution of the boundary problem for equation (41) with the boundary condition
\[ P|_{v=0} = P_0, \tag{C-1} \]
where \( P_0(t_0, x_0) \) is the zero-offset seismic section, and \( P(t, x, v) \) is the continued wavefield. In order to find the solution of the boundary problem composed of (41) and (C-1), it is convenient to apply the function transformation \( R(t, x, v) = t P(t, x, v) \), the time coordinate transformation \( \sigma = t^2/2 \), and, finally, the double Fourier transform over the squared time coordinate \( \sigma \) and the spatial coordinate \( x \):
\[ \hat{R}(v) = \int \int P(t, x, v) \exp(i\Omega \sigma - ikx) t^2 dt dx. \tag{C-2} \]

With the change of domain, equation (41) transforms to the ordinary differential equation
\[ \frac{d\hat{R}}{dv} = i \frac{k^2}{\Omega} v \hat{R}, \tag{C-3} \]
and the boundary condition (C-1) transforms to the initial value condition
\[ \hat{R}(0) = \hat{R}_0, \tag{C-4} \]
where
\[ \hat{R}_0 = \int \int P_0(t_0, x_0) \exp(i\Omega \sigma_0 - ikx_0) t_0^2 dt_0 dx_0, \tag{C-5} \]
and \( \sigma_0 = t_0^2/2 \). The unique solution of the initial value (Cauchy) problem (C-3) is easily found to be
\[ \hat{R}(v) = \hat{R}_0 \exp \left( i \frac{k^2}{2\Omega} v^2 \right). \tag{C-6} \]

In the transformed domain, velocity continuation appears to be a unitary phase-shift operator. An immediate consequence of this remarkable fact is the cascaded migration decomposition of post-stack migration (Larner and Beasley, 1987):
\[ \exp \left( i \frac{k^2}{2\Omega} (v_1^2 + \cdots + v_n^2) \right) = \exp \left( i \frac{k^2}{2\Omega} v_1^2 \right) \cdots \exp \left( i \frac{k^2}{2\Omega} v_n^2 \right). \tag{C-7} \]
Analogously, three-dimensional post-stack migration is decomposed into the two-pass procedure (Jakubowicz and Levin [1983]):

\[
\exp \left( i \frac{k_1^2 + k_2^2}{2 \Omega} v^2 \right) = \exp \left( i \frac{k_1^2}{2 \Omega} v^2 \right) \exp \left( i \frac{k_2^2}{2 \Omega} v^2 \right). \quad (C-8)
\]

The inverse double Fourier transform of both sides of equality (C-6) yields the integral (convolution) operator

\[
P(t, x, v) = \int \int P_0(t_0, x_0) K(t_0, x_0; t, x, v) \, dt_0 \, dx_0, \quad (C-9)
\]

with the kernel \( K \) defined by

\[
K = \frac{t_0^2/t}{(2 \pi)^{m+1}} \int \int \exp \left( i \frac{k^2}{2 \Omega} v^2 + i k (x - x_0) - \frac{i \Omega}{2} (t^2 - t_0^2) \right) \, dk \, d\Omega, \quad (C-10)
\]

where \( m \) is the number of dimensions in \( x \) and \( k \) (\( m \) equals 1 or 2). The inner integral on the wavenumber axis \( k \) in formula (C-10) is a known table integral (Gradshtein and Ryzhik [1994]). Evaluating this integral simplifies equation (C-10) to the form

\[
K = \frac{t_0^2/t}{(2 \pi)^{m/2+1} \, v^m} \int (i \Omega)^{m/2} \exp \left[ \frac{i \Omega}{2} \left( t_0^2 - t^2 - \frac{(x - x_0)^2}{v^2} \right) \right] \, d\Omega. \quad (C-11)
\]

The term \((i \Omega)^{m/2}\) is the spectrum of the anti-causal derivative operator \( \frac{d}{d\sigma} \) of the order \( m/2 \). Noting the equivalence

\[
\left( \frac{\partial}{\partial \sigma} \right)^{m/2} = \left( \frac{1}{t} \frac{\partial}{\partial t} \right)^{m/2} = \left( \frac{1}{t} \right)^{m/2} \left( \frac{\partial}{\partial t} \right)^{m/2}, \quad (C-12)
\]

which is exact in the 3-D case \( (m = 2) \) and asymptotically correct in the 2-D case \( (m = 1) \), and applying the convolution theorem transforms operator (C-9) to the form

\[
P(t, x, v) = \frac{1}{(2 \pi)^{m/2}} \int \frac{\cos \alpha}{(v \rho)^{m/2}} \left( - \frac{\partial}{\partial t_0} \right)^{m/2} P_0 \left( \frac{\rho}{v}, x_0 \right) \, dx_0, \quad (C-13)
\]

where \( \rho = \sqrt{v^2 t^2 + (x - x_0)^2} \), and \( \cos \alpha = t_0/t \). Operator (C-13) coincides with the Kirchhoff operator of conventional post-stack time migration (Schneider [1978]).
Velocity continuation by spectral methods

Sergey Fomel

ABSTRACT
I apply Fourier and Chebyshev spectral methods to derive accurate and efficient algorithms for velocity continuation. As expected, the accuracy of the spectral methods is noticeably superior to that of the finite-difference approach. Both methods apply a transformation of the time axis to squared time. The Chebyshev method is slightly less efficient than the Fourier method, but has less problems with the time transformation and also handles accurately the non-periodic boundary conditions.

INTRODUCTION
In a recent work (Fomel, 1994, 1996), I introduced the process of velocity continuation to describe a continuous transformation of seismic time-migrated images with a change of the migration velocity. Velocity continuation generalizes the ideas of residual migration (Rothman et al., 1985; Etgen, 1990) and cascaded migrations (Larner and Beasley, 1987). In the zero-offset (post-stack) case, the velocity continuation process is governed by a partial differential equation in midpoint, time, and velocity coordinates, first discovered by Claerbout (1986b). Hubral et al. (1996) and Schleicher et al. (1997) describe this process in a broader context of “image waves”. Generalizations are possible for the non-zero offset (prestack) case (Fomel, 1996, 1997).

A numerical implementation of velocity continuation process provides an efficient method of scanning the velocity dimension in the search of an optimally focused image. The first implementations (Li, 1986; Fomel, 1996) used an analogy with Claerbout’s 15-degree depth extrapolation equation to construct a finite-difference scheme with an implicit unconditionally stable advancement in velocity. Fomel and Claerbout (1997) presented an efficient three-dimensional generalization, applying the helix transform (Claerbout, 1997).

A low-order finite-difference method is probably the most efficient numerical approach to this method, requiring the least work per velocity step. However, its accuracy is not optimal because of the well-known numerical dispersion effect. Figure 1 shows impulse responses of post-stack velocity continuation for three impulses, computed by the second-order finite-difference method (Fomel, 1996). As expected from the residual migration theory (Rothman et al., 1985), continuation to a higher velocity (left plot) corresponds to migration with a residual velocity, and its impulse

*e-mail: sergey@sep.stanford.edu
Figure 1: Impulse responses (Green’s functions) of velocity continuation, computed by a second-order finite-difference method. The left plots corresponds to continuation to a larger velocity (+1 km/sec); the right plot, smaller velocity, (−1 km/sec).  

responses have an elliptical shape. Continuation to a smaller velocity (right plot in Figure 1) corresponds to demigration (modeling), and its impulse responses have a hyperbolic shape. The dispersion artifacts are clearly visible in the figure.

In this paper, I explore the possibility of implementing a numerical velocity continuation by spectral methods. I adopted two different methods, comparable in efficiency with finite differences. The first method is a direct application of the Fast Fourier Transform (FFT) technique. The second method transforms the time grid to Chebyshev collocation points, which leads to an application of the Chebyshev–τ method (Lanczos 1956; Gottlieb and Orszag 1977; Boyd 1989), combined with an unconditionally stable implicit advancement in velocity. Both methods employ a transformation of the grid from time $t$ to the squared time $\sigma = t^2$, which removes the dependence on $t$ from the coefficients of the velocity continuation equation. Additionally, the Fourier transform in the space (midpoint) variable $x$ takes care of the spatial dependencies. This transform is a major source of efficiency, because different wavenumber slices can be processed independently on a parallel computer before transforming them back to the physical space.

PROBLEM FORMULATION

The post-stack velocity continuation process is governed by a partial differential equation in the domain, composed by the seismic image coordinates (midpoint $x$ and vertical time $t$) and the additional velocity coordinate $v$. Neglecting some amplitude-correcting terms (Fomel 1996), the equation takes the form (Claerbout 1986b)

$$\frac{\partial^2 P}{\partial v \partial t} + vt \frac{\partial^2 P}{\partial x^2} = 0.$$  

(1)
Equation (1) is linear and belongs to the hyperbolic type. It describes a wave-type process with the velocity \( v \) acting as a “time-like” variable. Each constant-\( v \) slice of the function \( P(x,t,v) \) corresponds to an image with the corresponding constant velocity. The necessary boundary and initial conditions are

\[
P|_{t=T} = 0 \quad P|_{v=v_0} = P_0(x,t)
\]

where \( v_0 \) is the starting velocity, \( T = 0 \) for continuation to a smaller velocity and \( T \) is the largest time on the image (completely attenuated reflection energy) for continuation to a larger velocity. The first case corresponds to “modeling”; the latter case, to seismic migration.

Mathematically, equations (1) and (2) define a Goursat-type problem (Courant, 1962). Its analytical solution can be constructed by a variation of the Riemann method in the form of an integral operator (Fomel, 1994, 1996):

\[
P(t,x,v) = \frac{1}{(2\pi)^{m/2}} \int \frac{1}{(\sqrt{v^2 - v_0^2} \rho)^{m/2}} \left(- \frac{\partial}{\partial t_0}\right)^{m/2} P_0 \left(\frac{\rho}{\sqrt{v^2 - v_0^2}}, x_0 \right) dx_0,
\]

where \( \rho = \sqrt{(v^2 - v_0^2) t^2 + (x - x_0)^2} \), \( m = 1 \) in the 2-D case, and \( m = 2 \) in the 3-D case. In the case of continuation from zero velocity \( v_0 = 0 \), operator (3) is equivalent (up to the amplitude weighting) to conventional Kirchoff time migration (Schneider, 1978). Similarly, in the frequency-wavenumber domain, velocity continuation takes the form

\[
\hat{P}(\omega, k, v) = \hat{P}_0(\sqrt{\omega^2 + k^2(v^2 - v_0^2)}, k),
\]

which is equivalent (up to scaling coefficients) to Stolt migration (Stolt, 1985), regarded as the most efficient migration method.

If our task is to create many constant-velocity slices, there are other ways to construct the solution of problem (1-2). Two alternative spectral approaches are discussed in the next two sections.

**FOURIER APPROACH**

Introducing the change of variable \( \sigma = t^2 \), we can transform equation (1) to the form

\[
2 \frac{\partial^2 P}{\partial v \partial \sigma} + v \frac{\partial^2 P}{\partial x^2} = 0,
\]

whose coefficients don’t depend on the time variables. Double Fourier transform in \( \sigma \) and \( x \) further simplifies equation (6) to the ordinary differential equation

\[
2i\Omega \frac{d^2 \hat{P}}{dv} - v k^2 \hat{P} = 0,
\]
where the frequency $\Omega$ corresponds to the time coordinate $\sigma$, and $k$ is the wavenumber in $x$. Equation (7) has an explicit analytical solution
\[
\hat{P}(k, \Omega, v) = \hat{P}_0(k, \Omega) e^{ik\left(v_0^2 - v^2\right)/4\Omega},
\]
which defines a very simple algorithm for the numerical velocity continuation. The algorithms consists of the following steps:

1. Transform the input from a regular grid in $t$ to a regular grid in $\sigma$.
2. Apply FFT in $x$ and $\sigma$.
3. Multiply by the all-pass phase-shift filter $e^{ik\left(v_0^2 - v^2\right)/4\Omega}$.
4. Inverse FFT in $x$ and $\sigma$.
5. Inverse transform to a regular grid in $t$.

Figure 2: Synthetic seismic data before (left) and after (right) transformation to the $\sigma$ grid.

Figure 7 shows a simple synthetic model of seismic reflection data from (Claerbout, 1995) before and after transforming the grid, regularly spaced in $t$, to a grid, regular in $\sigma$. The left plot of Figure 8 shows the Fourier transform of the data. Except for the nearly vertical event, which corresponds to a stack of parallel layers in the shallow part of the data, the data frequency range is contained near the origin in the $\Omega - k$ space. The right plot of Figure 8 shows the phase-shift filter for continuation from zero imaging velocity (which corresponds to unprocessed data) to the velocity of 1 km/sec. The rapidly oscillating part (small frequencies and large wavenumbers) is exactly in the place, where the data spectrum is zero and corresponds to physically impossible reflection events.

Algorithm (9) is very attractive from the practical point of view because of its efficiency (based on the FFT algorithm). The operations count is roughly the same
Figure 3: Left: the real part of the data Fourier transform. Right: the real part of the velocity continuation operator (continuation from 0 to 1 km/s) in the Fourier domain.

as in Stolt migration (4): two forward and inverse FFTs and forward and inverse grid transform with interpolation (one complex-number transform in the case of Stolt migration). Algorithm (9) can be even more efficient than Stolt method because of the simpler structure of the innermost loop. However, its practical implementation faces two difficult problems: artifacts of the $t^2$ grid transform and wraparound artifacts.

Improving the accuracy of the $t^2$ grid transform

The first problem is the loss of information in the transform to the $t^2$ grid. As illustrated in Figure 7, the shallow part of the data gets severely compressed in the $t^2$ grid. The amount of compression can lead to inadequate sampling, and as a result, aliasing artifacts in the frequency domain. Moreover, it can be difficult to recover from the loss of information in the transformed domain when transforming back into the original grid. A partial remedy for this problem is to increase the grid size in the $t^2$ domain. The top plots in Figure 4 show the result of back transformation to the $t$ grid and the difference between this result and the original model (plotted on the same scale). We can see a noticeable loss of information in the upper (shallow) part of the data, caused by undersampling. The bottom plots in Figure 4 correspond to increasing the grid size by a factor of three. Some of the artifacts have been suppressed, at the expense of dealing with a larger grid.

To perform an accurate transform of the grid, I adopted the following method, inspired by (Claerbout, 1986a). Let $d_{\text{new}}$ denote the data on the new grid and $d_{\text{old}}$ be the data on the old grid. If $L$ is the interpolation operator, defined on the new grid, then the optimal least-square transformation is

$$d_{\text{new}} = (L^T L)^{-1} L d_{\text{old}},$$

(8)
Figure 4: The left plots show the reconstruction of the original data after transforming back from the $t^2$ grid to the original $t$ grid. The right plots show the difference with the original model. Top: using the original grid size ($N_t = 200$). Bottom: increasing the grid size by a factor of three. [velspec/sigvc/ fft-inv]
where \( L^T \) denotes the adjoint interpolation operator. The operator \((L^T L)^{-1}\) provides a proper scaling of the result. If we use simple linear interpolation for the \( L \) operator, then \( L^T L \) is a tridiagonal matrix, which can be easily inverted (in \( 8N \) operations). If some parts in \( d_{\text{new}} \) are not fully constrained, then the tridiagonal matrix is not invertible. To obtain a solution in this case, we can include a regularization operator \( D \) in (8), as follows:

\[
d_{\text{new}} = (L^T L + \epsilon^2 D)^{-1} L d_{\text{old}},
\]

A convenient choice for \( D \) is a second derivative operator, represented with the second-order finite-difference approximation. This operator allows the selection of the smoothest possible function \( d_{\text{new}} \) while preserving the efficient tridiagonal structure of \( L^T L + \epsilon^2 D \). In this problem, the parameter \( \epsilon \) can be chosen as small as possible, as long as it prevents the inversion from getting unstable.

### Suppressing wraparound artifacts of the Fourier method

The periodic boundary conditions both in the squared time \( \sigma \) and the spatial coordinate \( x \), implied by the Fourier approach, are artificial in the problem of velocity continuation. The artificial periodicity is convenient from the computational point of view. However, false periodic events (wraparound artifacts) should be suppressed in the final output. A natural method for attacking this problem is to apply zero padding in the physical space prior to Fourier transform. Of course, this method involves an additional expense of the grid size increase.

The top plots in Figure 5 show the numerical impulse responses of velocity continuation, computed by the Fourier method. The initial data contained three spikes, passed through a narrow-band filter. Theoretically, continuation to larger velocity (the left plot) should create three elliptical wavefronts, and continuation to smaller velocity (right plot) should create three hyperbolic wavefronts (Rothman et al., 1985). We can see that the results are largely contaminated with wraparound artifacts. The result of applying zero padding (the bottom plots in Figure 5) shows most of the artifacts suppressed.

Chebyshev spectral method, discussed in the next section, provides a spectral accuracy while dealing correctly with non-periodic data.

### CHEBYSHEV APPROACH

For an alternative spectral approach, I adopted the Chebyshev-\( \tau \) method (Lanczos, 1956; Gottlieb and Orszag, 1977). The Chebyshev-\( \tau \) velocity continuation algorithm consists of the following steps:

1. Transform the regular grid in \( t \) to Gauss-Lobato collocation points, required for the fast Chebyshev transform. First, a new variable \( \xi \) is introduced by the shift
Figure 5: Impulse responses (Green’s functions) of velocity continuation, computed by the Fourier method. Top: without zero padding, bottom: with zero padding. The left plots correspond to continuation to a larger velocity (+1 km/sec); the right plots, smaller velocity, (−1 km/sec).
transform:
\[ \xi = 1 - \frac{2t^2}{T^2} \]  
so that the domain \(0 \leq t \leq T\) is mapped into the domain \(1 \geq \xi \geq -1\). Second, the \(\xi\) grid points are distributed regularly in the cosine projection: \(\xi_j = \cos\left(\frac{\pi j}{N}\right), j = 0, 1, 2, \ldots, N\).

2. Transform the initial image \(P_0(x,t)\) into the Chebyshev space in \(\xi\) and Fourier transform in \(x\), using the FFT algorithm. The Chebyshev-Fourier representation of \(P_0(x,t)\) is
\[ P_0(x,t) = \sum_{k=-N_x/2}^{N_x/2-1} \sum_{j=0}^{N_t} \hat{P}_{kj}(\xi) e^{ikx}, \]  
where \(T_j\) denotes the Chebyshev polynomial of degree \(j\).

3. Apply equation (1) to advance the image in velocity \(v\). It is convenient to rewrite this equation in the form
\[ \frac{\partial P}{\partial v} = \frac{v T^2}{4} \int d\xi \frac{\partial^2 P}{\partial x^2}. \]  
In the Chebyshev-\(\tau\) domain, the double differentiation in \(x\) is performed by multiplying the Fourier transform of \(P\) by \(-k^2\), and integration in \(\xi\) is performed as a direct operations on the Chebyshev coefficients. In particular, if \(\sum_{j=0}^{N} a_j T_j(\xi)\) is the Chebyshev representation of the function \(f(\xi)\), then the coefficients \(b_j\) of \(\int f(\xi)d\xi\) are defined by the relation
\[ 2j b_j = c_{j-1}a_{j-1} - a_{j+1} \]  
where \(c_0 = 2\), \(c_j = 0\) for \(j < 0\), and \(c_j = 1\) for \(j > 0\). The constant of integration (and, correspondingly, the coefficient \(b_0\)) can be found at each velocity step from the boundary conditions (2), which are transformed to the form
\[ P|_{\xi=-1} = \sum_{j=0}^{N_t} \hat{P}_{kj}(-1)^j = 0. \]

For the velocity advancement I used an implicit Crank-Nicolson scheme, which is unconditionally stable independent of the velocity step size. By writing equation (12) in the matrix form
\[ \frac{\partial P}{\partial v} = A P, \]  
the Crank-Nicolson advancement is represented by the equation
\[ P_{v+dv} = \left( I - A \frac{dv}{2} \right)^{-1} \left( I + A \frac{dv}{2} \right) P_v. \]
where $I$ is the identity matrix. The inverted matrix $(I - A \frac{dv}{2})$ has a tridiagonal structure, except for the first row, implied by the boundary condition (14). A careful treatment of the boundary condition by the matrix-bordering method [Faddeev and Faddeeva, 1963; Boyd, 1989] allows for an efficient inversion at a tridiagonal solver speed.

4. Transform the result of the velocity advancement back to the physical domain.

5. Transform the grid back to being regularly space in $t$.

Figure 6: Synthetic seismic data before (left) and after (right) transformation to the Chebyshev grid in squared time. 

The first advantage of the Chebyshev approach comes from the better conditioning of the grid transform. Figure 6 shows the synthetic data before and after the grid transform. Figure 7 shows a reconstruction of the original data after transforming back from the Chebyshev grid (Gauss-Lobato collocation points). The difference with the original image is negligibly small.

The second advantage is the compactness of the Chebyshev representation. Figure 8 shows the data after the decomposition into Chebyshev polynomials in $\xi$ and Fourier transform in $x$. We observe a very rapid convergence of the Chebyshev representation: a relatively small number of polynomials suffices to represent the data.

The third advantage is the proper handling of the non-periodic boundary conditions. Figure 9 shows the velocity continuation impulse responses, computed by the Chebyshev method. As expected, no wraparound artifacts occur on the time axis, and the accuracy of the result is noticeably higher than in the case of finite differences (Figure 1).
Figure 7: The left plots show the reconstruction of the original data after transforming back from the Chebyshev grid to the original $t$ grid. The right plots show the difference with the original model. Top: using the original grid size ($N_t = 200$). Bottom: increasing the grid size by a factor of three.

Figure 8: Left: Synthetic data after Chebyshev transform. Right: the real part of the Fourier transform in the space coordinate.
Figure 9: Impulse responses (Green’s functions) of velocity continuation, computed by the Chebyshev-τ method. Top: without zero padding, bottom: with zero padding on the $x$ axis. The left plots correspond to continuation to a larger velocity ($+1$ km/sec); the right plots, smaller velocity, ($-1$ km/sec).
CONCLUSIONS

I have applied two spectral methods for a numerical solution of the velocity continuation problem.

The Fourier method is attractive because of its numerical efficiency. However, it requires additional computational effort to suppress numerical artifacts: the inaccuracy of the grid transform and the artificial periodicity in the physical space.

The Chebyshev-τ method is free of most of these difficulties, although its overall efficiency can be slightly inferior to that of the Fourier method.

Both methods possess a “spectral” accuracy, which is highly desired if accuracy is a concern.

Figure 10: Top left: synthetic model (the ideal image). Top right: synthetic data. Bottom left: the result of velocity continuation with the Fourier method. Bottom right: the result of velocity continuation with the Chebyshev method.

Figure 10 compares the results of velocity continuation with different methods. The top left plot shows an implied subsurface model (an “ideal image”). The top
right plot is the corresponding synthetic data. The bottom left plot is the output of the Fourier method, and the bottom right plot is the output of the Chebyshev method. The Fourier result shows a poor quality in the shallow part (caused by subsampling in the \( t^2 \) grid). The wraparound artifacts were suppressed by a zero-padding correction. The quality of the Chebyshev result is noticeably higher. It is close to the best possible accuracy, under the natural limitations of seismic resolution.

REFERENCES

Time migration velocity analysis by velocity continuation

Sergey Fomel

ABSTRACT

Time migration velocity analysis can be performed by velocity continuation, an incremental process that transforms migrated seismic sections according to changes in the migration velocity. Velocity continuation enhances residual normal moveout correction by properly taking into account both vertical and lateral movements of events on seismic images. Finite-difference and spectral algorithms provide efficient practical implementations for velocity continuation. Synthetic and field data examples demonstrate the performance of the method and confirm theoretical expectations.

INTRODUCTION

Migration velocity analysis is a routine part of prestack time migration applications. It serves both as a tool for velocity estimation (Deregowski, 1990) and as a tool for optimal stacking of migrated seismic sections prior to modeling zero-offset data for depth migration (Kim et al., 1997). In the most common form, migration velocity analysis amounts to residual moveout correction on CIP (common image point) gathers. However, in the case of dipping reflectors, this correction does not provide optimal focusing of reflection energy, since it does not account for lateral movement of reflectors caused by the change in migration velocity. In other words, different points on a stacking hyperbola in a CIP gather can correspond to different reflection points at the actual reflector. The situation is similar to that of the conventional normal moveout (NMO) velocity analysis, where the reflection point dispersal problem is usually overcome with the help of dip moveout (Deregowski, 1986; Hale, 1991). An analogous correction is required for optimal focusing in the post-migration domain.

In this paper, I propose and test velocity continuation as a method of migration velocity analysis. The method enhances the conventional residual moveout correction by taking into account lateral movements of migrated reflection events.

Velocity continuation is a process of transforming time migrated images according to the changes in migration velocity. This process has wave-like properties, which have been described in earlier papers (Fomel, 1994, 1997, 2003, Hubral et al., 1996) and Schleicher et al. (1997) use the term image waves to describe a similar concept.

*e-mail: sergey@sep.stanford.edu
Adler (2000, 2002) generalizes the velocity continuation approach for the case of variable background velocities, using the term *Kirchhoff image propagation*. Although the velocity continuation concept is tailored for time migration, it finds important applications in depth migration velocity analysis by recursive methods (Biondi and Sava, 1999; Vaillant et al., 2000).

Applying velocity continuation to migration velocity analysis involves the following steps:

1. prestack common-offset (and common-azimuth) migration - to generate the initial data for continuation,

2. velocity continuation with stacking and semblance analysis across different offsets - to transform the offset data dimension into the velocity dimension,

3. picking the optimal velocity and slicing through the migrated data volume - to generate an optimally focused image.

The first step transforms the data to the image space. The regularity of this space can be exploited for devising efficient algorithms for the next two steps. The idea of slicing through the velocity space goes back to the work of Shurtleff (1984), Fowler (1984, 1988), and Mikulich and Hale (1992). While the previous slicing methods constructed the velocity space by repeated migration with different velocities, velocity continuation navigates directly in the migration velocity space without returning to the original data. This leads to both more efficient algorithms and a better understanding of the theoretical continuation properties (Fomel, 2003).

In this paper, I demonstrate all three steps, using both synthetic data and a North Sea dataset. I introduce and exemplify two methods for the efficient practical implementation of velocity continuation: the finite-difference method and the Fourier spectral method. The Fourier method is recommended as optimal in terms of the accuracy versus efficiency trade-off. Although all the examples in this paper are two-dimensional, the method easily extends to 3-D under the assumption of common-azimuth geometry (one oriented offset). More investigation may be required to extend the method to the multi-azimuth case.

It is also important to note that although the velocity continuation result could be achieved in principle by using prestack residual migration in Kirchhoff (Etgen, 1990) or frequency-wavenumber (Stolt, 1996) formulation, the first is inferior in efficiency, and the second is not convenient for the conventional velocity analysis across different offsets, because it mixes them in the Fourier domain (Sava, 2000). Fourier-domain angle-gather analysis (Sava et al., 2001; Sava and Fomel, 2003) opens new possibilities for the future development of the Fourier-domain velocity continuation. New insights into the possibility of extending the method to depth migration can follow from the work of Adler (2002).
NUMERICAL VELOCITY CONTINUATION IN THE POST-STACK DOMAIN

The post-stack velocity continuation process is governed by a partial differential equation in the domain, composed by the seismic image coordinates (midpoint \(x\) and vertical time \(t\)) and the additional velocity coordinate \(v\). Neglecting some amplitude-correcting terms (Fomel, 2003), the equation takes the form (Claerbout, 1986)

\[
\frac{\partial^2 P}{\partial v \partial t} + v \frac{\partial^2 P}{\partial x^2} = 0.
\]

Equation (1) is linear and belongs to the hyperbolic type. It describes a wave-type process with the velocity \(v\) acting as a propagation variable. Each constant-\(v\) slice of the function \(P(x, t, v)\) corresponds to an image with the corresponding constant velocity. The necessary boundary and initial conditions are

\[
P|_{t=T} = 0 \quad P|_{v=v_0} = P_0(x, t),
\]

where \(v_0\) is the starting velocity, \(T = 0\) for continuation to a smaller velocity and \(T\) is the largest time on the image (completely attenuated reflection energy) for continuation to a larger velocity. The first case corresponds to “modeling” (demigration); the latter case, to seismic migration.

Mathematically, equations (1) and (2) define a Goursat-type problem (Courant and Hilbert, 1989). Its analytical solution can be constructed by a variation of the Riemann method in the form of an integral operator (Fomel, 1994, 2003):

\[
P(t, x, v) = \frac{1}{(2 \pi)^{m/2}} \int \frac{1}{(\sqrt{v^2 - v_0^2} \rho)^{m/2}} \left(-\frac{\partial}{\partial t_0}\right)^{m/2} \hat{P}_0 \left(\frac{\rho}{\sqrt{v^2 - v_0^2}}, x_0\right) \, dx_0,
\]

where \(\rho = \sqrt{(v^2 - v_0^2) t^2 + (x - x_0)^2}\), \(m = 1\) in the 2-D case, and \(m = 2\) in the 3-D case. In the case of continuation from zero velocity \(v_0 = 0\), operator (3) is equivalent (up to the amplitude weighting) to conventional Kirchhoff time migration (Schneider, 1978). Similarly, in the frequency-wavenumber domain, velocity continuation takes the form

\[
\hat{P}(\omega, k, v) = \hat{P}_0(\sqrt{\omega^2 + k^2(v^2 - v_0^2)}, k),
\]

which is equivalent (up to scaling coefficients) to Stolt migration (Stolt, 1978), regarded as the most efficient constant-velocity migration method.

If our task is to create many constant-velocity slices, there are other ways to construct the solution of problem (1-2). Two alternative approaches are discussed in the next two subsections.

Finite-difference approach

The differential equation (1) has a mathematical form analogous to that of the 15-degree wave extrapolation equation (Claerbout, 1976). Its finite-difference implementation, first described by Claerbout (1986) and Li (1986), is also analogous to that of
the 15-degree equation, except for the variable coefficients. One can write the implicit unconditionally stable finite-difference scheme for the velocity continuation equation in the form

\[(I + a^{i+1}_{j+1} \mathbf{T}) \mathbf{P}^{i+1}_{j+1} - (I - a^{i+1}_j \mathbf{T}) \mathbf{P}^{i+1}_j - (I - a^i_{j+1} \mathbf{T}) \mathbf{P}^i_{j+1} + (I + a^i_j \mathbf{T}) \mathbf{P}^i_j = 0, \tag{5}\]

where index \(i\) corresponds to the time dimension, index \(j\) corresponds to the velocity dimension, \(\mathbf{P}\) is a vector along the midpoint direction, \(I\) is the identity matrix, \(\mathbf{T}\) represents the finite-difference approximation to the second-derivative operator in midpoint, and \(a^i_j = v_j t_i \Delta v \Delta t\).

In the two-dimensional case, equation \(5\) reduces to a tridiagonal system of linear equations, which can be easily inverted. In 3-D, a straightforward extension can be obtained by using either directional splitting or helical schemes (Rickett et al., 1998). The direction of stable propagation is either forward in velocity and backward in time or backward in velocity and forward in time as shown in Figure 1.

![Figure 1: Finite-difference scheme for the velocity continuation equation. A stable propagation is either forward in velocity and backward in time (a) or backward in velocity and forward in time (b).](image)

In order to test the performance of the finite-difference velocity continuation method, I use a simple synthetic model from Claerbout (1995). The reflectivity model is shown in Figure 2. It contains several features that challenge the migration performance: dipping beds, unconformity, syncline, anticline, and fault. The velocity is taken to be constant \(v = 1.5\) km/s.

Figures 3–6 compare invertability of different migration methods. In all cases, constant-velocity modeling (demigration) was followed by migration with the correct velocity. Figures 3 and 4 show the results of modeling and migration with the Kirchhoff (Schneider, 1978) and \(f-k\) (Stolt, 1978) methods, respectively. These figures should be compared with Figure 5 showing the analogous result of the finite-difference velocity continuation. The comparison reveals a remarkable invertability of velocity continuation, which reconstructs accurately the main features and frequency content of the model. Since the forward operators were different for different migrations, this comparison did not test the migration properties themselves. For such a test, I compare the results of the Kirchhoff and velocity-continuation migrations after Stolt modeling. The result of velocity continuation, shown in Figure 6, is noticeably more accurate than that of the Kirchhoff method.
Figure 2: Synthetic model for testing finite-difference migration by velocity continuation.

Figure 3: Result of modeling and migration with the Kirchhoff method. Top left plot shows the reconstructed model. Top right plot compares the average amplitude spectrum of the true model with that of the reconstructed image. Bottom left is the reconstruction error. Bottom right is the absolute error in the spectrum.
Figure 4: Result of modeling and migration with the Stolt method. Top left plot shows the reconstructed model. Top right plot compares the average amplitude spectrum of the true model with that of the reconstructed image. Bottom left is the reconstruction error. Bottom right is the absolute error in the spectrum.
Figure 5: Result of modeling and migration with the finite-difference velocity continuation. Top left plot shows the reconstructed model. Top right plot compares the average amplitude spectrum of the true model with that of the reconstructed image. Bottom left is the reconstruction error. Bottom right is the absolute error in the spectrum. [vc2/vcmovd/vlvel]
Figure 6: (a) Modeling with Stolt method, migration with the Kirchhoff method. (b) Modeling with Stolt method, migration with the finite-difference velocity continuation. Left plots show the reconstructed models. Right plots show the reconstruction errors.
These tests confirm that finite-difference velocity continuation is an attractive migration method. It possesses remarkable invertability properties, which may be useful in applications that require inversion. While the traditional migration methods transform the data between two completely different domains (data-space and image-space), velocity continuation accomplishes the same transformation by propagating the data in the extended domain along the velocity direction. Inverse propagation restores the original data. According to [Li, 1986], the computational speed of this method compares favorably with that of Stolt migration. The advantage is apparent for cascaded migration or migration with multiple velocity models. In these cases, the cost of Stolt migration increases in direct proportion to the number of velocity models, while the cost of velocity continuation stays the same.

**Fourier approach**

The change of variable $\sigma = t^2$ transforms equation (1) to the form

$$2 \frac{\partial^2 P}{\partial v \partial \sigma} + v \frac{\partial^2 P}{\partial x^2} = 0,$$

whose coefficients do not depend on the time variables. Double Fourier transform in $\sigma$ and $x$ further simplifies equation (6) to the ordinary differential equation

$$2i\Omega \frac{d\hat{P}}{dv} - vk^2 \hat{P} = 0,$$

where the “frequency” variable $\Omega$ corresponds to the stretched time coordinate $\sigma$, and $k$ is the wavenumber in $x$:

$$\hat{P}(k, \Omega, v) = \int \int P(x, t, v) e^{-i\Omega t^2 + ikx} dx \, dt$$

Equation (7) has an explicit analytical solution

$$\hat{P}(k, \Omega, v) = \hat{P}_0(k, \Omega) e^{\frac{i k^2 (v_0^2 - v^2)}{4\Omega}},$$

which leads to a very simple algorithm for the numerical velocity continuation. The algorithm consists of the following steps:

1. Input the zero-offset (post-stack) data migrated with velocity $v_0$ (or unmigrated if $v_0 = 0$).
2. Transform the input from a regular grid in $t$ to a regular grid in $\sigma$.
3. Apply Fast Fourier Transform (FFT) in $x$ and $\sigma$.
4. Multiply by the all-pass phase-shift filter $e^{\frac{i k^2 (v_0^2 - v^2)}{4\Omega}}$. 
Figure 7: Synthetic seismic data before (left) and after (right) transformation to the \( \sigma \) grid.

5. Inverse FFT in \( x \) and \( \sigma \).

6. Inverse transform to a regular grid in \( t \).

Figure 7 shows a simple synthetic model of seismic reflection data generated from the model in Figure 2 before and after transforming the grid, regularly spaced in \( t \), to a grid, regular in \( \sigma \). The left plot of Figure 8 shows the Fourier transform of the data. Except for the nearly vertical event, which corresponds to a stack of parallel layers in the shallow part of the data, the data frequency range is contained near the origin in the \( \Omega-k \) space. The right plot of Figure 8 shows the phase-shift filter for continuation from zero imaging velocity (which corresponds to unprocessed data) to the velocity of 1 km/sec. The rapidly oscillating part (small frequencies and large wavenumbers) is exactly in the region, where the data spectrum is zero. It corresponds to physically impossible reflection events.

The described algorithm is very attractive from the practical point of view because of its efficiency (based on the FFT algorithm). The operation count is roughly the same as in the Stolt migration implemented with equation (4): two forward and inverse FFTs and forward and inverse grid transform with interpolation (one complex-number transform in the case of Stolt migration). The velocity continuation algorithm can be more efficient than the Stolt method because of the simpler structure of the innermost loop (step 4 in the algorithm).

**NUMERICAL VELOCITY CONTINUATION IN THE PRESTACK DOMAIN**

To generalize the algorithm of the previous section to the prestack case, it is first necessary to include the residual NMO term [Fomel 2003]. Residual normal moveout
can be formulated with the help of the differential equation:

\[
\frac{\partial P}{\partial v} + \frac{h^2}{v^3 t} \frac{\partial P}{\partial t} = 0 ,
\]

where \( h \) stands for the half-offset. The analytical solution of equation (10) has the form of the residual NMO operator:

\[
P(t, h, v) = P_0 \left( \sqrt{t^2 + h^2 \left( \frac{1}{v_0^2} - \frac{1}{v^2} \right)} , h \right).
\]

After transforming to the squared time \( \sigma = t^2 \) and the corresponding Fourier frequency \( \Omega \), equation (10) takes the form of the ordinary differential equation

\[
\frac{d\hat{P}}{dv} + i\Omega \frac{2h^2}{v^3} \hat{P} = 0
\]

with the analytical frequency-domain phase-shift solution

\[
\hat{P}(\Omega, h, v) = \hat{P}_0(\Omega, h)e^{i\Omega h^2 \left( \frac{1}{v_0^2} - \frac{1}{v^2} \right)}.
\]

To obtain a Fourier-domain prestack velocity continuation algorithm, one just needs to combine the phase-shift operators in equations (9) and (13) and to include stacking across different offsets. The exact velocity continuation theory also includes the residual DMO term (Fomel, 2003), which has a second-order effect, pronounced only at small depths. It is neglected here for simplicity. The algorithm takes the following form:

1. Input a set of common-offset images, migrated with velocity \( v_0 \).
2. Transform the time axis $t$ to the squared time coordinate: $\sigma = t^2$.

3. Apply a fast Fourier transform (FFT) on both the squared time and the midpoint axis. The squared time $\sigma$ transforms to the frequency $\Omega$, and the midpoint coordinate $x$ transforms to the wavenumber $k$.

4. Apply a phase-shift operator to transform to different velocities $v$:

$$\hat{P}(\Omega, k, v) = \sum_h \hat{P}_0(\Omega, k, h) e^{i\frac{\kappa^2(v_0 - v)^2}{4h} + i\Omega h^2 \left(\frac{1}{v_0^2} - \frac{1}{v^2}\right)}.$$  \hspace{1cm} (14)

To save memory, the continuation step is immediately followed by stacking. For velocity analysis purposes, a semblance measure ([Neidell and Taner, 1971]) is computed in addition to the simple stack analogously to the standard practice of stacking velocity analysis.

Implementing the residual moveout correction in the Fourier domain allows one to package it conveniently with the phase-shift operator without the need to transform the continuation result back to the time domain. The offset dimension in equation (14) is replaced by the velocity dimension similarly to the velocity transform of the conventional stacking velocity analysis ([Yilmaz, 2001]).

5. Apply an inverse FFT to transform from $\Omega$ and $k$ to $\sigma$ and $x$.

6. Apply an inverse time stretch to transform from $\sigma$ to $t$.

One can design similar algorithms by using the finite difference method. Although the finite-difference approach offers a faster continuation speed, the spectral algorithm has a higher accuracy while maintaining an acceptable cost.

Figure 9 shows impulse responses of prestack velocity continuation. The input for producing this figure was a time-migrated constant-offset section, corresponding to an offset of 1 km and a constant migration velocity of 1 km/s. In full accordance with the theory ([Fomel, 2003]), three spikes in the input section transformed into shifted ellipsoids after continuation to a higher velocity and into shifted hyperbolas after continuation to a smaller velocity. Padding of the time axis helps to avoid the wrap-around artifacts of the Fourier method. Alternatively, one could use the artifact-free but more expensive Chebyshev spectral method ([Fomel, 1998]).

Velocity continuation creates a time-midpoint-velocity cube (four-dimensional for 3-D data), which is convenient for picking imaging velocities in the same way as the result of common-midpoint or common-reflection-point velocity analysis. The important difference is that velocity continuation provides an optimal focusing of the reflection energy by properly taking into account both vertical and lateral movements of reflector images with changing migration velocity. An experimental evidence for this conclusion is provided in the examples section of this paper.

The next subsection discusses the velocity picking step in more detail.
Figure 9: Impulse responses of prestack velocity continuation. Left plot: continuation from 1 km/s to 1.5 km/s. Right plot: continuation from 1 km/s to 0.7 km/s. Both plots correspond to the offset of 1 km.

Velocity picking and slicing

After the velocity continuation process has created a time-midpoint-velocity cube, one can pick the best focusing velocity from that cube and create an optimally focused image by slicing through the cube. This step is common in other methods that involve velocity slicing (Shurtleff, 1984; Fowler, 1984; Mikulich and Hale, 1992). The algorithm described below has been also adopted by Sava (2000) for velocity analysis in wave-equation migration.

A simple automatic velocity picking algorithm follows from solving the following regularized least-squares system:

\[
\begin{align*}
W x & \approx W p \\
\epsilon D x & \approx 0
\end{align*}
\]

(15)

In the more standard notation, the solution \( x \) minimizes the least-squares objective function

\[
(x - p)^T W^2 (x - p) + \epsilon^2 x^T D^T D x
\]

(16)

Here \( p \) is the vector of blind maximum-semblance picks (possibly in a predefined fairway), \( x \) is the estimated velocity picks, \( W \) is the weighting operator with the weight corresponding to the semblance values at \( p \), \( \epsilon \) is the scalar regularization parameter, \( D \) is a roughening operator, and \( D^T \) is the adjoint operator. The first least-squares fitting goal in (15) states that the estimated velocity picks should match the measured picks where the semblance is high enough. The second fitting goal tries

\[
\text{Of course, this goal might be dangerous, if the original picks } p \text{ include regular noise (such as multiple reflections) with high semblance value (Toldi, 1985). For simplicity, and to preserve the linearity of the problem, I assume that this is not the case.} \]
to find the smoothest velocity function possible. The least-squares solution of problem (15) takes the form
\[ x = (W^2 + \epsilon^2 D^T D)^{-1} W^2 p. \] (17)

In the case of picking a one-dimensional velocity function from a single semblance panel, one can simplify the algorithm by choosing \( D \) to be a convolution with the derivative filter \((1, -1)\). It is easy to see that in this case the inverted matrix in formula (17) has a tridiagonal structure and therefore can be easily inverted with a linear-time algorithm. The regularization parameter \( \epsilon \) controls the amount of smoothing of the estimated velocity function. Figure 10 shows an example velocity spectrum and two automatic picks for different values of \( \epsilon \).

![Figure 10: Semblance panel (left) and automatic velocity picks for different values of the regularization parameter. Higher values of \( \epsilon \) lead to smoother velocities.](vc2/beivc/velpick)

In the case of picking two- or three-dimensional velocity functions, one could generalize problem (15) by defining \( D \) as a 2-D or 3-D roughening operator. I chose to use a more simplistic approach, which retains the one-dimensional structure of the algorithm. I transform system (15) to the form
\[
\begin{cases}
W x \approx W p \\
\epsilon D x \approx 0 \\
\lambda x \approx \lambda x_0
\end{cases},
\] (18)
where $\mathbf{x}$ is still one-dimensional, and $\mathbf{x}_0$ is the estimate from the previous midpoint location. The scalar parameter $\lambda$ controls the amount of lateral continuity in the estimated velocity function. The least-squares solution to system (18) takes the form

$$\mathbf{x} = (\mathbf{W}^T + \epsilon^2 \mathbf{D}^T \mathbf{D} + \lambda^2 \mathbf{I})^{-1} \left( \mathbf{W}^T \mathbf{p} + \lambda^2 \mathbf{x}_0 \right),$$

(19)

where $\mathbf{I}$ denotes the identity matrix. Formula (19) also reduces to an efficient tridiagonal matrix inversion.

After the velocity has been picked, an optimally focused image is constructed by slicing in the time-midpoint-velocity cube. I used simple linear interpolation for slicing between the velocity grid values. A more accurate interpolation technique can be easily adopted.

**EXAMPLES**

I demonstrate the performance of the method using a simple 2-D synthetic test and a field data example from the North Sea.

**Synthetic Test**

The synthetic test uses constant-velocity prestack modeling and migration to check the validity of the method when all the theoretical requirements are satisfied. The data were generated from the synthetic reflectivity model (Figure 2) and included 60 offsets ranging from 0 to 0.5 km. The exact velocity in the model is 1.5 km/s, and the initial velocity for starting the continuation process was chosen at 2 km/s.

Figure 11 compares the semblance panels for migration velocity analysis using velocity continuation and using the conventional (NMO) analysis. In the top part of the image, both panels show maximum picks at the correct velocity (1.5 km/s). The advantage of velocity continuation is immediately obvious in the deeper part of the image, where the events are noticeably better focused.

The final result of velocity continuation (after picking maximum semblance and slicing in the velocity cube) is shown in the bottom left plot of Figure 12. For comparison, Figure 12 also shows the result of migration with the correct velocity (the top left plot), initial velocity (the top right plot), and the result of velocity slicing after the simple NMO correction, corresponding to the conventional MVA (the bottom right plot). The same velocity picking and slicing program was used in both cases. The comparison clearly shows that, in this simple example, velocity continuation is able to accurately reproduce the correct image without using any prior information about the migration velocity and without any need for repeating the prestack migration procedure. Velocity continuation correctly images events with conflicting dips by properly taking into account both vertical and lateral shifts in the image position.
Figure 11: Semblance panels for migration velocity analysis at the common image point 1 km. Left: after velocity continuation. Right: after conventional (NMO) velocity analysis. In a structurally complex region, velocity continuation clearly provides better focusing. The correct velocity is 1.5 km/s.
Figure 12: Velocity continuation tested on the synthetic example. Top left: prestack migration with the correct velocity of 1.5 km/s. Top right: prestack migration with the velocity of 2 km/s. Bottom left: the result of velocity continuation. Bottom right: the result from picking migration image after only conventional NMO correction.
Field Data Example

Figure 13 compares the result of a constant-velocity prestack migration with the velocity of 2 km/s, applied to a dataset from the North Sea (courtesy of Elf Aquitaine) and the result of velocity continuation to the same velocity from a migration with a smaller velocity of 1.4 km/s (Figure 13a). The two images (Figures 13b and 13c) look remarkably similar, in full accordance with the theory.

Figure 13: Constant-offset section of the North Sea dataset after migration with the velocity of 1.4 km/s (a), migration with the velocity of 2 km/s (b), migration with the velocity of 1.4 km/s and velocity continuation to 2 km/s (c).

Figure 14 shows a result of two-dimensional velocity picking after velocity continuation. I used values of $\epsilon = 0.1$ and $\lambda = 0.1$. The first parameter controls the vertical smoothing of velocities, while the second parameter controls the amount of lateral continuity.

Figure 15 shows the final result of velocity continuation: an image, obtained by slicing through the velocity cube with the picked imaging velocities. The edges of the salt body in the middle of the section have been sharply focused by the velocity continuation process. To transform the already well focused image into the depth domain, one may proceed in a way similar to hybrid migration: demigration to zero-offset, followed by post-stack depth migration (Kim et al., 1997). This step would require constructing an interval velocity model from the picked imaging velocities.
Figure 14: Automatically picked migration velocity after velocity continuation.
Figure 15: Final result of velocity continuation: seismic image, obtained by slicing through the velocity cube.
CONCLUSIONS

Velocity continuation is a powerful method for time migration velocity analysis. The strength of this method follows from its ability to take into account both vertical and lateral movement of the reflection events in seismic images with the changes of migration velocity.

Efficient practical algorithms for velocity continuation can be constructed using either finite-difference or spectral methods. When applied in the post-stack (zero-offset) setting, velocity continuation can be used as a computationally attractive method of time migration. Both finite-difference and spectral approaches possess remarkable invertibility properties: continuation to a lower velocity reverses continuation to a higher velocity. For the finite-difference algorithm, this property is confirmed by synthetic tests. For the spectral algorithm, it follows from the fact that velocity continuation reduces to a simple phase-shift unitary operator.

Including velocity continuation in the practice of migration velocity analysis can improve the focusing power of time migration and reduce the production time by avoiding the need for iterative velocity refinement. No prior velocity model is required for this type of velocity analysis. This conclusion is confirmed by synthetic and field data examples.

ACKNOWLEDGMENTS

I thank Jon Claerbout, Biondo Biondi, and Bill Symes for useful and stimulating discussions, the sponsors of the Stanford Exploration Project for their financial support, and Elf Aquitaine for providing the data used in this work. I am also grateful to Paul Fowler, Samuel Gray, Hugh Geiger, and one anonymous reviewer for thorough and helpful reviews that improved the quality of the paper.
REFERENCES

———, 1990, Common-offset migrations and velocity analysis: First Break, 08, 224–234.
Sava, P., 2000, Prestack Stolt residual migration for migration velocity analysis: 70th


INTRODUCTION

Estimating an accurate velocity function is one of the most critical steps in building an accurate seismic depth image of the subsurface. In areas with significant structural complexity, one-dimensional updating schemes become unstable, and more robust algorithms are needed. Reflection tomography both in the premigrated (Bishop et al., 1985) and postmigrated domains (Stork, 1992; Kosloff et al., 1996) bring the powerful technologies of geophysical inversion theory to bear on the problem.

Unfortunately, however, inversion methods can be limited by the accuracy of their forward modeling operators, and most practical implementations of traveltime tomography are based on ray-theory, which assumes a high frequency wave, propagating through a smoothly varying velocity field, perhaps interrupted with a few discrete interfaces. Real world wave-propagation is much more complicated than this, and the failure of ray-based methods to adequately model wave propagation through complex media is fueling interest in “wave-equation” migration algorithms that both accurately model finite-frequency effects, and are practical for large 3-D datasets. As a direct consequence, finite-frequency velocity analysis and tomography algorithms are also becoming an important area of research (Woodward, 1992; Biondi and Sava, 1999).

Recent work in the global seismology community (Marquering et al., 1998, 1999) is drawing attention to a non-intuitive observation first made by Woodward (1992), that in the weak-scattering limit, finite-frequency traveltimes have zero-sensitivity to velocity perturbations along the geometric ray-path. This short-note aims to explore and explain this non-intuitive observation.

THEORY

A generic discrete linear inverse problem may be written as

\[ \mathbf{d} = \mathbf{A} \mathbf{m} \]  

where \( \mathbf{d} = (d_1 \ d_2 \ ...)^T \) is the known data vector, \( \mathbf{m} = (m_1 \ m_2 \ ...)^T \) is the unknown model vector, and \( \mathbf{A} \) represents the linear relationship between them. A natural question to ask is: which parts of the model influence a given observed data-point? The answer is that the row of matrix, \( \mathbf{A} \), corresponding to the data-point of interest will be non-zero where that point in model space influences the data-value. Rows of \( \mathbf{A} \) may therefore be thought of as sensitivity kernels, describing which points in model space are sensed by a given data-point.

For a generic linearized traveltime tomography problem, traveltime perturbations, \( \delta T \), are related to slowness perturbations, \( \delta S \), through a linear system,

\[ \delta T = \mathbf{A} \delta S. \]  

The form of the sensitivity kernels depend on the the modeling operator, \( \mathbf{A} \).
Under the ray-approximation, traveltime for a given ray, $T$, is calculated by integrating slowness along the ray-path,

$$ T = \int_{\text{ray}} s(x) \, dl. $$

(3)

Assuming that the ray-path is insensitive to a small slowness perturbation, the perturbation in traveltime is given by the path integral of the slowness perturbation along the ray,

$$ \delta T = \int_{\text{ray}} \delta s(x) \, dl. $$

(4)

Since traveltime perturbations given by equation (4) are insensitive to slowness perturbations anywhere off the geometric ray-path, the sensitivity kernel is identically zero everywhere in space, except along the ray-path where it is constant. The implication for ray-based traveltime tomography is that traveltime perturbations should be back-projected purely along the ray-path.

We are interested in more accurate tomographic systems of the form of equation (3), that model the effects of finite-frequency wave-propagation more accurately than simple ray-theory. Once we have such an operator, the first question to ask is: what do the rows look like?

**Born traveltime sensitivity**

One approach to building a linear finite-frequency traveltime operator is to apply the first-order Born approximation, to obtain a linear relationship between slowness perturbation, $\delta S$, and wavefield perturbation, $\delta U$,

$$ \delta U = B \delta S. $$

(5)

The Born operator, $B$, is a discrete implementation of equation (A-7), which is described in the Appendix.

Traveltime perturbations may then be calculated from the wavefield perturbation through a (linear) picking operator, $C$, such that

$$ \delta T = C \delta U = CB \delta S $$

(6)

where $C$ is a (linearized) picking operator, and a function of the background wavefield, $U_0$.

Cross-correlating the total wavefield, $U(t)$, with $U_0(t)$, provides a way of measuring their relative time-shift, $\delta T$. Marquering et al. (1999) uses this to provide the following explicit linear relationship between $\delta T$ and $\delta U(t)$,

$$ \delta T = \frac{\int_{t_1}^{t_2} \dot{U}(t) \, \delta U(t) \, dt}{\int_{t_1}^{t_2} \ddot{U}(t) \, U(t) \, dt}, $$

(7)
where dots denote differentiation with respect to \( t \), and \( t_1 \) and \( t_2 \) define a temporal window around the event of interest. Equation (7) is only valid for small time-shifts, \( \delta T \ll \lambda s_0 \).

**Rytov traveltime sensitivity**

The first Rytov approximation (or the phase-field linearization method, as it is also known) provides a linear relationship between the slowness and complex phase perturbations.

\[
\delta \Psi = R \delta S,
\]

where \( \Psi = \exp(U) \), and the Rytov operator, \( R \), is a discrete implementation of equation (A-10), which is also described in Appendix A.

Traveltime is related to the complex phase by the equation, \( \Im(\delta \psi) = \omega \delta t \). For a band-limited arrival with amplitude spectrum, \( F(\omega) \), traveltime perturbation can be calculated simply by summing over frequency (Woodward, 1992),

\[
\delta T = \sum_{\omega} \frac{F(\omega)}{\omega} \Im(\delta \Psi) = \sum_{\omega} \frac{F(\omega)}{\omega} \Im(R \delta S).
\]

Of the two approximations, several authors (Beydoun and Tarantola, 1988; Woodward, 1989) note that the Born approximation is the better choice for modeling reflected waves, while the Rytov approximation is better for transmitted waves. Differences tend to zero, however, as the scattering becomes small.

**KERNELS COMPARED**

This section contains images of traveltime kernels computed numerically for a simple model that may be encountered in a reflection tomography problem. The source is situated at the surface, and the receiver (known reflection point) is located at a depth of 1.8 km in the subsurface. The background velocity model, \( v_0(z) = 1/s_0(z) \), is a linear function of depth with \( v_0(0) = 1.5 \text{ km s}^{-1} \), and \( \frac{dv_0}{dz} = 0.8 \text{ s}^{-1} \). I chose a linear velocity function since Green’s functions can be computed on-the-fly with rapid two-point ray-tracing.

Figure 1 shows the ray-theoretical traveltime sensitivity kernel: zero except along the geometric ray-path.

Figures 2 and 3 show first Rytov traveltime sensitivity kernels for 30 Hz and 120 Hz wavelets respectively. The important features of these kernels are identical to the features of kernels that Marquering et al. (1999) obtained for teleseismic \( S-H \) wave scattering, and to Woodward’s band-limited wave-paths (Woodward, 1992). They have the appearance of a hollow banana: that is appearing as a banana if visualized in the plane of propagation, but as a doughnut on a cross-section perpendicular to the
Figure 1: Traveltime sensitivity kernel for ray-based tomography in a linear $v(z)$ model. The kernel is zero everywhere except along geometric ray-path. Right panel shows a cross-section at $X = 1$ km.

ray. Somewhat counter-intuitively, this suggests that traveltimes have zero sensitivity to small velocity perturbations along the geometric raypath. Fortunately, however, as the frequency of the seismic wavelet increases, the bananas become thinner, and approach the ray-theoretical kernels in the high-frequency limit. Parenthetically, it is also worth noticing that the width of the bananas increases with depth as the velocity (and seismic wavelength) increases.

I do not show the first-Born kernels here, since, in appearance, they are identical to the Rytov kernels shown in Figures 2 and 3.

THE BANANA-DOUGHNUT PARADOX

The important paradox is not the apparent contradiction between ray-theoretical and finite-frequency sensitivity kernels, since they are compatible in the high-frequency limit. Instead, the paradox is how do you reconcile the zero-sensitivity along the ray-path with your intuitive understanding of wave propagation?

A first potential resolution to the paradox is that the wavefront healing removes any effects of a slowness perturbation. This alone is a somewhat unsatisfactory explanation since it does not explain why traveltimes are sensitive to slowness perturbations just off the geometric ray-path.

A second potential resolution is that the hollowness of the banana is simply an artifact of modeling procedure. This is partially true. Both Born and Rytov are single scattering approximations, and a single scatterer located on the geometric ray-path may only contribute energy in-phase with the direct arrival. In contrast, if there are two scatterers on the geometric ray-path traveltimes may be affected. However, just
Figure 2: Rytov traveltime sensitivity kernel for 30 Hz wavelet in a linear $v(z)$ model. The kernel is zero along geometric ray-path. Right panel shows a cross-section at $X = 1$ km.

Figure 3: Rytov traveltime sensitivity kernel for 120 Hz wavelet in a linear $v(z)$ model. The kernel is zero along geometric ray-path. Right panel shows a cross-section at $X = 1$ km.
because the paradox may appear to be an artifact of the modeling procedure does not mean it is not a real phenomenon. In the weak scattering limit, traveltimes will indeed be insensitive to a slowness perturbation situated on the geometric ray-path.

CONCLUSIONS: DOES IT MATTER?

Practitioners of traveltime tomography typically understand the shortcomings of ray-theory; although they realize using “fat-rays” would be better, they smooth the slowness model both explicitly and by regularizing during the inversion procedure. In practice, any shortcomings of traveltime tomography are unlikely to be caused by whether or not the fat-rays are hollow.

However, the null space of seismic tomography problems is typically huge. Smoothing and regularization are often done with very ad hoc procedures. Understanding the effects of finite-frequency through sensitivity kernels may lead to incorporating more physics during the regularization and improve tomography results.

REFERENCES

Beydoun, W. B., and A. Tarantola, 1988, First Born and Rytov approximations:
———, 1999, Three-dimensional sensitivity kernels for finite-frequency traveltimes:
Morse, P. M., and H. Feshbach, 1953, Methods of theoretical physics: McGraw-Hill.
APPENDIX A

BORN/RYTOV REVIEW

Modeling with the first-order Born (and Rytov) approximations [e.g. Beydoun and Tarantola (1988)] can be justified by the assumption that slowness heterogeneity in the earth exists on two separate scales: a smoothly-varying background, $s_0$, within which the ray-approximation is valid, and weak higher-frequency perturbations, $\delta s$, that act to scatter the wavefield. The total slowness is given by the sum,

$$s(x) = s_0(x) + \delta s(x). \quad (A-1)$$

Similarly, the total wavefield, $U$, can be considered as the sum of a background wavefield, $U_0$, and a scattered field, $\delta U$, so that

$$U(x, \omega) = U_0(x, \omega) + \delta U(x, \omega), \quad (A-2)$$

where $U_0$ satisfies the Helmholtz equation in the background medium,

$$[\nabla^2 + \omega^2 s_0^2(x)] U_0(x, \omega) = 0, \quad (A-3)$$

and the scattered wavefield is given by the (exact) non-linear integral equation (Morse and Feshbach 1953),

$$\delta U(x, \omega) = \frac{\omega^2}{4\pi} \int_V G_0(x, \omega; x') U(x, \omega; x') \delta s(x') dV(x'). \quad (A-4)$$

In equation (A-4), $G_0$ is the Green’s function for the Helmholtz equation in the background medium: i.e. it is a solution of the equation

$$[\nabla^2 + \omega^2 s_0^2(x)] G_0(x, \omega; x_s) = -4\pi \delta(x - x_s). \quad (A-5)$$

Since the background medium is smooth, in this paper I use Green’s functions of the form,

$$G_0(x, \omega; x_s) = A_0(x, x_s) e^{i\omega T_0(x, x_s)}. \quad (A-6)$$

where $A_0$ and $T_0$ are ray-traced traveltimes and amplitudes respectively.

A Taylor series expansion of $U$ about $U_0$ for small $\delta s$, results in the infinite Born series, which is a Neumann series solution (Arfken 1985) to equation (A-4). The first term in the expansion is given below: it corresponds to the component of wavefield that interacts with scatters only once.

$$\delta U_{\text{Born}}(x, \omega) = \frac{\omega^2}{4\pi} \int_V G_0(x, \omega; x') U_0(x, \omega; x') \delta s(x') dV(x'). \quad (A-7)$$

The approximation implied by equation (A-7) is known as the first-order Born approximation. It provides a linear relationship between $\delta U$ and $\delta s$, and it can be computed more easily than the full solution to equation (A-4).
The Rytov formalism starts by assuming the heterogeneity perturbs the phase of the scattered wavefield. The total field, \( U = \exp(\psi) \), is therefore given by

\[
U(\mathbf{x}, \omega) = U_0(\mathbf{x}, \omega) \exp(\delta \psi) = \exp(\psi_0 + \delta \psi). \tag{A-8}
\]

The linearization based on small \( \delta \psi / \psi \) leads to the infinite Rytov series, on which the first term is given by

\[
\delta \psi_{\text{Rytov}}(\mathbf{x}, \omega) = \frac{\delta U_{\text{Born}}(\mathbf{x}, \omega)}{U_0(\mathbf{x}, \omega)} = \frac{\omega^2}{4\pi U_0(\mathbf{x}, \omega)} \int_V G_0(\mathbf{x}, \omega; \mathbf{x}') U_0(\mathbf{x}, \omega; \mathbf{x}') \delta s(\mathbf{x}') dV(\mathbf{x}'). \tag{A-10}
\]

The approximation implied by equation (A-10) is known as the first-order Rytov approximation. It provides a linear relationship between \( \delta \psi \) and \( \delta s \).
Modeling 3-D anisotropic fractal media

Nizar Chemingui

ABSTRACT

This paper presents stochastic descriptions of anisotropic fractal media. Second order statistics are used to represent the continuous random field as a stationary zero-mean process completely specified by its two-point covariance function. In analogy to the two-dimensional Goff and Jordan model for seafloor morphology, I present the von Karman functions as a generalization to media with exponential correlation functions. I also compute a two-state model by mapping the random field from continuous realizations to a binary field. The method can find application in modeling impedances from fractal media and in fluid flow problems.

INTRODUCTION

Our understanding of the physical phenomena occurring in the earth always involves the study of the medium itself. Unfortunately, the earth offers an unusually complicated medium in which heterogeneities are observed at every scale. Sometimes the problem is too difficult to deal with deterministically but it turns out to be quite simply treated by statistical methods. Solutions to the one-dimensional problem have targeted the study of reflectivity series obtained from well logs. Seismic impedance can be modeled as a special type of Markov chain, one which is constrained to have a purely exponential correlation function (Godfrey et al., 1980). The two-dimensional problem has gained a lot of attention in the recent years from studies of seismic scattering in heterogeneous media, e.g., (Wu and Aki, 1985; Frankel and Clayton, 1986; Goff and Jordan, 1988; Holliger and Levander, 1992; Holliger et al., 1993). Three dimensional simulations can be used in fluid flow experiments (Popovici and Muir, 1989).

This paper presents a method for simulating three-dimensional anisotropic random fields using second order-statistics. The method was introduced by Goff and Jordan (1988) to model a two-dimensional seafloor morphology. I have considered the cases of random media characterized by Gaussian, exponential and von Karman correlation functions. I use the von Karman functions as a generalizations to the exponential correlation functions in modeling random sequences. This type of correlation function was first introduced by von Karman (1948) for characterizing the random velocity field.
of a turbulent medium. It has since been frequently used in the statistical literature, studies of turbulence problems, e.g. (Tatarski, 1961), and studies of random media such as wave scattering, e.g. (Chernov, 1960). The von Karman functions were identified specifically as belonging to the class of continuous correlation functions (Matern, 1970). Holliger et al. (1993) used von Karman covariance functions to model binary fields and defined “binarization” as a mapping of all values in a continuous field to just two values of the new field. I have employed their technique to model two state models (i.e., rock/pore or sandstone/shale) from continuous realizations and test the increase in medium roughness through the “binarization” process.

RANDOM FIELDS

A stochastic model is constructed for the properties of the random medium. We first construct a distribution function $P(x)$ for the properties of the medium $h(x)$. From such a probability function, we can recover the statistical properties of the distribution (i.e., mean, variance, etc.) through its N-point statistical moments (Goff and Jordan, 1988).

$$C_N(x_1, x_2, ..., x_N) = \langle h(x_1) ... h(x_N) \rangle$$

$$= \int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} h_1 ... h_N P(h_1, ..., h_N) dh_1 ... dh_N$$  \hspace{1cm} (1)

where $h_N = h(x_N)$. The key assumption of spatial homogeneity (stationarity) means that the N-point moments are taken to depend only on the vector joining these points and not on their absolute positions. These moments describe the magnitude and smoothness of the fluctuations of $h(x)$.

Second-order Statistics

I restricted this research to the study of second-order statistics of random fields. This means the study of random media characterized by Gaussian distributions, where a Gaussian process is completely specified by its first- and second-order moments. Furthermore if I define the field $h(x)$ to be a zero mean process:

$$\langle h(x) \rangle = \int_{-\infty}^{+\infty} h(x) P(h(x)) dh = 0$$ \hspace{1cm} (2)

then $h(x)$ is fully described by its two-point moment that is its autocovariance function which we write as a function of the correlation function:

$$C_{hh}(r) = E[h(x)h(x + r)] = H^2 \rho_{hh}(r)$$ \hspace{1cm} (3)

where $P(h(x))$ is the probability density function of $h(x)$, $r$ is the lag vector, $E$ is the expected value, $H^2$ is the variance (i.e. $C_{hh}(0)$) and $\rho_{hh}$ is the three-dimensional
correlation function. Equation (3) shows that the random medium can be adequately specified by its autocorrelation function. More generally, an anisotropic random field can be described by a monotonically decaying autocorrelation function whose rate of decay depends on direction. The roughness of the medium is function of the decay rate of the correlation. The Fourier transform of the autocorrelation is the power spectrum of the field (Bracewell, 1978). Three types of correlation functions were commonly used in the field of seismic modeling: the Gaussian, exponential and von Karman functions. These special functions are described by analytic expressions of their autocorrelations and Fourier transforms.

Two-dimensional cases have been studied for some time (Wu and Aki, 1985; Frankel and Clayton, 1986; Goff and Jordan, 1988; Holliger and Levander, 1992; Holliger et al., 1993). Within the last several years, computer capacity and speed have grown rapidly. It is now feasible to extend our models and simulations to the three-dimensional case.

**von Karman correlation functions**

The three-dimensional anisotropic von Karman function is given by (Goff and Jordan, 1988):

\[
C(r) = \frac{4\pi \nu H^2 r^\nu K_\nu(r)}{K_\nu(0)}
\]

and its three-dimensional Fourier transform is:

\[
P(k) = \frac{4\pi \nu H^2 a_x^2 + a_y^2 + a_z^2}{K_\nu(0) (1 + k^2)^{\nu + \frac{3}{2}}}
\]

where \( r = \sqrt{\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2} + \frac{z^2}{a_z^2}} \), \( k = \sqrt{k_x^2 a_x^2 + k_y^2 a_y^2 + k_z^2 a_z^2} \); \( a_x, a_y \) and \( a_z \) are the characteristic scales of the medium along the 3-dimensions and \( k_x, k_y \) and \( k_z \) are the wavenumber components. \( K_\nu \) is the modified Bessel function of order \( \nu \), where \( 0.0 < \nu < 1.0 \) is the Hurst number (Mandelbrot, 1983, 1985). The fractal dimension of a stochastic field characterized by a von Karman autocorrelation is given by:

\[
D = E + 1 - \nu
\]

where \( E \) is the Euclidean dimension i.e., \( E = 3 \) for the three-dimensional problem. The special case of \( \nu = 0.5 \) yields to the exponential covariance function that corresponds to a Markov process (Feller, 1971).

\[
C(r) = H^2 e^{-r}
\]

whose three-dimensional Fourier transform is given by:

\[
P(k) = H^2 \frac{a_x^2 + a_y^2 + a_z^2}{(1 + k^2)^2}
\]

Decreasing the Hurst number, \( \nu \), increases the roughness of the medium. The lim-
Figure 1: Comparison of 1-dimensional isotropic von Karman autocorrelation functions for varying hurst number, $\nu$.

The cases of unity and zero correspond to a smooth Euclidean random field and a space-filling field respectively.

Figure 1 shows the one-dimensional isotropic von Karman correlation function plotted for different values of $\nu$. The functions have exponential behavior but different decay rates. The higher the slope, the rougher the medium (i.e., the lower is $\nu$). The exponential behavior is explained by the modified Bessel functions $K_\nu(x)$ which in the region $x \gg \nu$ behave as

$$K_\nu(x) \approx \frac{1}{\sqrt{2\pi x}} \exp(x)$$  \hspace{1cm} (9)

For comparison of the results, I also include the anisotropic Gaussian autocovariance function, which in 3-D has the familiar form:

$$C(r) = H^2 e^{-r^2}$$  \hspace{1cm} (10)

and its 3-dimensional Fourier transform is given by:

$$P(k) = \frac{a_x a_y a_z}{2} H^2 e^{k^2}$$  \hspace{1cm} (11)
FORWARD MODELING

Continuous random fields have frequently been used for statistical analyses, modeling perturbed media, scattering and diffraction studies, fluid flow simulations and, other related problems. Numerical realizations may describe the statistical character of random models at all scales.

Numerical Implementation

The generation of synthetic random media is done in the wave number domain. First, we compute the power spectrum of the field, i.e., the Fourier spectrum of the autocorrelation function. Then we compute the Fourier spectrum by multiplying the square root of the power spectrum by a random phase factor \( e^{2\pi \eta} \) where \( \eta \) is a uniform deviate that lies in the interval \([0,1)\). In a final step we apply an inverse fast Fourier transform to obtain the spatial domain representation of the random medium. The numerical implementation of the method is very straightforward, although special care is required to handle D.C. and Nyquist wavenumbers. Algorithms are similar for the one-, two- and three-dimensional problem although if we do not care about computer expenses, 1- and 2-D random sequences can be simply extracted as arrays or sections from 3-D simulations.

![Synthetic random field with anisotropic Gaussian autocorrelation function; \( a_x = 15, a_y = 25, a_z = 35 \).](fractal/random/gauss)
Figure 3: Synthetic random field with anisotropic exponential autocorrelation function; $a_x = 15$, $a_y = 25$, $a_z = 35$. \texttt{fractal/random/expo}

Figure 4: Synthetic random field with anisotropic von Karman autocorrelation function; $a_x = 15$, $a_y = 25$, $a_z = 35$. \texttt{fractal/random/fractal}
Modeling 3-D random media

I show three different realizations of an anisotropic model with different aspect ratios along the three coordinate axes. The model is a 64 points cube with characteristic scales $a_x = 15$, $a_y = 25$, and $a_z = 35$. The media are characterized by Gaussian, exponential, and van karman autocorrelation functions, respectively. We notice the increase in model roughness as we move from the Gaussian medium to the exponential field (i.e., $\nu = .5$) and then to the von Karman field with $\nu = .2$.

In the physical world, these fields may represent media at different scales varying from the microscopic to the megascopic.

Modeling seismic impedances

Seismic impedances have frequently been modeled as a Markov process. Godfrey et. al. (1980) modeled impedance as a special type of Markov chain, one that is constrained to have a purely exponential correlation function. They tested their method on three actual logs and compared the autocorrelation function to a best fit exponential curve. Apart from a small geologic noise component at the origin, their results showed excellent agreement between the theoretical exponential and the actual autocorrelation on two of the well logs they considered. For large lags, the actual correlation function had exponential behavior similar to that of the theoretical curve, but all data points fell below the synthetic curve showing a faster decay rate. The behavior of the autocorrelation could very well be interpreted as related to a rougher distribution than that predicted by the exponential correlation. A von Karman correlation function with a Hurst number smaller than 0.5 would have given a better fit to the autocorrelation of the impedance series. The autocorrelation of the impedance function provides information on the depositional pattern in the sedimentary column i.e, cyclic or transitional (O’Doherty and Anstey, 1971).

Figure 5 shows a comparison of one-dimensional random sequences that simulate synthetic impedances with von Karman correlation functions of varying fractal dimensions (i.e., Hurst number $\nu$). Again the smaller the value of $\nu$, the rougher the sequence. The impedance with exponential correlation seems smooth compared to the ones generated from autocorrelation functions with values of $\nu$ lower than 0.5.

GENERATING TWO-STATE MODELS

In the geophysical world we often deal with heterogeneous media whose inhomogeneities are caused by the presence of two different types of material with different mechanical properties. A typical example is the case of a stratified formation of shale embedded in sandstone. In fluid flow and reservoir engineering problems, the rock samples are generally composed of a matrix and pore space. Continuously random fields are therefore inadequate to describe randomness in similar settings. I seek to
describe a random field in which the medium can be represented as a two-state model. This new field is called a binary field and the process of deriving the binary field from the continuous field is called “binarization” (Holliger et al., 1993). The problem is to relate the statistics of the binary field to those of the continuous field. Holliger et al. (1993) gave a brief description of their mapped two-dimensional binary field which I apply in a straightforward generalization to the three-dimensional problem.

To illustrate the effects of “binarizing” a continuous field, let’s consider two examples of random fields with Gaussian and exponential autocorrelation functions, respectively. In the first example I simulate a randomly-stratified medium. The second example is a realization of a random medium with statistically isotropic homogeneous inclusions. I like to analyze the change in the medium properties by comparing the autocorrelation function of the distribution before and after “binarization”. For better observation, I limit the analysis to the study of the correlation function along one axis, i.e., in the x-direction.

Figure ?? shows the averaged 1-D correlation function along the x-axis for the randomly layered medium. The solid curve displays the autocorrelation of the continuous field; the dashed one represents the autocorrelation of the “binary” field. The two functions are noticeably different from one another; the slope near the origin is greatly increased after “binarization” indicating a rougher distribution compared to the continuous case. Figure ?? shows the same observations for the isotropic random field with Gaussian autocorrelation; again the roughness of the field has increased as
indicated by the steepening in the slope of the autocorrelation.

Figure 6: Synthetic continuous random field with apparent layering and Gaussian autocorrelation; \(a_x = 5, a_y = 80, a_z = 80\).

CONCLUSIONS

In this initial study I have tackled the forward problem for modeling anisotropic fractal media using second-order statistics. The method has close analogy with the two-dimensional Goff and Jordan model for seafloor morphology. The generation of synthetic models is done in the Fourier domain and the algorithms are similar for the one- two- and three-dimensional problems. The von Karman functions are presented as a generalization of the exponential correlation function associated with the Markov process in modeling seismic impedances. The von Karman functions can be used for better description of statistic lithology of stratigraphic columns and understanding their depositional pattern. I have also computed a two-state model (i.e., rock/pore or sandstone/shale) by mapping the random field from continuous realizations into a binary field. Comparisons of the autocorrelation functions of the continuous and binary fields show that the fractal dimension (i.e, the roughness of the medium) increases through the “binarization” process.

FUTURE WORK

Future goals of this effort will be to formulate the inverse problem for estimating the characteristic parameters of the anisotropic fractal medium, i.e, aspect ratios of anisotropy, and Hausdorff (fractal) dimension. The technique of deriving the binary
Figure 7: Synthetic binary field derived from the continuous realization of a layered random field with Gaussian autocorrelation.

Figure 8: Synthetic continuous random field with isotropic Gaussian autocorrelation function; $a_x = 15$, $a_y = 15$, $a_z = 15$.  

Figure 9: Synthetic binary field derived from the continuous realization of a random field with Gaussian autocorrelation.  

Figure 10: Autocorrelation functions of the continuous (solid lines) and binary (dashed lines) fields for the layered random medium with exponential correlation.  

Figure 11: Autocorrelation functions of the continuous (solid lines) and binary (dashed lines) fields for the isotropic random medium with Gaussian correlation field from the continuous random field should be extended to simulate M-state models, where M is the number of states or rocks composing an impedance well-log.

I also need to test the method on actual well-log data and demonstrate a better fit with von Karman correlation functions compared to the exponential fit. This would be the first application of the inverse problem. Two and three dimensional problems can find application in the field of wave scattering and diffraction and in fluid flow problems.

APPENDIX A

VON KARMAN COVARIANCE FUNCTION

Equation (4) in the text represents the autocovariance of a random medium of fractal nature. The power spectrum of the field corresponds to the Fourier transform of its covariance function:

\[ P(k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C(r)e^{-ik\cdot r} d^3r \]  \hspace{1cm} (A-1)

Using the N-dimensional Hankel transform \([\text{Lord} \, 1954]\), the covariance function and its Fourier transform can be related as follows:

\[ p(k) = (2\pi)^{N/2}k^{-N/2+1} \int_0^{\infty} r^{N/2}J_{N/2-1}(rk)C(r)dr \]  \hspace{1cm} (A-2)
\[ C(r) = (2\pi)^{-N/2}r^{-N/2+1} \int_0^\infty k^{N/2}J_{N/2-1}(rk)P(k)dk \]  
(A-3)

where \( J_{N/2-1} \) is the Bessel function of order \( N/2 - 1 \).

The covariance \( C(r) \) in (3) is specified in terms of the function:

\[ G_\nu(r) = r^\nu K_\nu(r) \quad 0 \leq r < \infty \quad \nu \in [0,1] \]  
(A-4)

whose Hankel transform pair has been derived by Lord (1954):

\[ P(k) = \frac{\Gamma(\nu + N/2)}{2^{1-N-\nu}\pi^{N/2}(1 + k^2)^{-\nu-N/2}} \]  
(A-5)

where \( \Gamma \) is the gamma function defined as:

\[ \Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt \]  
(A-6)

Finally if we normalize \( G_\nu(r) \) by \( G_\nu(0) \) as in Goff and Jordan (1988), we obtain, for the three-dimensional case, the power spectrum of the field whose covariance is defined by (4):

\[ P(k) = 4\pi\nu H^2(1 + k^2)^{-\nu-3/2} \]  
(A-7)

REFERENCES


Seismic AVO analysis of methane hydrate structures

Christine Ecker and David E. Lumley

ABSTRACT

Marine seismic data from the Blake Outer Ridge offshore Florida show strong “bottom simulating reflections” (BSR) associated with methane hydrate occurrence in deep marine sediments. We use a detailed amplitude versus offset (AVO) analysis of these data to explore the validity of models which might explain the origin of the bottom simulating reflector. After careful preprocessing steps, we determine a BSR model which can successfully reproduce the observed AVO responses. The P- and S-velocity behavior predicted by the forward modeling is further investigated by estimating the P- and S-impedance contrasts at all subsurface positions. Our results indicate that the Blake Outer Ridge BSR is compatible with a model of methane hydrate in sediment, overlaying a layer of free methane gas-saturated sediment. The hydrate-bearing sediments seem to be characterized by a high P-wave velocity of approximately 2.5 km/s, an anomalously low S-wave velocity of approximately 0.5 km/s, and a thickness of around 190 meters. The underlaying gas-saturated sediments have a P-wave velocity of 1.6 km/s, an S-wave velocity of 1.1 km/s, and a thickness of approximately 250 meters.

INTRODUCTION

Bottom simulating reflectors, so called BSRs, parallel the seafloor at subbottom depths of several hundred meters. Seismic investigations [Shipley et al. 1979, Miller et al. 1991, Hyndman and Spence 1992] indicate that they are characterized by large negative reflection coefficients and increasing subbottom depth with increasing water depth. The base of the stability field for methane hydrates appears to be associated with these bottom simulating reflectors. Due to the enormous amount of methane that is apparently contained within hydrate structures, they are likely to have a significant “greenhouse” effect on future global climate, and might also represent an important future energy resource [Kvenvolden 1993]. Therefore, a good understanding of the origin and characteristics of the hydrate zones and BSRs is desirable. Only limited information is available from deep-sea drilling, as the risk of heating and destabilizing the initial hydrate conditions during the process of drilling is considerably high. Thus, the core samples and well-logs do not necessarily reflect

*e-mail: james@sep.stanford.edu
the correct in situ hydrate characteristics and properties. Consequently, most information is inferred remotely from seismic reflection data (Shipley et al., 1979; Miller et al., 1991; Hyndman and Davis, 1992; Hyndman and Spence, 1992; Singh et al., 1993). Most of these investigations, which were based mainly on AVO responses and synthetic modeling, used primarily P-velocity information, accessible directly from the seismic data, and neglected possible important S-velocity effects entirely. The exact formation of the hydrate and its formation are still controversial, and different models have been proposed to explain the origin of the BSRs (Kvenvolden and Barnard, 1983a; Hyndman and Davis, 1992).

In this study, we use both P- and S-wave information inferred from synthetic modeling, velocity and AVO analysis of marine data from the Blake Outer Ridge to explain the bottom simulating reflector. The validity of the different BSR models is explored and the effects of two proposed models can be clearly discriminated. The reflection amplitude variation with offset can be an important indicator of free gas at an interface (Shuey, 1985) and, together with the estimation of material properties at the interface, considerably limits the possible explanations of the physical origin of the BSR.

This paper discusses our work with preprocessing, modeling, inversion, and interpretation of the methane hydrate seismic data from the Blake Outer Ridge. Preliminary results of this study were presented by Ecker and Lumley (1993a,b). After careful preprocessing, including source wavelet deconvolution, trace interpolation, and amplitude and moveout traveltime calibration, a detailed velocity analysis is performed on the resulting CMP gathers. The estimated interval velocities are used to constrain a BSR model which can successfully reproduce the observed AVO amplitude responses. The impedance structure predicted by the modeled data is further reinforced by estimating the P- and S-impedance contrasts at all subsurface positions. Combining the results of the synthetic modeling and the impedance inversion, we give an integrated geophysical interpretation of the data.

METHANE HYDRATES AND BSR MODELS

A methane hydrate is an ice-like, crystalline lattice of water molecules in which gas molecules are trapped physically without the aid of direct chemical bonds. They are stable under certain pressure and temperature conditions (Figure 1). Thus, the occurrence of bottom simulating reflectors is restricted to two distinct regions: deep oceanic and polar. In deep oceanic regions, BSRs are found in outer continental margins of slopes and rises where cold bottom water is present. In polar regions, the BSRs are normally associated with permafrost, both onshore in continental sediments and offshore in sediments of the continental shelves.

Two models have been proposed to account for gas hydrate formation and thus the development of BSRs. The first one assumes the local generation of methane from organic material at the depth of the hydrate. Gradually thickening and thus
Figure 1: Phase diagram showing the boundary between methane gas and methane hydrate. Redrawn after Kvenvolden (1993).

deepening of the methane hydrate zone causes it eventually to subside into a temperature region where hydrate is unstable. Consequently, free gas can be present in this region (Kvenvolden and Barnard, 1983a). The BSR is caused by the impedance contrast at the base of the hydrate layer and the top of the gas layer. A second model, on the other hand, supports the formation of methane hydrates through the removal of methane from rising pore fluids being expelled upwards from deeper in the sediment section (Hyndman and Davis, 1992). Most of the methane is generated microbially at depths below the hydrate stability zone but not at depths sufficient for the formation of thermogenic methane. Thus free gas does not have to be present below the BSR. In this case, the BSR can be the consequence of the impedance contrast between overlaying sediments containing substantial amounts of high-velocity hydrate and underlaying normal velocity brine sediments.

**PREPROCESSING STEPS**

In the first preprocessing step, the data were corrected for time-varying spherical divergence. Next, we performed a single-trace source wavelet deconvolution in order to regularize the source wavelet with offset. The deconvolved data were then bandpass filtered to the original data bandwidth to remove spurious deconvolutional high-frequency noise. Using an initial semblance estimate of the best stacking velocity function, a normal moveout correction of the data was carried out.
Two main assumptions were made to perform the amplitude calibration. First, it was assumed that an offset-dependent rather than an angle-dependent amplitude correction was sufficient, since the difference between the maximal angle of incidence at the seafloor (30°) and the BSR reflection (33°) is negligibly small. Second, we assumed a functional form for the AVO response of the seafloor reflection, based on the fluid-solid interface Zoeppritz PP reflection coefficient. Based on these assumptions, the amplitude calibration was performed by scaling each trace to the seabottom amplitude to match the predicted seafloor AVO as a function of offset.

Due to the use of a nonlinear streamer to record the data, a trace interpolation of the near offset data was necessary to regularize the receiver cable group spacings. After applying an inverse NMO correction to the interpolated data, a high resolution NMO stacking velocity analysis was performed. Having derived a good stacking model for the data, they were reprocessed in a second iteration using the new velocities. Since it is essential for the subsequent impedance contrast estimation that the reflector moveout is very flat after NMO correction or migration, an additional static shift was applied preceding the amplitude scaling to correct for some small non-hyperbolic, offset-dependent residual moveout in the CMP gathers.

Figure 2 shows the final data after preprocessing. The gather contains a BSR AVO effect that is representative of the average trend along the entire line. Picking the peak amplitudes along the BSR yielded the AVO curve shown in Figure 3. Starting with a negative zero-offset reflection coefficient that was obtained by assuming a seafloor reflection of approximately 0.2, the amplitudes become increasingly negative with increasing offset.

MODELING APPROACH

Using the estimated interval velocities, the effects of different impedance structures on the BSR AVO response were explored in an attempt to reproduce the seismic data. Several models were constructed which were constrained to preserve the average interval velocity of each macro layer. To avoid possible tuning effects in this first, basic modeling approach, all layers were assumed to be thicker than a quarter of a wavelength. Synthetic AVO amplitude responses were then estimated for the individual models using Zoeppritz equations and compared with the amplitude responses observed in the seismic data.

Figure 4 shows the initial P- and S-wave interval velocities. The P-wave velocity was inferred directly from the seismic data, while the S-wave velocity was determined by assuming a Poisson’s ratio of 0.4 which is consistent with brine saturated sediments.

A considerably high P-wave velocity of approximately 2.5 km/s was obtained above the BSR which appears to be underlain by a lower velocity of around 1.6 km/s. The P-velocity trend for normal brine-saturated sediments is indicated by the dotted line. Being higher than this trend above the BSR and lower below it, the measured P-wave velocities might be compatible with a model of hydrate-bearing sediment.
Figure 2: Final CMP gather after preprocessing containing an average AVO effect observed along the line.

Figure 3: AVO curve of the amplitude picks along the BSR. The near offset reflection amplitude was determined by assuming a seafloor reflection of approximately 0.2.
overlaying gas-saturated sediments.

In a first attempt to model the observed AVO amplitudes, a thin layer of high-velocity, hydrate-bearing sediments was assumed to overlay brine-saturated sediments. As the measured P-wave interval velocity of 2.5 km/s above the BSR has to be preserved, the hydrate layer can not be smaller than a certain thickness in order to obtain realistic velocity values for this model. The modeled P- and S-wave velocities above and below the BSR are shown in Figure 5. The initial model is given by the dotted line whenever the modeled velocities differ from the initial ones. The S-wave velocity was obtained by assuming that the Poisson’s ratio in the hydrate-bearing sediments is comparable with that of brine sediments.

Using Zoeppritz equations, the AVO trend corresponding to the modeled velocities at the transition from hydrate to brine-saturated sediments is determined and compared with the one observed in the data (Figure 6).

A comparison of both curves yields that the thin-hydrate model not only failed in reproducing the near offset reflection coefficients, but also the general AVO trend, having slightly increasing amplitudes with increasing offset. Assuming negligibly small density contributions, the near offset amplitudes are mainly dependent on the P-wave velocity contrast at the reflector, while the AVO trend is characterized primarily by the S-wave velocity contrast. Thus, the AVO response resulting from the thin-hydrate model implies the use of both incorrect P- and S-wave velocities at the bottom simulating reflector. Further thinning of the hydrate layer would increase the P-wave velocity in this zone due to the required preservation of the measured interval velocity, and thus the P-wave velocity contrast at the BSR. This would result in an even
Figure 5: Interval velocities above and below the BSR for a thin-hydrate layer overlying brine sediment. The dotted lines represent the velocities of the initial model, while the solid lines give the velocities of this model. The arrows indicate the direction of the velocity change.

Figure 6: AVO Curve obtained from the thin-hydrate model (solid line) compared with the one observed in the data (crosses).
more pronounced difference between the observed and modeled zero offset reflection amplitudes. Consequently, a thin-hydrate layer overlaying brine-saturated sediments is not sufficient to explain the seismic data.

Based on this result, the subsequent modeling attempted to decrease the P-wave velocity contrast at the BSR in order to recreate the observed zero offset reflection amplitudes. The required decrease was performed by thickening the hydrate layer, thus yielding a thick-hydrate over brine sediment model. An evaluation of the effects of several different velocity combinations on the reflection amplitudes resulted in the model shown in Figure 7. The estimated P-wave velocity in the hydrate corresponds to the measured interval velocity, yielding a considerable thickness of the hydrate zone. The S-wave velocity was again obtained using a Poisson’s ratio of 0.4 and is thus the same as in the initial model. The AVO curve based on the transition from the hydrate to the brine-saturated sediments was determined by Zoeppritz modeling and is shown in Figure 8.

The comparison of the modeled AVO responses with those observed indicates that this model could successfully reproduce the zero offset data. This suggests that the modeled P-wave velocities of 2.5 km/s in the hydrate and 1.6 km/s in the underlaying sediments might resemble the actual conditions at the BSR. However, the obtained AVO trend is still contrary to the observed one, displaying increasingly positive amplitudes with increasing offset. Hence, a change in Poisson’s ratio seems to be required at the transition from the hydrate-bearing sediments above the BSR to the sediments below the BSR.

Continuously changing the possible velocities in the hydrate zone and the characteristics of the underlaying sediments resulted finally in a hydrate layer characterized by a P-wave velocity of approximately 2.5 km/s and an anomalously low S-wave velocity of around 0.5 km/s yielding a Poisson’s ratio of 0.47. The hydrates appear to be underlain by sediments having a P-wave velocity of 1.6 km/s and an S-wave velocity of 1.1 km/s, yielding a Poisson’s ratio of 0.1 which is consistent with free gas. The final model can be seen in Figure 9. The initial model is given by the dotted line whenever the modeled velocities differ from the initial ones. Based on the determined interval velocities, the thickness of the hydrate layer was calculated to be approximately 190 meters and the one of the gas layer to be around 250 meters. Neglecting possible tuning effects, a thin gas layer was not a good model representation, as it required a decrease in P-wave velocity with respect to the hydrate layer to preserve the measured interval velocity below the BSR. Thus, it resulted in a significant deviation of the zero offset reflection amplitudes of the model and the true seismic data.

A comparison of the synthetic AVO curve obtained for the model shown in Figure 9 with the amplitude picks obtained from a representative CMP gather is shown in Figure 10. Both the synthetic and the real data AVO amplitude responses are in good agreement for near and far offsets. Thus, a significant increase in S-wave velocity and a simultaneous decrease in P-wave velocity at the transition from hydrate-bearing sediments to sediments containing free gas is required to explain the observed seismic data.
Figure 7: Interval velocities above and below the BSR for a thick-hydrate over brine sediment model. The modeled velocities are correspond to the initial interval velocities.

Figure 8: AVO curve obtained from the thick-hydrate model (solid line) compared with the one observed in the data (crosses).
Figure 9: Interval velocities for hydrate-bearing sediments overlaying gas-saturated sediments. The dotted lines represent the initial velocities. The arrows indicate where the modeled velocities had to be increased or decreased to match the seismic data.

Figure 10: Synthetic AVO curve of hydrates overlaying sediments saturated with free gas (solid line) compared with an observed one (crosses).
The final velocity model for the entire section is shown in Figure 11. The deviation from the initial interval velocities is indicated by the dotted line. While the initial P-wave interval velocities corresponded to the modeled velocities in the hydrate and gas sediments, the S-wave velocities had to be modified with regard to apparently different shear properties in the hydrates and the gas compared to the brine sediments.

Based on the modeled increase of the S-wave velocity at the bottom of the hydrate stability zone, a large positive S-impedance contrast can be predicted for the seismic data. On the other hand, a negative P-impedance contrast can be expected at the BSR due to the decrease in P-wave velocity at the transition from hydrate to gas. In order to determine the actual effect, we performed a prestack migration impedance inversion of the seismic data.

Figure 11: Final modeled interval velocity model. The dotted line indicates where the initial model differs from the final model. The arrows describe the direction the velocity had to be changed in order to fit the seismic data.

**IMPEDANCE ESTIMATION**

The P- and S-impedance contrasts at each subsurface position were estimated by applying a least-squares elastic parameter inversion method (Lumley and Beydoun, 1991; Lumley, 1993) to the CMP gathers. This technique fits the prestack migrated AVO gathers at each pseudo-depth and surface position to the theoretical P- and S-impedance curves which are based on linearized Zoeppritz equations. The least-squares impedance inversion results of the preprocessed CMP gathers are shown in Figures 12 and 13.
Figure 12: P-impedance contrast section.

Figure 13: S-impedance contrast section.
The P-impedance contrast section shows clearly that the seafloor reflection and the BSR have P-impedance contrasts of opposite polarity and approximately the same magnitude. In a small section above the BSR there is a “quiet” zone where no diffractions or reflections are visible, which can possibly be explained by the presence of disseminated methane hydrate. The S-impedance section is dominated by a very strong impedance contrast at the BSR. Although the seafloor has a much weaker impedance contrast, it is evident that both have the same polarity. Below the BSR, a horizontal reflector gives a strong P-impedance contrast of the same polarity as the seafloor, indicating an increase in P-wave velocity at the reflector, but weak S-impedance. This may be indicative of the base of the gas zone.

Assuming a seafloor reflection of 0.2 and assuming the P- and S-impedance contrasts at the seafloor, it is possible to estimate the relative impedance contrasts of the BSR by determining the average amplitude changes between seafloor and BSR. This results in a P-impedance contrast of −0.4 at the BSR which has the same magnitude but opposite polarity to the seafloor. The S-impedance contrast is very strong and amounts to approximately 0.8 to 1.2, which is two to three times as much as the seafloor impedance contrast of the same polarity. This anomalous S-impedance behavior is even more pronounced by making a simple P*S anomaly map shown in Figure 14. Normal impedance structure is plotted dark grey, while anomalous impedance structure is indicated by white. The high magnitude contrast has the effect of visually diminishing the S-impedance contrast of the seafloor (Figure 13) compared with the P-impedance contrast (figure 12) at the seafloor, which are actually the same magnitude.

![Figure 14: P*S impedance contrast.](avo/imp/ PSmap-ann)
The negative P-impedance contrast and the large positive S-impedance contrast at
the BSR are in good agreement with the prediction based on the Zoeppritz modeling. The strong positive S-impedance contrast clearly supports the modeled S-wave velocity behavior of anomalously low S-velocity in the hydrates and considerably increased S-velocity in the underlain gas sediments. Based on the synthetic modeling and the impedance inversion results, the geology was interpreted as seen in Figure 15. In this interpretation, the hydrate-bearing sediments are assumed to overlay sediments in which free gas is trapped. The flat reflector below the BSR might correspond to a gas-water contact at the base of the free gas zone based on the impedance contrasts obtained for this reflector.

![Figure 15: Interpretation of the methane hydrate seismic data from the Blake Outer Ridge.](image)

**CONCLUSIONS**

A detailed AVO analysis was performed on data from the Blake Outer Ridge to evaluate the origin of the bottom simulating reflector. Reflectivity clearly discriminates the effects of different models and shows that the observed BSR best fits a model of sediments containing substantial amounts of hydrate overlaying sediments containing free gas. This modeling result was supported by a prestack impedance inversion of the seismic data. A transition from hydrate to brine sediments is not consistent with the AVO amplitude responses and the impedance contrasts.

Based on the synthetic modeling, the thickness of the hydrate layer was determined to be approximately 190 meters. It is characterized by a P-wave velocity of around 2.5 km/s and an anomalously low S-wave velocity of 0.5 km/s. The thickness of the gas layer was calculated to be approximately 250 meters. It has a P-wave velocity of
1.6 km/s and an S-wave velocity of 1.1 km/s, yielding a Poisson’s ratio of 0.1 which is reasonable for gas. The considerable thickness of the gas layer might suggest the possibility of it being a source rock for the overlying methane hydrate.

It has to be considered, however, that the synthetic modeling excluded possible tuning effects and thus might represent a simplification of the actual conditions. Nevertheless, the velocity behavior predicted by the model was reinforced by the prestack impedance inversion, thus indicating that a transition from high P-wave velocity and anomalously low S-wave velocity in the hydrate to low P-wave velocity and high S-wave velocity in the gas sediments is required. A detailed investigation of this unusual behavior is performed by Ecker (1994).

ACKNOWLEDGMENTS

We thank the sponsors of the Stanford Exploration Project and Professor Jon Claerbout for supporting this research. We also thank Keith Kvenvolden, Bill Dillon and Myung Lee of the USGS for providing us with a copy of the Blake Outer Ridge seismic data used in this study.

REFERENCES

