Wavelets and warping PS seismic images

Christopher F. Graziano

- Master of Science Thesis -
Geophysics

Defended on October 30, 2015

Advisor: Prof. Dave Hale
Committee Members: Prof. Walter Lynn
Pro. Terence Young

Center for Wave Phenomena
Colorado School of Mines
Golden, Colorado 80401
303.384.2178
cwpinfo@mines.edu
cwp.mines.edu
WAVELETS AND WARPING PS SEISMIC IMAGES

by

Christopher F. Graziano
ABSTRACT

Many seismic processing techniques warp a seismic trace. This warping often results in wavelet distortion throughout the trace and needs to be reduced to prevent errors in future processing steps. I propose a new method to reduce wavelet distortion that deconvolves the wavelet from a trace to be warped, warps the resulting impulses without distorting them, and convolves with the wavelet in a reference trace. Moreover, this warping-with-wavelets algorithm can be used to estimate the required wavelets in the trace to be warped and in the reference trace. The primary advantage over previous methods is that this algorithm involves deconvolution of the wavelet from the trace to be warped before warping. I offer a specific example of where this method can be applied to reduce wavelet distortion by warping the reflectors in a PS seismic image to align with those in a PP seismic image. Not only is wavelet distortion reduced in the warped PS image, but PP and PS wavelets are also estimated.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>xi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>xiii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2 WARPING-WITH-WAVELETS</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Warping impulses without distortion</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Warping wavelets without distortion</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 3 ESTIMATING WAVELETS</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Estimating one wavelet</td>
<td>15</td>
</tr>
<tr>
<td>3.2 Estimating two wavelets with the cyclic search</td>
<td>23</td>
</tr>
<tr>
<td>3.3 Estimating two wavelets with the Gauss-Newton method</td>
<td>29</td>
</tr>
<tr>
<td>3.4 Comparison of cyclic search and Gauss-Newton method</td>
<td>38</td>
</tr>
<tr>
<td>CHAPTER 4 APPLICATION TO PP-PS IMAGES</td>
<td>42</td>
</tr>
<tr>
<td>4.1 Data Conditioning</td>
<td>42</td>
</tr>
<tr>
<td>4.2 Wavelets estimated with the cyclic search</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Wavelets estimated with the Gauss-Newton method</td>
<td>47</td>
</tr>
<tr>
<td>4.4 Warping-with-wavelets result compared with the shaping filter result</td>
<td>50</td>
</tr>
<tr>
<td>CHAPTER 5 CONCLUSIONS</td>
<td>58</td>
</tr>
<tr>
<td>5.1 Future Work</td>
<td>58</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 2.1 Impulsive signals $p(t)$ (a) and $q(t)$ (b) to be aligned by warping. The relationship between $p(t)$ and $q(t)$ is $p(t) = 2q(2t)$. Using the known time shifts to warp $q(t)$ to $p(t)$ yields no wavelet distortion (c). ............ 5

Figure 2.2 Impulsive sequences $p$ (a) and $q$ (b) to be aligned by warping. The relationship between $p$ and $q$ is $p = Sq$. Using the known time shifts to warp $q$ to be aligned with $p$ yields no wavelet distortion (c). ............... 8

Figure 2.3 Known wavelet $c$ in both the reference trace $f$ and the trace to be warped $g$ shown in Figure 2.4. ................................................. 8

Figure 2.4 Both sequences $f$ (a) and $g$ (b) have the same wavelet $c$. Note that simply squeezing $g$ to $f$ causes wavelet distortion (c). ..................... 9

Figure 2.5 Sequence $g$ (a), the sequence $Ag$ (b) obtained by deconvolving the wavelet $c$ from $g$, the sequence $SAg$ (c) obtained by squeezing $Ag$, and the sequence $CSAg$ (d) obtained by convolving $SAg$ with the wavelet $c$. 10

Figure 2.6 Sequence $f$ (a), the sequence $CSAg$ (b) obtained using the warping-with-wavelet algorithm, and the sequence $Sg$ (c) obtained using only warping. ............................................................ 11

Figure 2.7 Sequence $f$ (a) contains the wavelet $d$, and the sequence $g$ (b) contains the wavelet $c$. Simply warping $g$ to $f$ causes wavelet distortion (c). ........ 12

Figure 2.8 Known wavelet $c$ (a) in the reference trace $f$ and the known wavelet $d$ (b) in trace to be warped $g$. The traces $f$ and $g$ are shown in Figure 2.7. . 13

Figure 2.9 Sequence $f$ (a), the sequence $CSBg$ (b) obtained using the warping-with-wavelet algorithm, and the sequence $Sg$ (c) obtained using only warping. ............................................................ 13

Figure 3.1 A simple PP trace ($f$) (a), and a simple PS trace ($g$) (b). Both traces have the same wavelet and the same reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = 2n$, indicating that when one squeezes the PS trace to the PP trace, a constant amount of squeezing (d) is applied that causes the same wavelet distortion throughout the squeezed PS trace $Sg$ (c). ........ 16
Figure 3.2 Estimated wavelet (dots) computed by the wavelet-estimation process is nearly identical to the known wavelet (red curve) in $f$ and $g$ from Figure 3.1. To make the wavelet $c$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. ................................................................. 18

Figure 3.3 Estimated shaping filter $h$ used to shape $S_g$ (Figure 3.1c) to $f$ (Figure 3.1a). To make the shaping filter $h$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. .............................................................. 18

Figure 3.4 Sequence $f$, from Figure 3.1, is in each panel above in black. The sequence $CS_Ag$ (a) obtained using the warping-with-wavelets algorithm and the sequence $HS_g$ (b) obtained using the shaping filter $h$ are overlaid on top of $f$ in red. Only portions of $f$ and $CS_Ag$ are shown because this display highlights only the samples in $f$ that correspond to samples in $g$ and, therefore, $CS_Ag$. ............................................ 19

Figure 3.5 A simple PP trace ($f$) (a), and a simple PS trace ($g$) (b). Both traces have the same wavelet and reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = a \ln(1 + bn)$, indicating that when one squeezes the PS trace to the PP trace, a varying amount of squeezing is applied that causes different amounts of wavelet distortion throughout the squeezed PS trace $S_g$ (c). ............... 20

Figure 3.6 Estimated wavelet (dots) computed by the wavelet-estimation process is nearly identical to the known wavelet (red curve) in $f$ and $g$ from Figure 3.5. To make the wavelet $c$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. ................................................................. 21

Figure 3.7 Estimated shaping filter $h$ used to shape $S_g$ (Figure 3.5c) to $f$ (Figure 3.5a). To make the shaping filter $h$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. .............................................................. 21

Figure 3.8 Sequence $f$, from Figure 3.5, is in each panel above in black. The sequence $CS_Ag$ (a) obtained using the warping-with-wavelets algorithm and the sequence $HS_g$ (b) obtained using a shaping filter are overlaid on top of $f$ in red. ..................................................... 22
Figure 3.9 A simple PP trace (f) (a), and a simple PS trace (g) (b). f and g have different wavelets (c and d, respectively), but the same reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = a \ln(1 + bn)$, indicating that when one squeezes the PS trace to the PP trace, a varying amount of squeezing is applied that causes different amounts of wavelet distortion throughout the squeezed PS trace $Sg$ (c).

Figure 3.10 Starting wavelets for the cyclic search are c (a), a shaping filter h that shapes $Sg$ (Figure 3.9c) to f (Figure 3.9a), and d (b), a unit impulse. To make the wavelets c and d appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. Note that the wavelet d has only been interpolated with zeros to emphasize that this wavelet starts as an impulse.

Figure 3.11 RMS of all residuals for each iteration in the cyclic search.

Figure 3.12 Estimated c (a) and d (b) wavelets (dots) computed by the cyclic search are similar to the known c (a) and d (b) wavelets (red curve) in f and g from Figure 3.9. To make the wavelets c and d appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Figure 3.13 Sequence f, from Figure 3.9, is in black in each of the above panels. The sequence CSBg (a) obtained using the warping-with-wavelets algorithm and wavelets estimated by the cyclic search is in red and overlaid onto f. The sequence HSG (b) obtained using a shaping filter h is also in red and is overlaid onto f.

Figure 3.14 RMS of all residuals for each iteration in the Gauss-Newton method.

Figure 3.15 Estimated c (a) and d (b) wavelets (dots) computed by the Gauss-Newton method are nearly identical to the known c (a) and d (b) wavelets (red curve) in f and g from Figure 3.9. To make the wavelets c and d appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Figure 3.16 Sequence f, from Figure 3.9, is in black in each of the above panels. Sequence CSBg (a) obtained using the warping-with-wavelets algorithm and wavelets estimated by the Gauss-Newton method is in red and overlaid onto f. Sequence HSG (b) obtained using a shaping filter h is also in red and are overlaid onto f.
Figure 3.17  Sequence \( f \) with a noise-to-signal ratio of 0.0 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e), and 1.0 (f) and sequence \( g \) with a noise-to-signal ratio of 0.0 (g), 0.2 (h), 0.4 (i), 0.6 (j), 0.8 (k), and 1.0 (l). .......................... 38

Figure 3.18  Known \( c \) (a) and \( d \) (b) wavelets represented by the red line and the cyclic search estimated \( c \) and \( d \) wavelets represented by the black, green, blue, cyan, pink, and orange dots from \( f \) and \( g \) with noise-to-signal ratios of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, respectively. To make the wavelets \( c \) and \( d \) appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. .......................... 39

Figure 3.19  Known \( c \) (a) and \( d \) (b) wavelets represented by the red line and the Gauss-Newton method estimated \( c \) and \( d \) wavelets represented by the black, green, blue, cyan, pink, and orange dots from \( f \) and \( g \) with noise-to-signal ratios of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, respectively. To make the wavelets \( c \) and \( d \) appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. .......................... 40

Figure 4.1  PP image (a) and a PS image (b). Both are provided courtesy of SINOPEC. ................................................................. 42

Figure 4.2  Noisy PS image (a) and the result of applying a recursive exponential filter (b) to reduce the noise. All images have been scaled for display to have the same RMS amplitude. ............................................. 43

Figure 4.3  Initial PP (a) and PS (b) wavelets for cyclic search and Gauss-Newton method. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. Note that the PS wavelet has only been interpolated with zeros to emphasize that this wavelet starts as an impulse. .......................... 44

Figure 4.4  RMS of all residuals for each iteration in the cyclic search. .......................... 45

Figure 4.5  Estimated PP (a) and PS (b) wavelets using cyclic search. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. .... 45

Figure 4.6  Amplitude spectra of the PP (a) and PS (b) wavelets estimated using the cyclic search, and amplitude spectra of the PP (c) and PS (d) images. .... 46

Figure 4.7  RMS of all residuals for each iteration in the Gauss-Newton method. .... 47

Figure 4.8  RMS of all residuals for each iteration in the Gauss-Newton method with a stopping condition of 0.0001 percent. .......................... 48
Figure 4.9 Estimated PP (a) and PS (b) wavelets using Gauss-Newton method. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Figure 4.10 Amplitude spectra of the PP (a) and PS (b) wavelets estimated using the Gauss-Newton method, and amplitude spectra of the PP (c) and PS (d) images.

Figure 4.11 PP image (a) with PS images obtained by applying a shaping filter after warping (b), and simply warping the PS image (c). All images have been scaled for display to have the same RMS amplitude.

Figure 4.12 PP image (a) with PS images obtained by applying a shaping filter after warping (b), and simply warping the PS image (c). Note that only data between 0.4 and 0.8 seconds are shown. All images have been scaled for display to have the same RMS amplitude.

Figure 4.13 $V_p/V_s$ ratios estimated using smooth dynamic warping.

Figure 4.14 PP image (a) with PS images obtained by three warping methods: the warping-with-wavelets algorithm using the cyclic search (b), the warping-with-wavelets algorithm using the Gauss-Newton method (c), and the application of a shaping filter after warping (d). All images have been scaled for display to have the same RMS amplitude.

Figure 4.15 PS image (a) with deconvolved PS images obtained by deconvolving the PS wavelet estimated by the Gauss-Newton method (b) and the cyclic search (c). All images have been scaled for display to have the same RMS amplitude.

Figure 4.16 Amplitude spectra of the PS image (a), the PS image convolved with the inverse PS wavelet $b$ estimated by the Gauss-Newton method (b), and the PS image convolved with the inverse PS wavelet $b$ estimated by the cyclic search (c).
impulsive reference signal .......................................................... \( p(t) \)

impulsive signal to be warped ......................................................... \( q(t) \)

function that relates time in \( q(t) \) with time in \( p(t) \) .......................... \( u(t) \)

sampling interval ............................................................................ \( T \)

\( p(t) \) uniformly sampled................................................................. \( p \)

\( q(t) \) uniformly sampled................................................................. \( q \)

\( u(t) \) uniformly sampled................................................................. \( u \)

upsampling operator ........................................................................ \( U \)

squeezing operator (does not account for aliasing) .............................. \( W \)

low-pass filter operator .................................................................. \( L \)

downsampling operator ................................................................. \( V \)

squeezing operator (accounts for aliasing) ........................................ \( S \)

reference trace (PP trace or image) ................................................ \( f \)

wavelet in \( f \) (PP wavelet) ............................................................ \( c \)

inverse wavelet of the wavelet \( c \) (inverse PP wavelet) ....................... \( a \)

trace to be warped (PS trace or image) ........................................... \( g \)

wavelet in \( g \) (PS wavelet) ............................................................ \( d \)

inverse wavelet of the wavelet \( d \) (inverse PS wavelet) ....................... \( b \)

unit impulse.................................................................................. \( \delta \)

difference between \( F \) and \( SG \) ................................................. \( Y \)
shaping filter that shapes $Sg$ to $f$ ......................................................... $h$

sequence $SBg$ ......................................................................................... $q$

matrix $CSG$ .............................................................................................. $P$

number of samples in $f$ ........................................................................ $n_f$

number of samples in the wavelet $c$ ................................................ $n_c$

number of samples in the inverse wavelet $b$ ........................................ $n_b$

number of samples in the inverse wavelet $a$ ........................................ $n_a$

Jacobian matrix ......................................................................................... $J$

total residual vector ................................................................................ $r$

data residual vector ............................................................................. $r_d$

penalization residual vector .................................................................. $r_p$

scalar that represents the degree at which outer lags of $b$ are penalized .... $\gamma$

scaled diagonal matrix ............................................................................. $X$

total update vector for iteration $k$ ....................................................... $\delta_k$

update vector for the inverse wavelet $b$ for iteration $k$ ..................... $\delta_{b_k}$

update vector for the wavelet $c$ for iteration $k$ ................................ $\delta_{c_k}$
ACKNOWLEDGMENTS

I would first like to thank my family for always believing in me and for their endless support through the good times and the tough times (try harder, fly higher). I would also like to thank my advisor Dr. Dave Hale for his guidance and support. If I did not understand a concept at first, he would take the time to explain the topic in a different way until I understood it. My time at the Center for Wave Phenomena has been very enjoyable thanks to my friends at CWP and for Pamela Kraus, Diane Witters, and Shingo Ishida, they were always there to lend an ear. Thanks also to SINOPEC for providing the PP and PS images used in this study.
Many seismic processing techniques squeeze or stretch a seismic trace. One technique that stretches a trace is normal moveout (NMO) correction. Barnes (1992) points out that when NMO correction is applied to a common midpoint (CMP) gather, nonzero-offset traces are stretched, such that the events in these traces align with corresponding events in the zero-offset trace. A technique that squeezes a trace is the process of aligning the events in a PS image with those in a PP image. Events in a PS image appear at a later time than in a PP image because the events in a PP image are a result of a downward propagating P-wave that is reflected upward, while the events in a PS image are a result of a downward propagating P-wave that is converted to an upward propagating S-wave upon reflection. Because S-waves are slower than P-waves, it takes more time for the information seen in a PS image to reach the receivers (Stewart et al., 2002). In both cases, warping (stretching or squeezing) a trace will produce wavelet distortion, which can cause problems for later processing steps.

Barnes points out that after NMO correction, the wavelets present in corrected traces are stretched, which indicates a decrease in resolution. Because this effect is considerably worse at larger offsets and earlier times, a mute is used to zero data that is stretched beyond a set threshold. Several authors also note that the process of squeezing a PS image to PP time distorts the seismic wavelet (Bansal & Matheney, 2010; Gaiser et al., 2011, 2013; Ursenbach et al., 2013). If the required amount of squeezing varies in space or time, then the warped PS wavelet also varies. The resulting nonstationary warped PS wavelet can cause errors in inversions for subsurface properties such as density and P- and S-wave velocities (Bansal & Matheney, 2010; Jing & Rape, 2004; Khare & Rape, 2007; Veire & Landrø, 2006).

Several techniques have been developed to limit wavelet distortion caused by applying NMO correction. One of these techniques, proposed by Biondi et al. (2014), performs par-
tial NMO corrections followed by the application of shaping filters. By performing these partial corrections, the wavelets, along the reflector that is being corrected, become slightly distorted. This slight distortion is removed when these distorted wavelets are shaped to stretch-free wavelets, which are estimated along the same reflector in the original CMP gather. This technique is similar to techniques used to reduce wavelet distortion caused by squeezing a PS image because they involve shaping the distorted wavelets to some desired result.

A proposed solution to the problem of PS wavelet distortion has been to design a filter, for each time, that shapes the distorted PS wavelet to a single desired stationary PS wavelet (Bansal & Matheney, 2010). To design the necessary shaping filters, the original wavelet in the PS image must be extracted. Bansal & Matheney (2010) do not describe the wavelet extraction process, but given the original PS wavelet and the amount of squeezing, they compute a squeezed wavelet. The spectrum of the appropriate shaping filter is calculated by dividing the spectrum of the desired stationary PS wavelet by the spectrum of the squeezed wavelet. Applying time- and space-varying shaping filters computed in this way removes wavelet distortion from the squeezed PS image.

Another method of correcting PS wavelet distortion is proposed by Gaiser et al. (2011). This method uses nonstationary linear filters to modify the S-wave periods of warped PS-waves to match the periods of P-waves. This modification causes the warped PS-waves to resemble S-waves that have been transformed to P-wave time. Gaiser et al. (2011) point out that this method is able to better match the PS-wave data with the P-wave data than if only warping was applied, but wavelet distortions occur because the average $V_p/V_s$ ratios are assumed to be constant over the time of the wavelet. Gaiser et al. (2013) remove this assumption to reduce wavelet distortion and apply filters that compress PS wavelets to yield results similar to those in Gaiser et al. (2011).

In previously proposed solutions, the common step to correct wavelet distortion caused by NMO correction or by squeezing a PS image is to apply a filter after warping to shape
the distorted wavelet to some desired result. I propose a different solution that includes deconvolving the wavelet before warping, based on the observation that the problem of wavelet distortion would not exist if the wavelet was an impulse. Moreover, my warping-with-wavelets algorithm provides a method to estimate the required wavelet to deconvolve.

In this thesis, my main focus is to discuss how warping-with-wavelets algorithm can reduce PS wavelet distortion when squeezing a PS image to PP time, but recognize that these concepts can also be used to reduce wavelet distortion caused by NMO correction or any other geophysical transformation that involves squeezing or stretching. In Chapter 2, I explore how this method can be used to squeeze a simple trace to be warped to a reference trace, assuming that the wavelets in both traces are known. Chapter 3 then explains how one can estimate the wavelets in the reference trace and trace to be warped. This idea is then tested on synthetic PP and PS traces. Chapter 4 presents the results of applying this method to real PP and PS images.
CHAPTER 2
WARPING-WITH-WAVELETS

This chapter demonstrates how to squeeze a trace to be warped to a reference trace while avoiding wavelet distortion. After presenting the concept of squeezing impulses without wavelet distortion, I apply this same concept to squeezing a trace without wavelet distortion. In this work, I assume that the trace to be warped and the reference trace have the same reflection coefficients and the same wavelet. I also examine the case where the trace to be warped and the reference trace have different wavelets, which is similar to warping a PS trace to a PP time. The process of squeezing a trace without wavelet distortion involves 1) deconvolving the wavelet from the trace to be warped, 2) squeezing the impulses to their corresponding events in the reference trace, and 3) convolving these squeezed impulses with the wavelet from the reference trace.

2.1 Warping impulses without distortion

Suppose that the wavelet in a seismic signal is a delta function $\delta(t)$. Figures 2.1a and 2.1b represent two simple synthetic signals:

$$p(t) = \delta(t - t_1) - \delta(t - t_2),$$
$$q(t) = \delta(t - 2t_1) - \delta(t - 2t_2),$$

such that

$$p(t) = 2q(2t). \tag{2.1}$$

Equation 2.1 is a special case of the more general relationship

$$p(t) = u'(t)q(u(t)), \tag{2.2}$$

where $u(t)$ is the mapping from time in $q(t)$ to time in $p(t)$. The amplitude scaling by $u'(t)$ in equations 2.1 and 2.2 is necessary because both the squeezed and original delta functions
need to be equivalent to avoid wavelet distortion. To make these functions equivalent, one multiplies the squeezed delta function by the derivative of the input to the squeezed function as shown by the following property of the Dirac delta function: \( \delta(t) = \beta \delta(\beta t) \). Note that the function \( 2q(2t) \) displayed in Figure 2.1c exhibits no wavelet distortion.

![Figure 2.1: Impulsive signals](image)

In the case described above, the signals are assumed to be continuous and need to be sampled. Let column vectors \( \mathbf{p} \) and \( \mathbf{q} \) represent the sampled versions of continuous signals \( p(t) \) and \( q(t) \), respectively.

In the sampled world, one can represent equation 2.2 as

\[
p[n] = \frac{(u[n] - u[n - 1])}{T} q[u[n]], \tag{2.3}
\]

where \( n \) is a sample index in \( \mathbf{p} \), \( u \) is a sample index in \( \mathbf{q} \), and \( T \) is the sampling interval in \( \mathbf{p} \) and \( \mathbf{q} \). Note that \( u[n] \) has the same length as the number of samples in \( \mathbf{p} \) and contains the sample indices of \( \mathbf{q} \) (\( u \)) that correspond to the sample indices in \( \mathbf{p} \) (\( n \)). For simplicity, I have not included any sinc interpolation that would be required to evaluate \( q[u[n]] \) in equation 2.3.
One can rewrite equation 2.3 as

\[ p = Wq, \quad (2.4) \]

where the matrix \( W \) represents the linear operator required to squeeze \( q \) to \( p \) and scale \( q \) as outlined in equation 2.3. Note that according to the scaling theorem of the Fourier transform, when one squeezes a sequence in the sampled domain by some factor, the Fourier transform of the sequence is stretched by that same factor. This is an issue because \( q \) is composed of impulses that contain frequencies out to the Nyquist frequency and when \( q \) is squeezed, these frequencies will be aliased.

To solve this aliasing issue, I assume that the amount of squeezing applied to \( q \) is constant at 2 and is represented with the matrix \( W_c \). I also assume that the sampling interval for all sampled sequences in the rest of this thesis is 4 ms/sample, implying a Nyquist frequency of 125 Hz, which is also the maximum frequency of \( q \). When \( q \) is squeezed to \( p \) \((W_c q)\), the maximum frequency of \( W_c q \) is 250 Hz, which is above the current Nyquist frequency and indicates that \( W_c q \) is aliased. To prevent aliasing, I increase the Nyquist frequency by the maximum amount of squeezing applied to \( q \) by the squeezing operator \( W \). In the case of \( W_c \), the maximum amount of squeezing is 2.

To increase the Nyquist frequency by 2, I decrease the sampling interval by 2 using interpolation. Because the original sampling interval is 4 ms/sample, the new sampling interval needs to be 2 ms/sample. By decreasing the sampling interval by 2, one increases the Nyquist frequency to 250 Hz. This upsampling operation is represented by the matrix \( U \).

The sequence \( Uq \) has a Nyquist frequency of 250 Hz and a maximum frequency of only 125 Hz. By applying the squeezing operator \( W_c \), the maximum frequency increases to 250 Hz, which is not above the Nyquist frequency of the sequence and indicates that aliasing does not occur. This squeezed sequence is labeled \( W_c Uq \).

Once I have squeezed the events in \( q \) to the events in \( p \), the sampling interval of \( W_c Uq \) needs to be increased back to the original sampling interval of 4 ms/sample. Recall that the
maximum and Nyquist frequencies of $W_cUq$ are both 250 Hz. If I were to simply increase the sampling interval from 2 to 4 ms/sample, the resulting sequence would be aliased because the maximum frequency (250 Hz) of the sequence would be above its new Nyquist frequency (125 Hz). This indicates that before downsampling, I need to apply a low-pass anti-aliasing filter to $W_cUq$ to suppress any frequencies that are above the new Nyquist frequency of 125 Hz. The low-pass filter operator is represented by the matrix $L$.

The sequence $LW_cUq$ has a Nyquist frequency of 250 Hz and a maximum frequency of 125 Hz. When I increase the sampling interval of $LW_cUq$ from 2 to 4 ms/sample, the sequence is not aliased because the maximum and Nyquist frequencies are both 125 Hz. This downsampling operator is represented by the matrix $V$, which produces the final warped sequence $VLW_cUq$.

The above scenario where $W_c$ is used as a squeezing operator is a simple case. It is also a case where $W_cLq$ would have been sufficient to prevent aliasing. In more complicated and realistic cases, the amount of squeezing from the trace to be warped to the reference trace varies (represented by the matrix $W_v$), simply applying a low-pass filter before squeezing would be insufficient.

For simplicity, I assume that $W_v$ applies an amount of squeezing that varies from 3 to 2. This indicates that the low-pass filter $L$ in $W_vLq$ will suppress all frequencies above one-third of the current Nyquist frequency (the Nyquist frequency divided by the maximum amount of squeezing) to prevent aliasing. Because the amount of squeezing applied by $W_v$ varies with time, the stretching of the Fourier transform is not constant at 3 and the resulting Fourier transform will not have frequencies stretched back up to the Nyquist frequency, which will not preserve the impulsive characteristic of $q$.

One can represent the cascade of operators $VLW_cUq$ or $VLW_vUq$ by a single matrix $S$. The result of squeezing $q$ to $p$ without aliasing is represented by $p = Sq$. For completeness and clarification, Figure 2.1 has been relabeled with this new squeezing notation in Figure 2.2.
2.2 Warping wavelets without distortion

Consider a non-impulsive wavelet $c$ (Figure 2.3) with Z transform

$$C(z) = \frac{1}{(1 - 1.68z + 0.76z^2)(1 - 1.27z^{-1} + 0.76z^{-2})},$$  

(2.5)

which has an infinite number of wavelet coefficients and five inverse wavelet coefficients,

Figure 2.3: Known wavelet $c$ in both the reference trace $f$ and the trace to be warped $g$ shown in Figure 2.4.

and also consider synthetic traces computed by convolving this wavelet with the sequences
of impulses p and q from Figure 2.2:

\[
f = c \ast p, \quad (2.6)
\]

\[
g = c \ast q. \quad (2.7)
\]

This convolution with the wavelet c complicates the relationship between f (Figure 2.4a) and g (Figure 2.4b), so that \( f \neq Sg \) (Figure 2.4c).

Figure 2.4c illustrates that, where the wavelet is not an impulse, simply warping one trace to align with another trace distorts the wavelet. The examples in Figures 2.2 and 2.4 suggest that deconvolution of the wavelet should be performed before warping. That is, letting a denote the inverse of the wavelet c such that \( a \ast c = \delta \), one should (1) convolve with the inverse wavelet a, (2) warp, and (3) convolve with the wavelet c. This process is known as warping-with-wavelets.

Figure 2.4: Both sequences f (a) and g (b) have the same wavelet c. Note that simply squeezing g to f causes wavelet distortion (c).

Recall that the convolution of two sequences (a and g) can be written as the multiplication of a matrix with a vector. For illustration, let us suppose that the inverse wavelet a has three coefficients \( (a_0, a_1, a_2) \) and g has 5 coefficients \( (g_0, g_1, g_2, g_3, g_4) \), such that convolution of
a with g is represented by

\[
A_g = \begin{bmatrix}
    a_0 & 0 & 0 & 0 & 0 \\
    a_1 & a_0 & 0 & 0 & 0 \\
    a_2 & a_1 & a_0 & 0 & 0 \\
    0 & a_2 & a_1 & a_0 & 0 \\
    0 & 0 & a_2 & a_1 & a_0
\end{bmatrix}
\begin{bmatrix}
    g_0 \\
    g_1 \\
    g_2 \\
    g_3 \\
    g_4
\end{bmatrix},
\]

or equivalently,

\[
G_a = \begin{bmatrix}
    g_0 & 0 & 0 \\
    g_1 & g_0 & 0 \\
    g_2 & g_1 & g_0 \\
    g_3 & g_2 & g_1 \\
    g_4 & g_3 & g_2
\end{bmatrix}
\begin{bmatrix}
    a_0 \\
    a_1 \\
    a_2
\end{bmatrix}.
\]

Note that columns of A contain only delayed copies of a and columns of G contain only delayed copies of g, so that both A and G are Toeplitz matrices. Also, recall that convolution is commutative (\(A_g = G_a\)) and that convolution of the inverse wavelet a with g is equivalent to deconvolution of the wavelet c from g.

![Figure 2.5: Sequence g (a), the sequence Ag (b) obtained by deconvolving the wavelet c from g, the sequence SAg (c) obtained by squeezing Ag, and the sequence CSAg (d) obtained by convolving SAg with the wavelet c.](image)

To warp g (Figure 2.4b) to f (Figure 2.4a) without wavelet distortion, warping-with-wavelets requires convolving g with its inverse wavelet a, yielding Ag (Figure 2.5b). This
deconvolved trace is then squeezed to align its events with those in \( f \) to construct \( SAg \), as shown in Figure 2.5c. The squeezing operator \( S \) is constructed using the known amount of squeezing required to squeeze events in \( g \) to the events in \( f \). Note that when working with real PS traces, which contain noise, it is unlikely that \( SAg \) will contain a set of impulses as in Figure 2.5c. Because it is unlikely that \( SAg \) will contain an ideal set of impulses, \( SAg \) is not the final product of the warping-with-wavelets algorithm. Finally, the wavelet \( c \) is convolved with \( SAg \) to get \( CSAg \) (Figure 2.5d). Figure 2.6 highlights that wavelet distortion is reduced by using warping-with-wavelets (Figure 2.6b) instead of by simply warping (Figure 2.6c).

![Figure 2.6: Sequence \( f \) (a), the sequence \( CSAg \) (b) obtained using the warping-with-wavelet algorithm, and the sequence \( Sg \) (c) obtained using only warping.](image)

Note that \( S \) is a linear and time-varying operator, meaning that \( S \) cannot commute (\( CSAg \neq CASg \) and \( CSAg \neq SCAg \)). If \( S \) did commute, \( CSAg \) would equal \( CASg \), which would simply be \( Sg \) because \( CA = I \). However, note that in Figure 2.6, \( CSAg \neq Sg \).

Thus far, I have presented a situation where one can warp \( g \) to \( f \) if both traces have the same wavelet. At this point, I assume that \( f \) and \( g \) have different wavelets, as shown in Figure 2.7.
To create these new \( f \) and \( g \), I redefine the Z transform of the wavelet \( c \) to be

\[
C(z) = \frac{1}{(1 - 1.27z + 0.76z^2)(1 - 0.54z^{-1} + 0.76z^{-2})}
\]  

and define the Z transform of a new wavelet \( d \) to be

\[
D(z) = \frac{1}{(1 - 1.68z + 0.76z^2)(1 - 1.27z^{-1} + 0.76z^{-2})}.
\]  

Figures 2.8a and 2.8b show the new wavelets \( c \) and \( d \), respectively. Both of these wavelets have an infinite number of wavelet coefficients and five inverse wavelet coefficients. The new \( f \) and \( g \) are a result of convolving \( p \) and \( q \) (from Figure 2.2) with wavelets \( c \) (Figure 2.8a) and \( d \) (Figure 2.8b), respectively.

Figure 2.7: Sequence \( f \) (a) contains the wavelet \( d \), and the sequence \( g \) (b) contains the wavelet \( c \). Simply warping \( g \) to \( f \) causes wavelet distortion (c).

To warp the trace to be warped \( g \) (Figure 2.7b) to the reference trace \( f \) (Figure 2.7a), following the warping-with-wavelets algorithm, one first convolves \( g \) with the inverse wavelet in \( g \) (b), squeezes, and then convolves \( SBg \) with the wavelet in \( f \) (c) so that

\[
f \approx CSBg.
\]  

The purpose for convolving \( SBg \) with the wavelet in \( f \) (c) is to compare differences between
Figure 2.8: Known wavelet c (a) in the reference trace f and the known wavelet d (b) in trace to be warped g. The traces f and g are shown in Figure 2.7.

Figure 2.9: Sequence f (a), the sequence CSBg (b) obtained using the warping-with-wavelet algorithm, and the sequence Sg (c) obtained using only warping.
f and g that are not related to the wavelet.

The result of using the warping-with-wavelets algorithm with two wavelets is seen in Figure 2.9. Note that, unlike the case where simply warping is used (Figure 2.9c), the warping-with-wavelets result has no wavelet distortion (Figure 2.9b). Also, note that by performing warping-with-wavelets and forcing f and g to have the same wavelet, one can tell that f and g have similar reflection coefficients, unlike the case in Figure 2.7 where this similarity in reflection coefficients is not apparent.
CHAPTER 3
ESTIMATING WAVELETS

While the warping-with-wavelets algorithm requires that the inverse wavelet in the trace to be warped and the wavelet in the reference trace are known, these wavelets are rarely known and need to be estimated. This chapter demonstrates how one can use the warping-with-wavelets algorithm to estimate the wavelets in the reference trace and the trace to be warped. If the reference trace and the trace to be warped are composed of the same wavelet, only one wavelet needs to be estimated, and if they are composed of two different wavelets, two wavelets need to be estimated. In the text that follows, I present one method to estimate one wavelet and two methods to estimate two wavelets.

3.1 Estimating one wavelet

A trace to be warped \((f)\) and a reference trace \((g)\) are displayed in Figure 3.1. Both traces are a result of convolving the wavelet \(c\) (defined by the Z transform in equation 2.5 and shown in Figure 2.3) with the same random reflection coefficients that are located at different samples within \(f\) and \(g\). The relationship between reflection coefficients in \(f\) and \(g\) is \(u[n] = 2n\) (Figure 3.1d). These traces represent simple PP \((f)\) and PS \((g)\) traces that have the same wavelet and the same reflection coefficients.

In Figure 3.1c, note that simply squeezing \(g\) to \(f\) causes wavelet distortion. To avoid this distortion using the warping-with-wavelets algorithm, the wavelet \(c\), or equivalently, its inverse wavelet \(a\), must be known. I now consider how one might use warping-with-wavelets to estimate \(a\) and, hence, \(c\).

Because only the wavelet \(c\) exists in \(f\) and \(g\), the warping-with-wavelets algorithm is simply

\[
f \approx \text{CSAg}.
\]

(3.1)
Figure 3.1: A simple PP trace (f) (a), and a simple PS trace (g) (b). Both traces have the same wavelet and the same reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = 2n$, indicating that when one squeezes the PS trace to the PP trace, a constant amount of squeezing (d) is applied that causes the same wavelet distortion throughout the squeezed PS trace $S_g$ (c).

Multiplying both sides of equation 3.1 by $A$, one obtains

$$A_f \approx S A_g. \quad (3.2)$$

Because convolution is commutative, equation 3.2 can be rewritten as

$$F_a \approx S G_a, \quad (3.3)$$

or

$$F_a - S G_a \approx 0. \quad (3.4)$$

I define $Y \equiv F - S G$ so that

$$Y_a \approx 0. \quad (3.5)$$
The number of columns in $Y$ equals the number of unknown coefficients in the inverse wavelet $a$. For simplicity, I assume that the inverse wavelet $a$ has three coefficients: $a_0$, $a_1$, and $a_2$.

$$Y = \begin{bmatrix} y_0 & y_1 & y_2 \end{bmatrix}, \quad (3.6)$$

where

$$y_0 = f_0 - Sg_0, \quad (3.7)$$
$$y_1 = f_1 - Sg_1, \quad (3.8)$$
$$y_2 = f_2 - Sg_2. \quad (3.9)$$

Here, $f_0$ is simply $f$, $f_1$ is $f$ delayed by one sample, and $f_2$ is $f$ delayed by two samples. Likewise, $g_0$ is simply $g$, $g_1$ is $g$ delayed by one sample, and $g_2$ is $g$ delayed by two samples. Notice that in computing the matrix $Y$, the warping operator $S$ is applied multiple times to different delayed versions of $g$.

The trivial solution to equation 3.5 is $a = 0$. I eliminate this solution by setting the coefficient $a_0 = 1$. One could set $a_0$ to any value, but this would only scale the coefficients of the wavelet $c$ by $a_0$. This scale ambiguity indicates that using equation 3.5 alone does not allow the true amplitudes of $c$ and $a$ to be recovered; only their shapes can be estimated. With $a_0 = 1$, equation 3.5 becomes

$$\begin{bmatrix} y_1 & y_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \approx -y_0. \quad (3.10)$$

In equation 3.10, there are as many equations as time samples in $f$ and $g$, but only two unknowns $a_1$ and $a_2$, which leads one to use the least-squares method and minimize $\|y_0 + a_1 y_1 + a_2 y_2\|^2$. To compute $a_1$ and $a_2$, I solve the normal equations obtained by multiplying both sides of equation 3.10 by $[y_1 \ y_2]^T$:

$$\begin{bmatrix} y_1^T y_1 & y_1^T y_2 \\ y_2^T y_1 & y_2^T y_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -y_1^T y_0 \\ -y_2^T y_0 \end{bmatrix}. \quad (3.11)$$

The leftmost matrix in equation 3.11 is not Toeplitz because the matrix $S$ is a time-varying operator that is applied to delayed copies of $g$. This matrix is, however, symmetric positive semidefinite, which enables one to solve equation 3.11 by Cholesky decomposition. A similar
system of equations can be obtained for any number of coefficients in the inverse wavelet $a$, which need not be causal.

Figure 3.2: Estimated wavelet (dots) computed by the wavelet-estimation process is nearly identical to the known wavelet (red curve) in $f$ and $g$ from Figure 3.1. To make the wavelet $c$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Figure 3.3: Estimated shaping filter $h$ used to shape $Sg$ (Figure 3.1c) to $f$ (Figure 3.1a). To make the shaping filter $h$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

After estimating the inverse wavelet $a$, one can easily recover the wavelet $c$ as the filter that shapes the inverse wavelet $a$ to a unit impulse (e.g., Robinson & Treitel, 2000). Five inverse wavelet coefficients ($n_a = 5$) and 81 wavelet coefficients ($n_c = 81$) are used for this wavelet estimation. Figure 3.2 shows how the estimated wavelet (dots) lies on top of the
known wavelet (red line). Both wavelets have been normalized to have a maximum amplitude of one because the true amplitudes cannot be estimated.

Estimations of the wavelet $c$ and inverse wavelet $a$ allow one to apply the warping-with-wavelets algorithm; that is, compute $\text{CSAg}$ (Figure 3.4a). Note how the result of the warping-with-wavelets algorithm $\text{CSAg}$ (red) lies on top of $f$ (black), indicating that wavelet distortion has not occurred. Also, note that Figure 3.4 only shows portions of $f$ and $\text{CSAg}$ between 0 and 1.1 seconds. This display highlights only the samples in $f$ that correspond to samples in $g$ and, therefore, $\text{CSAg}$.

![Figure 3.4: Sequence $f$, from Figure 3.1, is in each panel above in black. The sequence $\text{CSAg}$ (a) obtained using the warping-with-wavelets algorithm and the sequence $\text{HSg}$ (b) obtained using the shaping filter $h$ are overlaid on top of $f$ in red. Only portions of $f$ and $\text{CSAg}$ are shown because this display highlights only the samples in $f$ that correspond to samples in $g$ and, therefore, $\text{CSAg}$.]

An alternative to using warping-with-wavelets to reduce wavelet distortion is to use a shaping filter to shape $Sg$ (with wavelet distortion) to $f$ (without wavelet distortion). This alternative is similar to the shaping filters designed by Bansal & Matheney (2010), except
that my shaping filter is time-invariant. A shaping filter of 81 coefficients is estimated (Figure 3.3) and convolved with $S_g$ to create $HS_g$ (Figure 3.4b). Note that the shaping filter result (Figure 3.4b) is the same as the warping-with-wavelets result (Figure 3.4a).

The shaping filter and warping-with-wavelets results are the same in Figure 3.4 because the amount of squeezing applied by $S$ is constant, as shown in Figure 3.1d. Applying constant squeezing to all samples in $g$ means that every sample has been distorted in the same way. A simple shaping filter can correct this constant wavelet distortion.

A simple shaping filter can correct this constant wavelet distortion.

Figure 3.5: A simple PP trace ($f$) (a), and a simple PS trace ($g$) (b). Both traces have the same wavelet and reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = a \ln(1 + bn)$, indicating that when one squeezes the PS trace to the PP trace, a varying amount of squeezing is applied that causes different amounts of wavelet distortion throughout the squeezed PS trace $S_g$ (c).

Thus far, the relationship between reflection coefficients in $f$ and $g$ has only required a constant amount of squeezing to align reflection coefficients in $g$ with those in $f$ (Figure 3.1). At this point, I make the relationship between reflection coefficients in $f$ and $g$ more complicated and realistic, such that the amount of squeezing required to align reflection coefficients

20
in \( g \) with those in \( f \) is time-varying (Figure 3.5). In Figure 3.5, the relationship between reflection coefficients in \( f \) and \( g \) is

\[
u[n] = a \ln(1 + bn),
\]

(3.12)

where \( a = \frac{u_m}{\ln(r_0)} \), \( b = \frac{r_0}{u_m} \ln(r_0) \), \( u_m \) is the last sample index in \( g \), \( r_0 \) is the amount of squeezing applied to the sample in \( g \) that corresponds to the first sample in \( f \), and \( r_1 \) is the amount of squeezing applied to the last sample in \( g \) (\( u_m \)).

Figure 3.6: Estimated wavelet (dots) computed by the wavelet-estimation process is nearly identical to the known wavelet (red curve) in \( f \) and \( g \) from Figure 3.5. To make the wavelet appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Figure 3.7: Estimated shaping filter \( h \) used to shape \( Sg \) (Figure 3.5c) to \( f \) (Figure 3.5a). To make the shaping filter \( h \) appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.
Work done by Compton (2014) shows that the amount of squeezing applied to a PS trace \((u[n] - u[n-1])\) to warp it to a PP trace is related to the \(V_p/V_s\) ratios in the subsurface by

\[
\frac{(u[n] - u[n-1])}{T} = \frac{V_p}{V_s} + 1 - \frac{1}{2}.
\]  

(3.13)

According to Hardage et al. (2011), a possible range of \(V_p/V_s\) ratios in the Texas Gulf Coast is 5.3 through 2.1, which corresponds to an \(r_0\) and \(r_1\) of 3.15 and 1.55, respectively. For the rest of the thesis, these values of \(r_0\) and \(r_1\) are used for any other traces that use the relation described in equation 3.12.

Figure 3.8: Sequence \(f\), from Figure 3.5, is in each panel above in black. The sequence \(CSAg\) (a) obtained using the warping-with-wavelets algorithm and the sequence \(HSg\) (b) obtained using a shaping filter are overlaid on top of \(f\) in red.

When a varying amount of squeezing (Figure 3.5d) is applied to the events in \(g\) to align them with the events in \(f\), the amount of wavelet distortion throughout \(g\) also varies. The varying amount of wavelet distortion is highlighted in Figure 3.5c, where \(Sg\) is more compressed in the beginning than at the end.

22
A varying amount of wavelet distortion does not prevent one from estimating the wavelet \( c \). In Figure 3.6, the known and estimated wavelets are represented by the red line and black dots, respectively. As with the constant amount squeezing case (Figure 3.2), the estimated wavelet lies on top of the known wavelet.

With the estimated wavelet \( c \), one can use the warping-with-wavelets algorithm to squeeze \( g \) to \( f \). Figure 3.8a shows \( f \) in black and the warping-with-wavelets result (CSAg) in red. Like in the constant amount of squeezing case, wavelet distortion does not exist in CSAg.

A shaping filter \( h \), with the same parameters of the shaping filter in Figure 3.3, is estimated to shape \( Sg \) (Figure 3.5c) to \( f \) (Figure 3.5a). The result of convolving \( h \) (Figure 3.7) with \( Sg \) is \( HSg \) (Figure 3.8b). Note that, in contrast to CSAg, \( HSg \) (red line) does not lie on top of \( f \) (black line). \( HSg \) is different from \( f \) because a reasonably sized shaping filter cannot account for the varying amount of wavelet distortion that occurs in \( Sg \) (Figure 3.5c).

### 3.2 Estimating two wavelets with the cyclic search

Wavelets in PP and PS traces are unlikely to be identical, in part, because attenuation often affects S-waves more than it does P-waves (Ursenbach et al., 2013). An example of a simple PP (\( f \)) and PS (\( g \)) trace with different wavelets is shown in Figures 3.9a and 3.9b, respectively. Traces \( f \) and \( g \) are created by convolving reflection coefficients with wavelets \( c \) (Figure 2.8a) and \( d \) (Figure 2.8b), respectively. The reflection coefficients used to create \( f \) and \( g \) have the same amplitude, but the coefficients in \( f \) are a squeezed version of the coefficients in \( g \). The relation between the coefficients in \( f \) and \( g \) is outlined in equation 3.12.

Different wavelets in the reference trace (\( f \)) and the trace to be warped (\( g \)) suggests that I should estimate two wavelets instead of one wavelet. To estimate two wavelets, I start with the warping-with-wavelets algorithm shown in equation 2.12, which accounts for \( f \) and \( g \) containing different wavelets \( c \) and \( d \), respectively. Note that this equation is nonlinear with respect to the unknown wavelet coefficients and contains the unknown wavelet \( c \) and unknown inverse wavelet \( b \) (inverse wavelet of the wavelet \( d \)). Because \( b \) is not an inverse of \( c \) (\( CB\neq I \)), I cannot eliminate the unknown wavelet \( c \) by simply multiplying both sides
of equation 2.12 by $B$ and instead iteratively compute five coefficients of $b$ ($n_b = 5$) and 81 coefficients of $c$ ($n_c = 81$).

Figure 3.9: A simple PP trace ($f$) (a), and a simple PS trace ($g$) (b). $f$ and $g$ have different wavelets ($c$ and $d$, respectively), but the same reflection coefficients. The relationship between events in the PP trace and the PS trace is $u[n] = a \ln(1 + bn)$, indicating that when one squeezes the PS trace to the PP trace, a varying amount of squeezing is applied that causes different amounts of wavelet distortion throughout the squeezed PS trace $S_g$ (c).

To start this iterative process, an initial wavelet is selected for either $c$ or $b$. In previous work, wavelet distortion is reduced by using a shaping filter to shape $S_g$ (Figure 3.9c) to a desired result, which I choose to be $f$ (Figure 3.9a). To recreate this shaping filter with the warping-with-wavelets algorithm (equation 2.12), the inverse wavelet $b$ needs to be a unit impulse ($\delta$). This initial choice of $b$ indicates that the estimated $c$ is simply the shaping filter that shapes $S_g$ to $f$.

Computing $q = S_g$ simplifies equation 2.12 to

$$f \approx C_q,$$  \hspace{1cm} (3.14)
or, equivalently,

\[ f \approx Qc, \quad (3.15) \]

where columns of the matrix \( Q \) contain delayed copies of \( q \).

---

**Figure 3.10:** Starting wavelets for the cyclic search are \( c \) (a), a shaping filter \( h \) that shapes \( Sg \) (Figure 3.9c) to \( f \) (Figure 3.9a), and \( d \) (b), a unit impulse. To make the wavelets \( c \) and \( d \) appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. Note that the wavelet \( d \) has only been interpolated with zeros to emphasize that this wavelet starts as an impulse.

In equation 3.15, there are as many equations as samples in \( f \) and \( g \), and a significantly smaller number of unknown wavelet coefficients in \( c \). Due to the relatively small number of unknowns, I estimate \( c \) using the least-squares method to minimize \( \| f - Qc \|_2^2 \). Specifically, I solve the normal equations obtained by multiplying both sides of equation 3.15 by \( Q^T \):

\[ Q^T Qc = Q^T f. \quad (3.16) \]
The matrices $Q$, $Q^\top$, and $Q^\top Q$ are Toeplitz because columns of $Q$ contain delayed copies of $q$. Therefore, the resulting wavelet $c$ is a filter that shapes the warped $g$ ($Sg$) to match $f$. In other words, the estimated wavelet $c$ is similar to the shaping filters designed by Bansal & Matheney (2010), but here is time-invariant.

At this point in the cyclic search, one iteration is complete. The current estimations of $c$ and $d$ are shown in Figures 3.10a and 3.10b, respectively. Note that the wavelet $c$ in Figure 3.10a is the same as the shaping filter $h$ that shapes $Sg$ to $f$, and the wavelet $d$ in Figure 3.10b is the inverse of a unit impulse.

With the current estimation of the wavelet $c$, one can estimate an updated inverse wavelet $b$. This is done by computing $P = CSG$, which reduces equation 2.12 to

$$f \approx Pb.$$  \hspace{1cm} (3.17)

As in equation 3.15, equation 3.17 contains as many equations as samples in $f$ and $g$ and a significantly smaller number of unknown inverse wavelet coefficients in $b$. The relatively small number of unknowns compared to equations again leads to the least-squares method and to minimize $\|f - Pa\|^2_2$. That is, solve the normal equations obtained by multiplying both sides of equation 3.17 by $P^\top$:

$$P^\top Pb = P^\top f.$$  \hspace{1cm} (3.18)

Here, the matrices $P$, $P^\top$, and $P^\top P$ are not Toeplitz because the time-varying operator $S$ in $P = CSG$ is applied to all delayed copies of $g$ in the columns of $G$. Therefore, I compute the inverse wavelet $b$ using Cholesky decomposition of $P^\top P$ instead of using faster solvers appropriate for Toeplitz matrices.

In equation 3.17, note that if one multiplies the wavelet $c$ by a constant and divides the estimated inverse wavelet $b$ by the same constant, the matrix on the right-hand side is unchanged, which means that only the shape of the inverse wavelet $b$ can be recovered, not its true amplitudes. This same logic can be applied to equation 3.15 (where one solves for the wavelet $c$), meaning that only the shape of the wavelet $c$ can be recovered, not its true
amplitudes.

To compute an updated estimate of the wavelet $c$, I use equation 3.15. Using the most recent estimate of the inverse wavelet $b$, I compute $q = S B g$ and solve equation 3.16 for the wavelet $c$. At this point, two iterations are complete.

![Figure 3.11: RMS of all residuals for each iteration in the cyclic search.](image)

Figure 3.11: RMS of all residuals for each iteration in the cyclic search.

This iterative process continues until the maximum number of iterations has been reached or the percent change in the root-mean-square (RMS) of all residuals is below 0.01 percent. In the cyclic search, the RMS of all residuals is defined as

$$\sqrt{\frac{(C S B g - f)^\top (C S B g - f)}{n_f + n_b}}$$

(3.19)

because the RMS of all residuals in the cyclic search needs to be comparable to the RMS of all residuals in the Gauss-Newton method, which is discussed in the next subsection. In equation 3.19, $n_f$ is the number of samples analyzed within $f$ and $n_b$ is the number of coefficients in $b$. The stopping condition of 0.01 percent is based on when the synthetic tests converge to the correct solution.
In this synthetic case, the stopping condition is not reached until around 160 iterations are complete, as shown in Figure 3.11. Note that the RMS of all residuals does not change significantly from the first to the last iteration. This indicates that the shaping filter $h$ (iteration 0) reduces most of the wavelet distortion, but not all of it.

Figure 3.12: Estimated $c$ (a) and $d$ (b) wavelets (dots) computed by the cyclic search are similar to the known $c$ (a) and $d$ (b) wavelets (red curve) in $f$ and $g$ from Figure 3.9. To make the wavelets $c$ and $d$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

Final estimates of the wavelets $c$ and $d$ are shown in Figures 3.12a and 3.12b, respectively. The known wavelets are shown in red, and the estimated wavelets are shown as dots. Note how the estimated wavelets are similar to the known wavelets.

With estimated wavelets $c$ and $d$, one can use the warping-with-wavelets algorithm to align events in $g$ with those in $f$. Note how the warping-with-wavelets algorithm (Figure 3.13a) reduces wavelet distortion more effectively than a shaping filter (Figure 3.13b).
The only example in this subsection contains a varying amount of squeezing, as shown in Figure 3.9d, because two wavelets cannot be estimated when the amount of squeezing is constant. A constant amount of squeezing would result in a constant amount of wavelet distortion throughout $S_g$, which, as shown in Figure 3.4b, a shaping filter $h$ can account for. Because a shaping filter $h$ can account for all wavelet distortion, the wavelet $c$ in the warping-with-wavelets algorithm for two wavelets (equation 2.12) can shape $SB_g$ to be the same as $f$, no matter what $b$ is.

Figure 3.13: Sequence $f$, from Figure 3.9, is in black in each of the above panels. The sequence $CSB_g$ (a) obtained using the warping-with-wavelets algorithm and wavelets estimated by the cyclic search is in red and overlaid onto $f$. The sequence $HS_g$ (b) obtained using a shaping filter $h$ is also in red and is overlaid onto $f$.

3.3 Estimating two wavelets with the Gauss-Newton method

The cyclic search is a simple way to solve for unknown wavelets $c$ and $b$ in the nonlinear system of equations shown in equation 2.12. A more elegant way to solve for these unknowns is the Gauss-Newton method. Nocedal & Wright (2006) describe this method as an iterative
technique that solves for unknown parameters simultaneously by minimizing a nonlinear objective function. In the case of solving for the wavelet $c$ and inverse wavelet $b$, the nonlinear objective function

$$\theta = \|r\|_2^2 \quad (3.20)$$

is minimized. In equation 3.20,

$$r = CSB g - f. \quad (3.21)$$

The general form of the normal equations solved in the Gauss-Newton method is

$$J_k^T J_k \delta_k = -J_k^T r_k, \quad (3.22)$$

where $J_k$ is the Jacobian matrix, $\delta_k$ is the update vector, $r_k$ is the residual vector, and $k$ is the iteration number (Nocedal & Wright, 2006). Note that in equation 3.22, $J_k^T J_k$ and $-J_k^T r_k$ are the approximated Hessian and the negative of the gradient of $\theta$, respectively. Also note that one does not solve for an unknown vector directly, but solves for an update vector $\delta_k$ that updates previous estimates of an unknown vector.

I use the Gauss-Newton method over Newton’s method because it has a number of advantages over Newton’s method. First, the Gauss-Newton method reduces computation time by computing an approximation to the Hessian and not the full Hessian. Second, when $J_k \delta_k \neq 0$, $\delta_k$ is always pointing in the descent direction. In Newton’s method, $\delta_k$ only points in the descent direction when the Hessian is positive-definite. Finally, the update vector $\delta_k$ minimizes the linear least-squares problem $\|J_k \delta_k + r_k\|_2^2$, which indicates that linear least-square methods can be used to solve for $\delta_k$.

To use equation 3.22 to solve for updates to unknowns $b$ and $c$, the terms $r_k$, $\delta_k$, and $J_k$ need to be defined using equation 3.21. The residual vector $r_k$ is defined as

$$r_k = C_k SB_k g - f, \quad (3.23)$$

which is simply equation 3.21. The update vector $\delta_k$ is defined as

$$\delta_k = \begin{bmatrix} \delta_{b_k} \\ \delta_{c_k} \end{bmatrix}. \quad (3.24)$$
A Jacobian matrix \( J \) contains the first partial derivatives of the residuals and is defined as
\[
J = \begin{bmatrix}
\frac{\partial r}{\partial b} & \frac{\partial r}{\partial c}
\end{bmatrix}.
\tag{3.25}
\]

Recall that \( \text{CSB} \mathbf{g} \) can be reduced to \( \mathbf{Qc} \) and \( \mathbf{Pb} \) (equation 3.15 and 3.17, respectively), which indicates that the residual vector \( \mathbf{r} \) can be rewritten as \( \mathbf{r} = \mathbf{Qc} - \mathbf{f} \) or \( \mathbf{r} = \mathbf{Pb} - \mathbf{f} \). These forms of \( \mathbf{r} \) allow one to easily reduce equation 3.25 to
\[
J = \begin{bmatrix}
\mathbf{P} & \mathbf{Q}
\end{bmatrix},
\tag{3.26}
\]
which indicates that \( J_k \) is
\[
J_k = \begin{bmatrix}
\mathbf{P}_k & \mathbf{Q}_k
\end{bmatrix}.
\tag{3.27}
\]

By using equations 3.23, 3.24, 3.27, equation 3.22 is expanded to
\[
\begin{bmatrix}
\mathbf{P}^T_k \mathbf{P}_k & \mathbf{P}^T_k \mathbf{Q}_k \\
\mathbf{Q}_k^T \mathbf{P}_k & \mathbf{Q}_k^T \mathbf{Q}_k
\end{bmatrix}
\begin{bmatrix}
\delta b_k \\
\delta c_k
\end{bmatrix} =
\begin{bmatrix}
\mathbf{P}^T_k (\mathbf{C}_k \text{SB} \mathbf{g} - \mathbf{f}) \\
\mathbf{Q}_k^T (\mathbf{C}_k \text{SB} \mathbf{g} - \mathbf{f})
\end{bmatrix}.
\tag{3.28}
\]

To compute the approximated Hessian and gradient in equation 3.28, initial estimates of \( \mathbf{b} \) and \( \mathbf{c} \) are needed. In the cyclic search, the initial estimates of \( \mathbf{b} \) and \( \mathbf{c} \) are a unit impulse and a shaping filter \( \mathbf{h} \) that shapes \( \mathbf{Sg} \) to \( \mathbf{f} \), respectively; therefore, the starting solution for \( \mathbf{b} \) and \( \mathbf{c} \) in the Gauss-method method is the same as the cyclic search’s initial estimates.

It is possible for two different sets of \( \mathbf{c} \) and \( \mathbf{b} \) to be estimated and have the same residual vector \( \mathbf{r} \). For example, one could estimate \( \mathbf{c} \) and \( \mathbf{b} \) and also \( 10 \mathbf{c} \) and \( \frac{1}{10} \mathbf{b} \). In this case, \( \mathbf{r} = 10 \text{CS} \frac{1}{10} \mathbf{B} \mathbf{g} - \mathbf{f} = \text{CSB} \mathbf{g} - \mathbf{f} \). To eliminate this scale factor ambiguity, I set the unknown coefficient \( b_0 \) to 1 and modify equation 3.28, so that this coefficient remains at 1 for all iterations. Note that \( b_0 \) can be set to any value. If one were to set \( b_0 \) to 10, then \( \mathbf{c} \) would be scaled by \( \frac{1}{10} \), which indicates that the true amplitudes of \( \mathbf{c} \) and \( \mathbf{b} \) cannot be estimated.

To keep \( b_0 \) equal to 1 for all iterations, \( \delta b_0 \) needs to be 0 for all iterations. Below is an example where I assume that the wavelet \( \mathbf{c} \) and the inverse wavelet \( \mathbf{b} \) have only two coefficients and are both causal; then equation 3.28 can be expanded to
Because $\delta_{b0} = 0$ is located in the first row of the update vector, the first row and column of the approximated Hessian are set to zero, except for the element on the diagonal; the first element of the gradient is also set to zero. These modifications reduce equation 3.29 to

$$\begin{bmatrix} p_0^\top p_0 & 0 & 0 & 0 \\ 0 & p_1^\top p_1 & p_1^\top q_0 & p_1^\top q_1 \\ 0 & q_0^\top p_1 & q_0^\top q_0 & q_0^\top q_1 \\ q_1^\top p_0 & q_1^\top p_1 & q_1^\top q_0 & q_1^\top q_1 \end{bmatrix} \begin{bmatrix} \delta_{b0} \\ \delta_{b1} \\ \delta_{c0} \\ \delta_{c1} \end{bmatrix} = - \begin{bmatrix} p_1^\top r \\ p_1^\top q \\ q_0^\top r \\ q_1^\top r \end{bmatrix}. \quad (3.30)$$

Once $b_0$ is constrained to 1, I can use equation 3.28 to solve for $\delta_{b_k}$ and $\delta_{c_k}$. Note that the approximated Hessian in equation 3.28 is a symmetric positive semidefinite matrix, and thus one could use Cholesky decomposition to solve for $\delta_{b_k}$ and $\delta_{c_k}$. This $\delta_k$ will be pointing in the descent direction of the nonlinear objective function, but it may not decrease equation 3.20 with each iteration. Therefore, a line search is used to determine the fraction of $\delta_k$ that will be used for the update of $c$ and $b$. This line search determines an $\alpha$, between 0 and 1, such that a decrease in the nonlinear objective function (equation 3.20) occurs with each iteration. Once an $\alpha$ is chosen, previous estimates of $b$ and $c$ are updated, so that the new estimates of $b$ and $c$ are

$$b_{k+1} = b_k + \alpha \delta_{b_k} \quad c_{k+1} = c_k + \alpha \delta_{c_k}. \quad (3.31)$$

Recall that the cyclic search stops when the maximum number of iterations has been reached or when the percent change in the RMS of all residuals is below 0.01 percent. Even though the Gauss-Newton method has the same stopping condition as the cyclic search, it does not end when the stopping condition is met. Instead of stopping the Gauss-Newton method, I penalize the outer lags of the inverse wavelet $b$, which drives the coefficients of the outer lags of $b$ towards zero. To understand why this is reasonable to do, consider the following scenario where I have estimated the true inverse wavelet $b$. If I increase the number of coefficients by one and re-estimate the inverse wavelet $b$, because the last estimate was
the true estimate, the value of the new coefficient is zero. By penalizing the outer lags of \( b \) and driving these coefficients towards zero, I create a situation that occurs when the true wavelet is known.

Outer lags in \( b \), but not in \( c \), are penalized because the starting inverse wavelet \( b \) is an impulse, which only has one non-zero coefficient. The other coefficients, that are equal to zero, can be anything, which is why I choose to control these coefficients by penalizing them.

To incorporate penalizing outer lags of \( b \) into equation 3.28, the residual vector \( r \) is redefined as

\[
r = \begin{bmatrix} r_d \\ \sqrt{\gamma} r_p \end{bmatrix}.
\] (3.32)

In equation 3.32, \( r_d \) is the data residual vector \( (r_d = CSBg - f) \) and \( r_p \) is the penalization residual vector, where \( r_p = Xb \), \( X \) is a scaled diagonal matrix, and \( \gamma \) is the degree at which the outer lags of \( b \) are penalized. The diagonal elements of the matrix \( X \) are equal to the absolute value of the lag between the coefficient of \( b \) that the diagonal element in \( X \) is multiplied by in \( r_p \) and the coefficient of \( b \) corresponding to zero seconds. Expanding equation 3.20 using the newest definition of the residual vector yields

\[
\theta = r_d^\top r_d + \gamma r_p^\top r_p,
\] (3.33)

which can be further expanded to

\[
\theta = (CSBg - f)^\top (CSBg - f) + \gamma b^\top X^\top X b.
\] (3.34)

The magnitude of \( b^\top X^\top X b \) is not comparable to the magnitude of \( (CSBg - f)^\top (CSBg - f) \). To make the magnitude of those terms comparable, I multiply each diagonal element of \( X \) by \( \sqrt{f^\top f} \). The new definition of the residual vector simply modifies equation 3.28 to

\[
\begin{bmatrix}
P_k^\top P_k + \gamma X^\top X & P_k^\top Q_k \\
Q_k^\top P_k & Q_k^\top Q_k
\end{bmatrix} \begin{bmatrix} \delta_{bk} \\ \delta_{ck} \end{bmatrix} = - \begin{bmatrix}
P_k^\top (C_k SB_k g - f) + \gamma X^\top X b_k \\
Q_k^\top (C_k SB_k g - f)
\end{bmatrix}.
\] (3.35)

To determine the value of \( \gamma \), a line search is used. With each \( \gamma \) value tested, equation 3.35 is solved for \( \delta_{b} \) and \( \delta_{c} \) and an appropriate \( \alpha \) is chosen as previously described. After \( \delta_{b} \), \( \delta_{c} \), and \( \alpha \) are computed, \( c \) and \( b \) are updated using equation 3.31. The \( \gamma \) that produces
an updated $c$ and $b$ that yield the smallest value of equation 3.33 is selected for the next iteration of the Gauss-Newton method.

Too large of a $\gamma$ can increase the value of equation 3.33 past its starting value in the iterative process ($r_{d_0}^T r_{d_0}$). Thus, I set the maximum value of $\gamma$ to

$$
\gamma_{\text{max}} = \frac{1}{2} \frac{r_{d_0}^T r_{d_0} - r_{d_{bp}}^T r_{d_{bp}}}{r_{bp}^T r_{bp}},
$$

(3.36)

where $r_{d_0}^T r_{d_0}$ is the two norm squared of the data residual from the starting iteration, $r_{d_{bp}}^T r_{d_{bp}}$ is the two norm squared of the data residual before penalization, and $r_{bp}^T r_{bp}$ is the two norm squared of the penalization residual calculated with the estimate of $b$ acquired before penalization. This definition of $\gamma_{\text{max}}$ prevents the value of equation 3.33 from increasing past the midpoint between its value at iteration 0 ($r_{d_0}^T r_{d_0}$) and its value before penalization ($r_{d_{bp}}^T r_{d_{bp}}$).

$\gamma = 0$ does not penalize the outer lags of $b$, so a lower bound of $\gamma$ is also calculated:

$$
\gamma_{\text{min}} = \frac{1}{10} \gamma_{\text{max}}.
$$

(3.37)

This ensures the outer lags of $b$ are penalized when required.

Once $\gamma$ is found, the Gauss-Newton iterations continue to use equation 3.35, and $\gamma$ is chosen only once. When the stopping condition is met again, $\gamma$ is set to 0 to try and drive down the value of equation 3.33 even further. When the stopping condition is met for a third time, the Gauss-Newton method officially ends. At this point, the method confirms whether or not the value of equation 3.33 at the end of the method is lower than the value reached before penalizing the outer lags in $b$. If the former is lower than the latter, the final $b$ and $c$ are the wavelets returned by the method; otherwise, the $b$ and $c$ achieved before penalization are returned.

If the approximated Hessian in equations 3.28 or 3.35 become not positive definite, due to rounding errors, the penalization scheme discussed previously can be used; this is similar to adding values to the diagonal of the matrix to make the matrix positive definite again.

Recall that an example of a simple PP ($f$) and PS ($g$) trace with different wavelets is shown in Figures 3.9a and 3.9b, respectively. Traces $f$ and $g$ are created by convolving the
same reflection coefficients with wavelets \( c \) (Figure 2.8a) and \( d \) (Figure 2.8b), respectively. The relation between reflection coefficients in \( f \) and \( g \) is outlined in equation 3.12.

The starting PP (c) and PS (d) wavelets used for the Gauss-Newton method are the same as the those for the cyclic search and are shown in Figures 3.10a and 3.10b, respectively. These starting wavelets result in the first RMS of all residuals value shown in Figure 3.14. Note that in the Gauss-Newton method the RMS of all residuals is defined as

\[
\text{RMS} = \sqrt{\frac{(\text{CSB} - f)^\top (\text{CSB} - f) + \gamma r_p^\top r_p}{n_f + n_b}}.
\]  

(3.38)

Also note that, as with the cyclic search, 81 unknown coefficients in \( c \) \((n_c = 81)\) and five unknown coefficients in \( b \) \((n_b = 5)\) are estimated.

As discussed previously, the Gauss-Newton method can be divided into three stages (shown in Figure 3.14): before-penalization, during-penalization, and after-penalization. The before-penalization stage spans from iteration 0 to 180. This stage exists for most of the iterations because the stopping condition has not been reached until iteration 180.
during-penalization stage exists from iteration 181 to 182, and starts with a large spike in the RMS of all residuals. This spike is due to the fact that the outer lags in the inverse PS wavelet $b$ are penalized. The calculation of the RMS of all residuals for iteration 181 uses the $b$ and $c$ from iteration 180 and the chosen $\gamma$, so that it is possible to observe the increase in the RMS of all residuals due to penalizing outer lags of $b$. After the initial spike, the penalized Gauss-Newton method iterates until the stopping condition is met, which occurs quickly. The after-penalization stage spans from iteration 183 to 185. The sudden drop in the

![Graph](image)

Figure 3.15: Estimated $c$ (a) and $d$ (b) wavelets (dots) computed by the Gauss-Newton method are nearly identical to the known $c$ (a) and $d$ (b) wavelets (red curve) in f and g from Figure 3.9. To make the wavelets $c$ and $d$ appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.

RMS of all residuals occurs over one iteration because the RMS of all residuals for iteration 183 is calculated using $\gamma = 0$ and the $b$ and $c$ from the previous iteration. The third stage also ends quickly. The final RMS of all residuals in the after-penalization stage is smaller
than that in the before-penalization stage, which indicates that the $b$ and $c$ determined in the after-penalization stage are returned by the Gauss-Newton method.

The during-penalization stage allows the Gauss-Newton method to escape any perceived local minimum encountered by the stopping condition. By including the penalization residual, I am attempting to fill in any local minimum holes or pass regions of the nonlinear objective function that have a small gradient. In this synthetic case, the during-penalization stage is not effective at decreasing the RMS of all residuals.

The completion of the Gauss-Newton method yields estimates of the PP ($c$) and PS ($d$) wavelets, as shown in Figure 3.15. Note how the estimated wavelets lie on top of the known wavelets.

![Figure 3.16: Sequence $f$, from Figure 3.9, is in black in each of the above panels. Sequence CSBg (a) obtained using the warping-with-wavelets algorithm and wavelets estimated by the Gauss-Newton method is in red and overlaid onto $f$. Sequence HSg (b) obtained using a shaping filter $h$ is also in red and are overlaid onto $f$.](image)

With the estimated wavelets $c$ and $d$, one can use the warping-with-wavelets algorithm to align events in $g$ with those in $f$. As the plot of the RMS of all residuals indicates...
(Figure 3.14), the shaping filter result reduces most of the wavelet distortion (Figure 3.16b), while the warping-with-wavelets algorithm reduces all distortion (Figure 3.16a). This is similar to the result obtained with the cyclic search.

3.4 Comparison of cyclic search and Gauss-Newton method

Two methods have been proposed to estimate two wavelets. The cyclic search is a simple way to solve nonlinear systems, while the Gauss-Newton method is a more common way. Both methods have their advantages and disadvantages in terms of time to complete one iteration and the ability to estimate wavelets in the presence of noise.

Figure 3.17: Sequence \( f \) with a noise-to-signal ratio of 0.0 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e), and 1.0 (f) and sequence \( g \) with a noise-to-signal ratio of 0.0 (g), 0.2 (h), 0.4 (i), 0.6 (j), 0.8 (k), and 1.0 (l).
The normal equations solved in the cyclic search are shown in equations 3.16 and 3.18. The size of the matrices $Q^\top Q$ and $P^\top P$ in these normal equations are $n_c \times n_c$ and $n_b \times n_b$, respectively. Because $Q^\top Q$ is a Toeplitz matrix, the corresponding normal equations can be solved by a fast Toeplitz solver (Levinson recursion) where the computational cost is proportional to $n_c^2$ floating point operations (flops) (Robinson & Treitel, 2000). The matrix $P^\top P$ is not Toeplitz, but is symmetric positive semidefinite, indicating that the corresponding normal equations can be solved by Cholesky decomposition. To solve equation 3.18 with Cholesky decomposition involves a computational cost proportional to $\frac{1}{3}n_b^3$ flops (Trefethen & Bau, 1997).

Figure 3.18: Known c (a) and d (b) wavelets represented by the red line and the cyclic search estimated c and d wavelets represented by the black, green, blue, cyan, pink, and orange dots from f and g with noise-to-signal ratios of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, respectively. To make the wavelets c and d appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.
The normal equations solved in the Gauss-Newton method are shown in equation 3.35. The approximated Hessian is a symmetric positive semidefinite matrix indicating that these normal equations can be solved by Cholesky decomposition, which has a computational cost proportional to $\frac{1}{3}(n_b + n_c)^3$ flops. Because the Gauss-Newton method has a computational cost proportional to $\frac{1}{3}(n_b + n_c)^3$ and the cyclic search has a computational cost proportional to only $\frac{1}{3}n_b^3 + n_c^2$ flops, each cyclic search iteration is faster than a Gauss-Newton iteration.

![Figure 3.19: Known c (a) and d (b) wavelets represented by the red line and the Gauss-Newton method estimated c and d wavelets represented by the black, green, blue, cyan, pink, and orange dots from f and g with noise-to-signal ratios of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, respectively. To make the wavelets c and d appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.](image-url)

I have analyzed the cyclic search and the Gauss-Newton method with the assumption that the noise-to-signal ratio is zero, where the noise-to-signal ratio is the ratio of RMS of the noise and the RMS of the signal. Without this assumption, one can add bandlimited
noise to a trace and increase its noise-to-signal ratio.

In Figure 3.17, six different pairs of \( f \) and \( g \) are shown. The \( f \) and \( g \) seen in Figures 3.17a and 3.17g, respectively, have a noise-to-signal ratio of zero, and thus are the same \( f \) and \( g \) seen in Figure 3.9. Figure 3.17 shows \( f \) and \( g \) with a noise-to-signal ratio of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0.

The corresponding estimates of the wavelets \( c \) and \( d \) from an \( f \) and \( g \) with different noise-to-signal ratios are seen in Figure 3.18, using the cyclic search, and in Figure 3.19, using the Gauss-Newton method. The known wavelets in Figures 3.18 and 3.19 are represented by the red solid line, and the estimated wavelets for different noise-to-signal ratios are represented by dots. Note that the estimates of \( c \) and \( d \) using the Gauss-Newton method (Figure 3.19) appear to be more concentrated over the known wavelets than the estimates from the cyclic search (Figure 3.18).
CHAPTER 4
APPLICATION TO PP-PS IMAGES

In this chapter, real PP and PS images are used to test the warping-with-wavelets algorithm, and the cyclic search and the Gauss-Newton wavelet estimation techniques. To use the warping-with-wavelets algorithm to warp images, let the PP and PS images be represented as column vectors \( f \) and \( g \), respectively. Before estimating the PP and PS wavelets, the noise in the PS image is reduced and the amplitudes in the PP and PS images are made comparable. The estimated PP and PS wavelets are used in this algorithm to squeeze the PS image without wavelet distortion, and this squeezed image is compared with the image produced by using a shaping filter to reduce wavelet distortion.

4.1 Data Conditioning

Subsets of real PP and PS images are shown in Figures 4.1a and 4.1b, respectively. The

Figure 4.1: PP image (a) and a PS image (b). Both are provided courtesy of SINOPEC.
images are plotted in a way that aligns events in the PP image $f$ with those in the PS image $g$. Note how the PS image $g$ is noisier than the PP image $f$. Ursenbach et al. (2013) point out that PS images tend to have a lower frequency content than PP images because attenuation often affects S-waves more than it does P-waves. This lack of frequency content often results in poorer resolution in the PS image than in the PP image.

Before PP and PS wavelets are estimated, I apply a recursive exponential smoothing filter horizontally across the PS image to reduce noise. The resulting image is seen in Figure 4.2b and is less noisy than the original PS image (Figure 4.2a). I then make the PP and PS amplitudes comparable by applying a time-varying gain to both the PP and PS images, which makes the RMS of the amplitudes of each image equal to one.

Figure 4.2: Noisy PS image (a) and the result of applying a recursive exponential filter (b) to reduce the noise. All images have been scaled for display to have the same RMS amplitude.
4.2 Wavelets estimated with the cyclic search

A method that estimates PP and PS wavelets is the cyclic search. With this method, I estimate 81 PP wavelet coefficients \( n_c = 81 \) and 23 inverse PS wavelet coefficients \( n_b = 23 \). Figures 4.3a and 4.3b show initial estimates of the PP and PS wavelets after the first iteration. Recall that initial estimates of the PP wavelet \( c \) and the PS wavelet \( d \) are equivalent to the shaping filter \( h \) that shapes the squeezed PS image \( S_g \) to the PP image \( f \) and the inverse of a unit impulse, respectively. In the rest of this chapter, I construct the squeezing operator \( S \) using time shifts found by smooth dynamic warping (Compton & Hale, 2013).

Figure 4.3: Initial PP (a) and PS (b) wavelets for cyclic search and Gauss-Newton method. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients. Note that the PS wavelet has only been interpolated with zeros to emphasize that this wavelet starts as an impulse.
Figure 4.4: RMS of all residuals for each iteration in the cyclic search.

Figure 4.5: Estimated PP (a) and PS (b) wavelets using cyclic search. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.
Figure 4.6: Amplitude spectra of the PP (a) and PS (b) wavelets estimated using the cyclic search, and amplitude spectra of the PP (c) and PS (d) images.
A plot of the RMS of all residuals with iterations shows that only a small decrease in the residuals occurs (Figure 4.4), which indicates that the shaping filter $h$ reduces most of the wavelet distortion. The final PP and PS wavelets estimated after 23 iterations are shown in Figures 4.5a and 4.5b, respectively. Note how the final PP wavelet $c$ (Figure 4.5a) is similar to the shaping filter $h$ (Figure 4.3a). Also notice that amplitude spectra of the PP wavelet $c$ and the PS wavelet $d$ (Figures 4.6a and 4.6b, respectively) are similar to the amplitude spectra of the PP and PS images (Figures 4.6c and 4.6d, respectively).

### 4.3 Wavelets estimated with the Gauss-Newton method

Another way to estimate PP and PS wavelets is the Gauss-Newton method. Recall that the initial wavelets used to start the Gauss-Newton method are the shaping filter $h$ that shapes the squeezed PS image $S_g$ to the PP image $f$ (Figure 4.3a) and the inverse of the unit impulse (Figure 4.3b).

The final RMS of all residuals for the Gauss-Newton method (Figure 4.7) is smaller than the final RMS of all residuals for the cyclic search (Figure 4.4). The before-, during-, and after-penalization stages span from iterations 0 to 1, 2 to 38, and 39 to 40, respectively. Before-penalization and after-penalization stages are only one iteration long because the stopping condition is met quickly. Recall that the during-penalization stage is designed

![Figure 4.7: RMS of all residuals for each iteration in the Gauss-Newton method.](image)
Figure 4.8: RMS of all residuals for each iteration in the Gauss-Newton method with a stopping condition of 0.0001 percent.

Figure 4.9: Estimated PP (a) and PS (b) wavelets using Gauss-Newton method. To make the PP and PS wavelets appear more continuous in this figure, I use interpolation to increase the estimated 81 coefficients to 321 coefficients.
Figure 4.10: Amplitude spectra of the PP (a) and PS (b) wavelets estimated using the Gauss-Newton method, and amplitude spectra of the PP (c) and PS (d) images.
to navigate around local minima or portions of the objective function that have a small gradient. To see if the during-penalization stage helps decrease the RMS of all residuals faster, I decrease the stopping condition of the Gauss-Newton method to 0.0001 percent (Figure 4.8). The before-, during-, and after-penalization stages now span from iterations 0 to 95, 96 to 103, and 104 to 109, respectively. Note that if the during-penalization stage is not started early in the iterative process, the RMS of all residuals decreases slowly. Also note that there is not a large difference in the final RMS of all residuals in Figures 4.7 and 4.8, which indicates that the decrease in the RMS of all residuals per iteration is less when the stopping condition is equal to 0.0001 percent than when the stopping condition is equal to 0.01 percent.

After 41 iterations, the Gauss-Newton method estimates the PP and PS wavelets (Figures 4.9a and 4.9b, respectively). As with the cyclic search, notice how the estimated PP wavelet $c$ (Figures 4.9a) is similar to the shaping filter $h$ (Figures 4.3a). The amplitude spectra of the PP wavelet $c$ and PS wavelet $d$ estimated by the Gauss-Newton method (Figures 4.10a and 4.10b, respectively) have different shapes than the amplitude spectra of the PP wavelet $c$ and PS wavelet $d$ estimated by the cyclic search (Figures 4.6a and 4.6b, respectively). These differences are due to the Gauss-Newton method driving down the RMS of all residuals more than the cyclic search and also due to penalizing the outer coefficients in the inverse PS wavelet $b$.

### 4.4 Warping-with-wavelets result compared with the shaping filter result

Simply squeezing the PS image $g$ to the PP image $f$ results in a squeezed PS image $Sg$ (Figure 4.11c) that contains wavelet distortion. The wavelet distortion at early samples within the squeezed PS image $Sg$ is more extreme than at other samples within $Sg$ because the early samples had a higher amount of squeezing applied to them as shown by the $V_p/V_s$ ratio image (Figure 4.13). Horizontal rows of similar $V_p/V_s$ ratios appear in Figure 4.13 because the time shifts estimated by smooth dynamic warping, which are used to calculate $V_p/V_s$ ratios, are estimated every 0.2 seconds (Compton, 2014).
To reduce the wavelet distortion in the squeezed PS image \( S_g \), the shaping filter \( h \) (Figure 4.3a) is used to shape the squeezed PS image \( S_g \) (Figure 4.11c) to the PP image \( f \) (Figure 4.11a). The application of this shaping filter yields the shaped and squeezed PS image \( H S_g \) (Figure 4.11b). Note that there are no large differences between the squeezed PS image \( S_g \) and the shaped and squeezed PS image \( H S_g \). The reason for the lack of significant differences is that there is not a large variation in the estimated \( V_p/V_s \) ratios with time (Figure 4.13). The largest variations in the \( V_p/V_s \) ratios occur between 0.4 and 0.8 seconds and between 1.8 and 2.0 seconds. To observe the slight differences between the squeezed PS image \( S_g \) and the shaped and squeezed PS image \( H S_g \), I only examine the
data within 0.4 and 0.8 seconds.

Figure 4.12 is Figure 4.11, except that only samples between 0.4 and 0.8 seconds are shown. At this small scale, one can observe the differences between the squeezed PS image $S_g$ (Figure 4.12c) and the shaped and squeezed PS image $H S_g$ (Figure 4.12b). These differences are due to the reduction in wavelet distortion caused by applying the shaping filter $h$.

![Figure 4.12](image)

Figure 4.12: PP image (a) with PS images obtained by applying a shaping filter after warping (b), and simply warping the PS image (c). Note that only data between 0.4 and 0.8 seconds are shown. All images have been scaled for display to have the same RMS amplitude.

By applying the shaping filter $h$ to the squeezed PS image $S_g$, most, but not all, of the wavelet distortion is reduced. The wavelet distortion is reduced further by using the warping-with-wavelets algorithm. Recall that the warping-with-wavelets algorithm is deconvolving
the PS wavelet $d$ from the PS image $g$, squeezing the deconvolved PS image $Bg$ to PP time, and convolving the squeezed and deconvolved PS image $SBg$ with the PP wavelet $c$. By convolving $SBg$ with the PP wavelet $c$, remaining differences between the PP image $f$ and the warping-with-wavelets result $CSBg$ are mainly due to variations in PP and PS reflectivity. Because I have two sets of estimated PP and PS wavelets, I apply this algorithm to generate two warped PS images $CSBg$ using the wavelets estimated by the cyclic search and Gauss-Newton method (Figures 4.14b and 4.14c, respectively). Because both sets of wavelets produce similar RMS of all residuals, there is little difference between the two warping-with-wavelets results.

Figure 4.13: $V_p/V_s$ ratios estimated using smooth dynamic warping.
Although both warping-with-wavelets results look similar, they were created by using different PP and PS wavelets. To attempt to determine which set of wavelets best matches the true wavelets, I use the wavelets estimated by the cyclic search and the Gauss-Newton method to calculate the RMS of the difference between $\text{CSB}_g$ and $f$ and compare these RMS values. The lowest RMS value is reached by using the wavelets estimated by the Gauss-Newton method, as shown in Figures 4.4 and 4.7. I also examine how well the estimated inverse PS wavelet $b$ is able to deconvolve the PS image $g$. I expect the reflectors in a deconvolved PS image to look more impulsive than how they appear in a PS image. Convolving the PS image $g$ with the inverse PS wavelet $b$, estimated by the Gauss-Newton
method, produces a deconvolved PS image $\mathbf{B}_g$ (Figure 4.15b). This deconvolved image contains several reflectors that are more impulsive than the corresponding reflectors in the original PS image $\mathbf{g}$; most of these reflectors exist between 0.9 and 1.9 seconds. If the inverse PS wavelet $\mathbf{b}$, estimated by the cyclic search, is used to deconvolve the PS image $\mathbf{g}$, reflectors are barely visible because an abundance of high frequencies exist (Figure 4.15c).

![Figure 4.15: PS image (a) with deconvolved PS images obtained by deconvolving the PS wavelet estimated by the Gauss-Newton method (b) and the cyclic search (c). All images have been scaled for display to have the same RMS amplitude.](image)

A deconvolved image should also have a whitened amplitude spectrum. In Figures 4.16a, 4.16b, and 4.16c are the amplitude spectra of the PS image $\mathbf{g}$, the deconvolved PS image $\mathbf{B}_g$ constructed using the estimated PS wavelet $\mathbf{b}$ from the Gauss-Newton method, and the deconvolved PS image $\mathbf{B}_g$ constructed using the estimated $\mathbf{b}$ from the cyclic search,
respectively.

Note how the frequencies in the spectrum of the $B_g$ related to the Gauss-Newton method (Figure 4.16b) appear to be more evenly distributed than the frequencies in the spectrum of the $B_g$ related to the cyclic search (Figure 4.16c). This observation also suggests that the PP and PS wavelets estimated by the Gauss-Newton method are better representations of the true PP and PS wavelets than the wavelets estimated by the cyclic search.

Figure 4.16: Amplitude spectra of the PS image (a), the PS image convolved with the inverse PS wavelet $b$ estimated by the Gauss-Newton method (b), and the PS image convolved with the inverse PS wavelet $b$ estimated by the cyclic search (c).
Variations between the warping-with-wavelets results (Figures 4.14b and 4.14c) and the shaping filter result (Figures 4.14d) are due to the ability of the warping-with-wavelets algorithm to better reduce wavelet distortion. Most of these variations exist between 0.4 and 0.8 seconds because, within this interval, the shaping filter $h$ cannot account for the wavelet distortion produced by higher than average $V_p/V_s$ ratios.
In this thesis, I presented two warping-with-wavelets algorithms to minimize wavelet distortion caused by warping. One algorithm assumes a single wavelet for both the PP and PS images and the other algorithm assumes different wavelets in those images. In the case of different wavelets, the warping-with-wavelets algorithm deconvolves the PS wavelet from the PS image, reducing the image to a set of impulses, these impulses are then squeezed without distortion, and are convolved with the PP wavelet. By convolving the squeezed impulses with the PP wavelet, the differences between the PP image and the warped PS image are due to differences in the PP and PS reflection coefficients.

To estimate the PP and PS wavelets to use in the warping-with-wavelets algorithm, a nonlinear objective function is minimized using either a cyclic search or the Gauss-Newton method. This nonlinear objective function is constructed by using the warping-with-wavelets algorithm. Whether the PP and PS wavelets are estimated with a cyclic search or a Gauss-Newton method, the resulting warped PS image contains less wavelet distortion than simply using a shaping filter to shape the warped PS image to some desired result.

Although the focus of this thesis is on reducing wavelet distortion caused by squeezing a PS image to PP time, the warping-with-wavelets algorithm can be applied to any geophysical transformation that involves squeezing or stretching, an example would be NMO correction. In each application of this algorithm, a reduction in wavelet distortion occurs and estimates of the wavelets in the image to be warped and the reference image are obtained.

5.1 Future Work

The current implementation of the warping-with-wavelets algorithm does not account for attenuation, meaning that the algorithm assumes that a seismic image is the result of convolving one wavelet with reflection coefficients. Because attenuation decreases the
frequency content of the wavelet with time, a seismic image contains multiple wavelets. To estimate several wavelets using the cyclic search or the Gauss-Newton method, multiple subsets of the image are created, so that a single set of wavelets is estimated with each subset. Because the amount of warping applied within each subset is likely to be less than the amount of warping applied throughout the entire image, estimating two wavelets becomes more difficult. Future implementations of this method should account for attenuation by using some other technique than estimating wavelets of several subsets of the image.

To estimate the PP and PS wavelets with the cyclic search or Gauss-Newton method, 81 coefficients of the PP and PS wavelets and 23 coefficients of the inverse PS wavelet are estimated. 81 wavelet coefficients were estimated because the length of the wavelet spans several reflectors within the seismic data. 23 inverse wavelet coefficients were estimated because this number of coefficients produced a significant decrease in the RMS of all residuals and wavelets that looked realistic. A more concrete way to determine how many inverse wavelet coefficients to estimate needs to be developed.
REFERENCES CITED


