Fall 2015

INVESTIGATION OF AMBIENT SEISMIC NOISE USING SEISMIC INTERFEROMETRY IN WESTERN MONTANA

Natalia Krzywosz

Montana Tech of the University of Montana

Follow this and additional works at: http://digitalcommons.mtech.edu/grad_rsch

Part of the Geophysics and Seismology Commons

Recommended Citation

http://digitalcommons.mtech.edu/grad_rsch/57

This Thesis is brought to you for free and open access by the Student Scholarship at Digital Commons @ Montana Tech. It has been accepted for inclusion in Graduate Theses & Non-Theses by an authorized administrator of Digital Commons @ Montana Tech. For more information, please contact sjuskiewicz@mtech.edu.
INVESTIGATION OF AMBIENT SEISMIC NOISE USING SEISMIC INTERFEROMETRY IN WESTERN MONTANA

by

Natalia Krzywosz

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Geophysical Engineering

Montana Tech
2015
Abstract

Passive seismic interferometry is a process by which ambient noise data recorded at different seismic stations can be cross-correlated to estimate Green's functions. In the past, both surface waves and body waves have successfully been extracted by cross-correlation of ambient noise data on both regional and global scales. In this study, I have generated Matlab code to simulate an application of seismic interferometry on a synthetic model with pre-defined layers and p-wave velocities. For areas with known velocity models, the Matlab code produced in this study can be used to generate synthetic seismograms, and model the effects of cross-correlation on receiver responses. In order to develop a general understanding of the ambient noise wavefield in western Montana, a spectral analysis program was developed in Matlab. This program is used to process ambient noise data from the Transportable Array (TA) Seismographic Network, and to generate its power spectral density plots and probability density functions. The detailed spectral analysis provides some insight to the ambient noise sources, and their energy distribution throughout western Montana. In addition, an attempt was made to preprocess ambient noise data from the TA array in Matlab for later use. Although preprocessing of the data was successful, limitations in computing power and time, allowed for temporal stacking of only one month of data. The one month period was not long enough to produce Green's functions which contain coherent body waves.

Keywords: Seismic interferometry, ambient seismic noise, spectral analysis, cross-correlation, synthetic seismograms.
Acknowledgements

I would like to acknowledge my advisor Dr. Khalid Miah, and committee members Dr. Marvin Speece and Michael Stickney for their contributions to my understanding in seismic processes and the success of this project. Without their help this project would not be possible. I would like to thank Curtis Link for his efforts in helping me choose viable ambient noise data for use in seismic interferometry. This project would not be possible without the information and data made available through the Incorporated Research Institutions for Seismology (IRIS). Lastly, I would like to thank my friends and family for their continued support throughout the completion of this thesis.
Table of Contents

ABSTRACT ................................................................................................................. II

ACKNOWLEDGEMENTS ......................................................................................... III

LIST OF FIGURES ................................................................................................. VI

LIST OF EQUATIONS ............................................................................................ IX

1. INTRODUCTION ........................................................................................................ 1

2. BACKGROUND .......................................................................................................... 4

  2.1. Seismic Interferometry: Power Conservation and Power Reciprocity Theory used in Demonstrating Concepts of SI .............................................................. 5

  2.2. Spectral Analysis ................................................................................................... 9

  2.3. Applications of Seismic Interferometry in Ambient Noise Data ............................. 12

3. METHOD ..................................................................................................................... 16

  3.1. Seismic Interferometry on a Given Geologic Model ........................................... 16

  3.2. Data Selection ..................................................................................................... 23

  3.3. Spectral analysis of Ambient Noise Data in western Montana ............................ 25

  3.4. Preprocessing of Ambient Noise Data ................................................................. 39

  3.5. Generating Estimated Green’s Functions .............................................................. 47

4. RESULTS ..................................................................................................................... 49

  4.1. Spectral Analysis ................................................................................................. 49

  4.2. Green’s Functions Estimation .............................................................................. 57

5. CONCLUSIONS ......................................................................................................... 63

6. FUTURE WORK ......................................................................................................... 65

7. REFERENCES ............................................................................................................. 67

8. APPENDIX A: ADDITIONAL EQUATIONS AND DETAILS REGARDING GENERATION OF PSD’S. ................................................................. 71

9. APPENDIX B: PRELIMINARY CROSS CORRELATION RESULT ........................... 73
10. APPENDIX C: SUPPORT FIGURES ........................................................................................................ 74
11. APPENDIX D: MATLAB CODE .............................................................................................................. 77
   11.1. Concepts of Seismic Interferometry Used on Synthetic Data .......................................................... 79
   11.2. Data Retrieval .................................................................................................................................. 88
   11.3. PSD .................................................................................................................................................. 90
   11.4. PDF .................................................................................................................................................. 96
   11.5. Preprocessing for Ambient Noise Cross-correlations ................................................................. 103
   11.6. Cross-correlations and Stacking ..................................................................................................... 109
List of Figures

Figure 1. Reflection and transmission ray paths for a horizontally layered medium. ........5
Figure 2. Probability Density Function (McNamara & Buland, 2003). .........................10
Figure 3. PDF generated by Berger (2004) for analysis of global ambient noise (Berger & Davis, 2004). ..................................................................................................................12
Figure 4. Shows the changes made to the velocity model for western Montana since 1984 (Zeiler, et al., 2005). ..................................................................................................................17
Figure 5. The p-wave velocities are shown for models developed by Cammarano et al., 2005, PREM, and the spherically symmetric Earth model AK135 (Frost, 2008). ..........17
Figure 6. Geometry and ray tracing of a hypothetical geologic model showing layers of the Earth for western Montana up to the bottom of the transition zone. ...................... 19
Figure 7. Receiver responses generated through amplitude versus offset modeling. ........20
Figure 8. Geometry and ray tracing of a new hypothetical geologic model showing layers of the Earth for western Montana up to the bottom of the transition zone. ..............21
Figure 9. The receiver response and interferometric response for the geologic model shown in Figure 8. ..................................................................................................................22
Figure 10. Comparison between the interferometric trace (red) and the directly modeled trace (blue) at 300 km inter-station distance. .................................................................23
Figure 11. Transportable Array (TA) network map (Incorporated Research Institutions for Seismology, 2015) .................................................................................................24
Figure 12. Station map for the stations used in this study. .............................................25
Figure 13. Raw data from TA station F15A on January 15, 2009. ...............................27
Figure 14. First segment of raw data from TA station F15A on January 15, 2009. ........29
Figure 15. Data from TA station F15A on January 15, 2009, after the mean and linear trends are removed and the data are bandpass filtered between 19.5 and 1000Hz.

Figure 16. Data from TA station F15A on January 15, 2009, after preprocessing and a 10% cosine taper is applied.

Figure 17. The causal part of the frequency spectrum for a preprocessed segment of the data.

Figure 18. Amplitude spectrum of the data after transformation to acceleration in the frequency domain.

Figure 19. Amplitude spectrum of a one hour segment of the fully preprocessed data with the instrument response removed.

Figure 20. The power spectrum of a one hour segment of data.

Figure 21. Smooth PSD estimates for TA station F15A on January 15, 2009.

Figure 22. Data from TA station F15A on January 15, 2009 after removal of the instrument response function and transformation back into the time domain.

Figure 23. Preprocessed data after temporal normalization and after bandpass application.

Figure 24. Amplitude spectrum of the data in the pass band of interest (5s to 100s) prior to and after spectral whitening.

Figure 25. All preprocessed data for TA station F15A in January 2009.

Figure 26. A frequency distribution plot (histogram) for TA station F15A in January 2009.

Figure 27. PSD and Probability Density Function for TA station F15A for the month of January 2009.
Figure 28. Probability Density Function for TA station A15 for the month of January 2009. ..........................................................52

Figure 29. A frequency distribution plot (histogram) for all stations in this study for January 2009. ..........................................................53

Figure 30. Probability Density Function for all stations in this study for January 2009. ..........................................................56

Figure 31. Estimated Green's functions plotted as a function of lag-time (s) and inter-station distance (km). ..........................................................58

Figure 32. Estimated Green's functions stacked for the month of January and plotted as a function of inter-station distance and lag time up to 5000 s. ........................................60

Figure 33. Cross-correlations for the vertical components of 26 transportable array stations in western Montana bandpassed between 1 and 10 seconds........................................61

Figure 34. Estimated greens function for cross-correlations of all station pairs plotted as a function of inter-station distance. Data bandpassed between 7 and 150 seconds. 73

Figure 35. Map of the 118 GSN stations use in the Berger study of ambient earth noise.74

Figure 36. PDF mode noise levels above the NLNM mapped across the US in 3 separate period bands (McNamara and Buland, 2003). ..................................................75

Figure 37. The global paths of selected body-wave arrivals and their predicted travel times (Wang 2014). ..........................................................76

Figure 38. Virtual seismograms of spatially stacked cross-correlations by 50 km stacking distance bins (Wang, 2014) ..........................................................76
List of Equations

Equation 1 ........................................................................................................6
Equation 2 ........................................................................................................6
Equation 3 ........................................................................................................6
Equation 4 ........................................................................................................6
Equation 5 ........................................................................................................7
Equation 6 ........................................................................................................8
Equation 7 .......................................................................................................19
Equation 8 .......................................................................................................28
Equation 9 .......................................................................................................30
Equation 10 ....................................................................................................30
Equation 11 ....................................................................................................31
Equation 12 ....................................................................................................35
Equation 13 ....................................................................................................36
Equation 14 ....................................................................................................41
Equation 15 ....................................................................................................48
Equation 16 ....................................................................................................48
Equation 17 ....................................................................................................71
Equation 18 ....................................................................................................71
Equation 19 ....................................................................................................72
1. Introduction

Seismic interferometry (SI) is a concept derived from Claerbout's conjecture, which was developed in 1968. This conjecture stated that the cross-correlation of noise traces recorded at two different receiver locations gives the response that would be observed at one of the receiver locations if there was a source at the other (Curtis, et al. 2006). This is known as a virtual source and can be achieved if the receivers are located in a three dimensional, heterogeneous medium, with a diffuse noise wavefield. This cross-correlated response is also referred to as Green’s function, and "diffuse” means that the amplitudes of the normal modes are uncorrelated but have equal expected energies in all directions (Wapenaar, et al., 2010). The Green's function is important because it contains information about how energy travels through the Earth between two locations. The conjecture was mathematically proven for acoustic mediums by Wapenaar (2003, 2004), Snieder (2004), and Van-Manen et al. (2006). Similarly mathematical derivations were completed for elastic medium and the concepts were demonstrated in laboratory experiments (Nicolson, 2012). Turning unwanted noise into signal is a valuable concept and has already been applied in different disciplines such as: ultrasonics, regional seismology, exploration seismology, underwater acoustics, medical imaging, etc. (Wapenaar & Thorbecke, 2013).

SI has many applications and potential where other seismic analysis/interpretation methods may not succeed. This method of analysis is completely data driven and the processing is generally straightforward. However the processing of long time-series data can be computationally intensive. The capability to retrieve information hidden in noise or complex scattering coda is likely the most important feature of the SI process. Ambient seismic noise can be used to obtain surface seismic profiles/ reflections in areas where active-source seismic data
acquisition may not be feasible (i.e. rugged terrain, harsh weather conditions). Some other areas where this technique could be useful are environmentally sensitive areas; areas under permanent monitoring for reservoir production; and in CO₂ sequestration surveillance (Draganov, et al., 2009).

The estimated Green’s functions obtained using SI contain information about the subsurface geological structures, which is based on changes in the relative speeds of body and surface waves travelling through the Earth. Seismic interferometry involves the cross-correlation of multiple scattered waves, recorded at different receiver locations, which converge to produce estimated Green’s functions. Since only seismic noise is involved in the process, the requirement for an active source is eliminated (Snieder, 2004). The generated noise cross-correlations from passive SI can be used to determine low-frequency characteristics of body waves. These low-frequency characteristics can be used in conjunction with active seismic data which usually lack low-frequency content. By combining results from both active and passive seismic data, a broadband reflection response can be produced (Draganov, et al., 2014).

The main function of any seismic network, like the transportable array implemented by IRIS (Incorporated Research Institutions for Seismology), is to provide high quality data for earthquake monitoring, investigations of possible noise sources, and research into the interior structure of the earth. A better understanding of seismic noise provides the ability to characterize the frequency response of the data which may contain valuable information for use in further analysis. Seismic noise analysis also provides the basis for reducing random noise in the data. Another advantage of this analysis would be identification of problems with seismic station equipment, and determining requirements for sensors and data acquisition systems for future use. Thus, spectral analysis of the ambient noise data is an integral part in selecting the proper
frequency bands which can be processed for cross-correlation, and therefore generation of estimated Green's function in appropriate frequency bands.

In order to present a thorough understanding of seismic interferometry concepts, I developed and modified existing Matlab tools to simulate Green's functions; provided a known subsurface structure, its relative p-wave velocities and source/receiver geometries. For more information regarding the Matlab codes used in this project please refer to Appendix D. Then, I conducted an investigation on the available ambient noise data from western Montana and analyzed the selected data in the frequency domain so that noise sources at their relative frequencies could be identified. Lastly, I determined if the ambient noise data currently available in western Montana and concepts of seismic interferometry could be used to effectively generate estimated Green's functions from the data. Limitations in time and computational power influence the amount of data which can be processed. Typically, years of ambient noise data are used to produce reliable cross-correlations or Green's functions. However, in this study, only one month of data was processed due to time constraint. Programming for all portions of this project was created in Matlab in a Windows operating system environment. My programming involves significant contributions from the seismic processing toolbox in Matlab and previous works credited in Appendix D of this thesis.
2. Background

Seismic interferometry can be divided into two separate categories: active source interferometry uses noise recordings from seismic surveys with active sources (e.g. explosives, vibrators, thump trucks, etc.), and passive source interferometry utilizes the naturally occurring events. These events include microseisms, thermal noise, storms, wind, tectonic/volcanic earthquakes and tremors, as well as cultural noise generated by industry machinery and traffic, etc. (Bormann, et al., 2009). In addition, passive source interferometry can be further categorized depending on the type of noise observed which can be either transient or ambient. Transient sources are seismic signals generated by earthquakes. Ambient sources are continuous noise due to wind, wave oscillation, cultural noise such as power lines, or traffic on a busy street.

In SI, either active or passive, cross-correlations of two receiver recordings can be interpreted as the response that would be measured at one of the receiver locations as if there were a source at the other (Wapenaar, et al., 2010). Active source interferometry involves cross-correlations and then summations of those cross-correlations over different source positions. In contrast, passive source interferometry turns passive seismic measurements (ambient noise) into deterministic seismic responses. This process works without explicit summation of correlations over various source positions because the correlated responses at the receivers are a superposition of simultaneously acting uncorrelated virtual sources (Wapenaar, et al., 2010). In passive source interferometry, it is common for years of data to be stacked together. This is done because when cross-correlations between specific pairs of receivers are stacked in time, over long periods (days to weeks to years), the estimated Green's functions (EGF) converge to the actual Earth response at the separation distance between the two stations being cross-correlated. Typically, the results for passive source interferometry are presented as a function of inter-station distance. Cross-
correlation or convolution of ambient noise records can both produce EGF between stations. However, cross-correlation is the most commonly used technique in practice and will be discussed in detail later on (Curtis, et al., 2006).

### 2.1. Seismic Interferometry: Power Conservation and Power Reciprocity Theory used in Demonstrating Concepts of SI

This subsection discusses Wapenaar's derivations of equations for seismic interferometry according to the principles of conservation of acoustic power and its direct relationship to Claerbout's conjecture (Wapenaar, 2002). Wapenaar’s 1D derivation consists of flux normalized up and down going wavefields as well as a horizontally layered medium. Figure 1 summarizes the behavior of (a) \( a_1 \) incident downward and (b) \( a_1 \) incident upward wavefield as well as their reflection and transmission responses. The boundary for these plane wave sources is the surface of the Earth. Also, the transmitted response in Figure 1(a) \( (T) \) is equal to the upward transmitted response in Figure 1 (b) when the source is below the deepest reflector.

![Figure 1](image_url)

**Figure 1.** a] An incident down-ward plane wave indicated by \( a_1 \) and its reflection/transmission (R/T) response in a horizontally layered medium. b] An impulsive up-ward plane wave source denoted by \( a_1 \), originating below the deepest layer in the model, and the resultant transmission response.
When the medium of interest is lossless and has only one incident downward wave \(a_1\) as shown in Figure 1 (a), the net downward power flux for the top \(\Phi_{dptop}\) and bottom \(\Phi_{dpbottom}\) layers in the model are equal (according to principles of power conservation). Equations 1-3 summarize these principles given the variables in Figure 1. The asterisks in the equations denote a complex conjugation of the preceding term. The downward plane wave in Figure 1 \((a_1)\) is now represented by the number 1. Equation 2 shows the signals which contribute to the net power flux in the top layer and equates these signals to those which generate the net power flux in the bottom layer. Equation 3 shows how Equation 2 can be modified to produce a general formula for obtaining reflection responses from the transmission responses obtained from the Earth and an upward plane wave.

**Equation 1**
\[
\Phi_{dptop} = \Phi_{dpbottom}
\]

**Equation 2**
\[
1 - R - R^* = T^* \cdot T
\]

**Equation 3**
\[
R + R^* = 1 - (T^* \cdot T)
\]

The above equations are valid in the frequency domain. However, Equation 3 can be expressed in the time domain using the Laplace transform as:

**Equation 4**
\[
R(t) + R(-t) = \delta(t) - T(-t) * T(t)
\]

Where \(*\) denotes convolution (not complex conjugate as in Equations 1-3), \(t\) is time, and \(\delta(t)\) is the source signal. This equation states that the reflection response can be obtained from the autocorrelation of the transmission response of an impulsive source deep in the subsurface. Autocorrelation is the process of convolving a function with its complex conjugate as shown in Equation 4. In fact, the autocorrelation does not change when the impulsive source is replaced by
any source of which the autocorrelation is once more an impulse. Then, the autocorrelation of the
transmission response of an ambient noise source in the subsurface can be used to retrieve the
reflection response of an impulsive source as if it were at the surface. One advantage of
autocorrelation of the transmission response is that it eliminates any time shift that may affect the
response. This implies that the depth of the ambient noise source is irrelevant as long as it is
located below the deepest boundary of the layered medium (Wapenaar K., 2003). When the net
upward and downward flux is conserved, and the medium is heterogeneous and without sources,
the Equation 4 can be modified to compensate for this complexity in the subsurface in a 3D
model. In this case, a one way reciprocity theorem of the correlation type is applied to take into
account the concepts of reflection and transmission applied to upward and downward
wavefields. Using source-receiver reciprocity, the net downward power flux as a function of
location and frequency for a 3D case is shown below.

Equation 5

\[ R(x_A, x_B, \omega) + R^*(x_B, x_A, \omega) = \delta(x_{HA} - x_{HB}) - \int_{\partial Dm} T^*(x, x_A, \omega)T(x, x_B, \omega)\,d^2x_H \]

The terms \( T^*(x, x_A, \omega)T(x, x_B, \omega) \), represents the cross-correlation of traces recorded at
the two receiver locations \( x_A \) and \( x_B \) on the surface with a source located at \( x \) on the boundary
\( \partial Dm \) located at the depth of the deepest reflector of interest. The term \( R(x_A, x_B, \omega) \), is the
receiver response which would be recorded at \( x_A \) if a source is located at \( x_B \). Again, \( \delta(x_{HA} - x_{HB}) \), represents the ambient noise source signal. The integral in Equation 5 cannot be solved
because ambient noise sources at different locations in the subsurface cannot be measured
sequentially. This problem is mitigated when the sources for different source positions on \( \partial Dm \)
are mutually uncorrelated and can be achieved by having transmission response data convolved
with ambient noise. Coincidently, this is the state ambient noise data are already in when
recorded at the surface. An inverse Fourier transform is applied, and the integral in Equation 5 is discretized and transformed into the time domain.

**Equation 6**

\[ R(x_A, x_B, t) + R(x_A, x_B, -t) = \delta(x_{H,A} - x_{H,B})\delta(t) \]

\[ -\sum_{i} \sum_{j} T(x_A, x_i, -t) \ast N_i(-t) \ast T(x_B, x_j, t) \ast N_j(t) \]

This results in the summation of all transmitted signals originating at the lowest reflector, which is consistent with stacking in the time domain. Typically, the terms included in the summation(s) can be interpreted as the transmission response observed at the surface (at \(x_A\) and \(x_B\)) from a distribution of uncorrelated noise sources defined by the vector \(x_i\) and /or \(x_j\) on the boundary \(\partial D_m\). Finally, the point source reflection response at receiver location \(x_A\) can be retrieved by solving for the causal part of the left hand side of Equation 6 (Wapenaar, 2003).

The effects of directionality of wavefields in the subsurface are strongly influenced by the receiver geometry used in the retrieval of Green's function. False reflection events can be observed if the directionality of a wavefield is ignored. If the acausal and causal parts of the spectrum are not identical, this implies that there is directionality in the wavefield. Therefore the subsurface sources that are contributing to the ambient noise record are not uniformly distributed. According to Sneider's (2004) analysis of the extraction of the Green's function from coda, seismic sources located in-line with the receiver pair \(x_A\) and \(x_B\) contribute the most in the interferometric Green's function. Therefore, a whole enclosing boundary of sources around the receiver pairs is not necessary in order to approximate Green's function, only the receivers found in the Fresnel zones (2D) / Fresnel Volume (3D) and in-line with the designated source/receiver pair contribute significantly (Wapenaar, et al., 2010). Sources not located in either of those two
positions interfere destructively and therefore do not provide any coherent contribution to the sum of the correlated transmission responses at $x_A$ and $x_B$.

### 2.2. Spectral Analysis

In order to characterize noise sources in ambient seismic noise, several different noise models have been developed over the last 20 years. The most well-known would be the New Low Noise Model (NLNM) and New High Noise Model (NHNM) presented by Peterson in 1993. The results from US station HLID which is an Advanced National Seismic System (ANSS) station (located approximately 10 km from Hailey Idaho) is presented below (McNamara & Buland, 2003). Although the algorithm for producing noise analyses of seismic data was originally developed by Peterson (1993), McNamara and Buland (2003) modified it to produce the PDF Analysis System (stand-alone software for use in the UNIX environment). This software was used to create the following image which includes the characterization of noise sources for this particular station.
Figure 2. Probability Density Function for an ANSS station 10 km outside Hailey Idaho. Also, shows the characterization of noise sources for the station (McNamara & Buland, 2003).

About 10 years later, Berger and Davis (2004) developed a new comprehensive model which depicts significant differences between the normal mode and body wave bands when compared to Peterson's original model of ambient earth noise (1993). The increased availability of data provided by IRIS and the Global Seismographic Network largely contributed to the differences between Peterson and the Berger models. Peterson's model is still widely used even though it is over 20 years old. The data which was analyzed came from a variety of stations throughout the globe during periods of apparent low seismic noise (Peterson, 1993). In contrast,
due to the increase in computational power over the ten year period between these comprehensive noise models, Berger developed a new model based on the entire global seismographic network over a period of one year between July 2001 and June 2002. The model developed by Berger contains enough information to include seasonal variations which allows better understanding of the global seismographic network. Also, there was no effort made to screen the data for earthquakes and select quiet periods like in the Peterson's model. All the ambient noise information is retained in the power spectral density (PSD) and probability density function (PDF) of Berger's model. Figure 3 shows the results found by Berger after computing the density of observations using only the vertical component of the data for all 118 stations used in that study. The color bar in the figure identifies the number of stations which fall into each 1 dB bin used in the model. The map of the stations used in Berger's study is included in Appendix C (Figure 35) of this paper.
2.3. Applications of Seismic Interferometry in Ambient Noise Data

SI is a processing technique which is used to achieve a variety of results including reconstructing the Earth’s reflection response, ambient noise tomography (ANT), and retrieval of surface waves. These results depend on the processing techniques and the type of data used. In this thesis, ambient noise data from western Montana was processed to generate the Earth’s reflection response after spectral analysis. Draganov used recorded acoustic transmission data and transformed it into simulated reflection data using the reciprocity equation shown in Equation 5 (Draganov, et al. 2006). This study revealed that it is normal for the reconstructed reflection response to have a noisier nature than a directly modeled reflection response; this phenomena is caused in part by the incoherent summation in the reconstruction process.
(Equation 6). Since the subsurface below the source locations is not homogeneous, it is normal for "ghost" error terms to be present in the reconstructed reflection response. However, when the subsurface sources are distributed randomly as described by Draganov in 2004, then these ghost events can be significantly weakened by the incoherent summation mentioned earlier (Draganov, 2004).

On a regional scale body waves are successfully extracted through spatial stacking of long term cross-correlations of ambient seismic noise (Wang, et.al, 2014). Wang showed that although body waves are much more difficult to extract than surface waves, stacking ambient noise in 50 km inter-station bins exhibits coherent core phases (ScS, PKIKPPKIKP, PcPPKPPKP) and crustal-mantle phases (Pn, P, PL, Sn, S, SPL, SnSn, SS, SSPL) for inter-station distances between 0 and 4000 km. Furthermore, Nishida was able to exhibit the global propagation of body waves through the cross-correlation of 658 stations over a period of 6 years (Nishida, 2013). Similarly, Boue was also able to produce a global representation of body wave propagation by cross-correlating the responses from 339 broadband stations distributed globally over a period of one year (Boue, et al., 2013). The resolution in the results from Nishida's investigation is much higher than those of Boue, this directly correlated to the shorter time-series used by Boue. Theoretically, the noise cross-correlations should converge to the complete Green's function as the square root of the time over which the cross-correlations are evaluated (Poli, et.al. 2012a). All of the above mentioned global investigations explored the interaction of long period or low frequency seismic waves ranging between 5 and 150 seconds. Typically seismic waves with periods between 20-100 seconds are referred to as seismic "hum" (Bensen, et al., 2007).
Other investigations were conducted to extract body waves focused on data with short periods between 1 and 10 seconds. This frequency range includes the secondary microseism peak which originates from oceanic wave oscillation. Some of these studies include the works of Poli (2012a) where the emergence of body waves from extended time-series data are observed and the Earth's impulse response effectively shows p-waves reflected from the Earth's mantle discontinuities at the transitions zones. The transition zones are located at approximately 410 km and 660 km beneath the surface and appear on synthetic seismogram stacks at approximately 100 seconds and 160 seconds (Poli, et.al., 2012a). Poli also conducted an experiment involving only 42 stations broadband stations located north of the Fennoscandian region. Here, they managed to effectively resolve Rayleigh waves and the PmP (p-wave reflected from the Mohorovičić discontinuity) from the vertical component of the broadband stations over the duration of one year (Poli, et.al., 2012b). On a smaller scale, Roux was able to verify that within the period band from 1 to 10 seconds the signal is dominated by Rayleigh waves which originate from microseism energy (Roux, et.al., 2005). Some of this energy is converted to p-waves because of heterogeneities in the Earth's upper crust. The number of stations in the Roux study was 30 and they were located in a 11km square which was located in Parkfield, CA. One month of continuous data was used to generate noise correlation functions which identified surface waves dominating between 0.1-0.5Hz and visible p-waves in the frequency band 0.7-1.3 Hz.

The two other common uses of direct wave interferometry are the retrieval of surface waves between seismometers and ambient noise tomography (ANT). Surface waves can be approximated as the solution of the 2D wave equation with a frequency dependent propagation velocity (Wapenaar, 2010), under the assumption that the sources are in-line with the two receivers and within the Fresnel zones for that receiver geometry. Therefore, Green's function for
the fundamental mode (main propagating mode of the surface waves) can be extracted by cross correlating the ambient noise recording at two seismometers and generating a map of Rayleigh wave responses.
3. Method

I used Matlab to develop tools which were used to apply the SI process on synthetic receiver responses for a given geometry and structural model. Then, the ambient noise data from various stations in western Montana was imported into Matlab for spectral analysis. Following the spectral analysis, recorded data for a given time period was preprocessed following a procedure similar to Bensen et. al., (2007). After preprocessing the ambient noise data from several stations, cross-correlations were performed to produce synthetic seismograms from ambient data.

3.1. Seismic Interferometry on a Given Geologic Model

I used Matlab to generate a geologic model which was then modified to fit various source/receiver geometries. The geologic model is designed with horizontal layers that have particular p-wave velocities associated with each layer. The SI process is applied to this model and generates EGF's from the receiver responses. To better display the targets which this thesis attempted to resolve using SI, I created a geologic model with the depth to each interface in the crust, and its associated p-wave velocity, defined by Zeiler’s model (2005) (Figure 4). The depth and p-wave velocities associated with the transition zones were obtained from Frost (2008) (Figure 5).
Figure 4. Shows the changes made to the velocity model for western Montana since 1984. Depth is on the y-axis in km and each layers p-wave velocities are shown inside each layer respectively (km/s). The model developed in 2003 is used in this paper (Zeiler, et.al., 2005).

Figure 5. The p-wave velocity (y-axis in km/s) is shown for models developed by Cammarano et. al., 2005, PREM, and the spherically symmetric Earth model AK135. Depth is shown on the x-axis in km (Frost, 2008).

I used the velocity models shown in Figures 4 and 5 to develop the depth and relative p-wave velocities for each layer in my geologic model used for simulation. The depths between each layer are: 0.0 km-7.0 km, 7.0 km-19.8 km, 19.8 km-39.7 km, 39.7 km-410.0 km, and 410.0
km - 660.0 km The p-wave velocities for each layer are: 5.70 km/s, 6.12 km/s, 6.53 km/s, 8 km/s, and 10 km/s.

The associated elastic properties including s-wave velocities were calculated using Castagna's rule and each layer's respective density using Gardner's Rule. Then, Poisson's ratio is calculated based on the p-wave and s-wave velocities. The geometry for the first model in this paper is constructed using 5 layers, 8 receivers (at the surface), and a single source located at the surface with 1 m horizontal displacement. After the geometry was defined, ray tracing was completed and applied by determining the reflection boundaries and then applying a shooting function which calculates the length, angle, and time traveled for each ray in each layer above the lowest layer defined in the geologic model. The resulting model for the given geometry and geology is shown below in Figure 6.
Figure 6. Geometry and ray tracing of a hypothetical geologic model showing layers of the Earth for western Montana up to the bottom of the transition zone. P-wave velocities are shown for each respective layer in the model and are associated with the color bar at the bottom of the figure. Here distance and depth are shown in meters and velocity is in m/s.

The downward and upward reflection coefficients (amplitudes of reflected and transmitted waves) are then calculated using the Zoeppritz approximation. This approximation is based on the principals of conservation of stress and displacement across layer boundaries (Sheriff & Geldart, 1995).

Equation 7

\[
\begin{bmatrix}
R_P(\theta_1) \\
R_S(\theta_1) \\
T_P(\theta_1) \\
T_S(\theta_1)
\end{bmatrix} =
\begin{bmatrix}
-\sin \theta_1 & -\cos \phi_1 & \sin \theta_2 & \cos \phi_2 \\
\cos \theta_1 & \sin \phi_1 & \cos \theta_2 & \sin \phi_2 \\
\sin 2\theta_1 & \frac{v_{p1}}{v_{s1}} \cos 2\phi_1 & \sin 2\theta_2 & \frac{v_{p2}}{v_{s2}} \cos 2\phi_2 \\
-\cos 2\theta_2 & \frac{v_{s2}}{v_{p2}} \cos 2\phi_2 & \sin 2\theta_1 & \frac{v_{s1}}{v_{p1}} \cos 2\phi_1
\end{bmatrix}^{-1}
\begin{bmatrix}
\sin \theta_1 \\
\cos \theta_1 \\
\sin 2\theta_1 \\
\cos 2\theta_1
\end{bmatrix}
\]

In my code only information pertaining to the p-waves is retained (i.e. \(R_p(\theta_1)\) and \(T_p(\theta_1)\)) because I only want to model body wave reflections. The next step is to perform amplitude
versus offset modeling. A Ricker wavelet is used as an impulse source with a frequency of 8 Hz, and a sampling frequency of 40 samples per second. Even though the source function used was a Ricker wavelet, it can be modified to adapt for any wavelet with various sampling rates and frequency. Then, reflectivity was taken to each reflecting interface using the reflection coefficient determined by the Zoeppritz equation (Equation 7). The next step was to convolve the matrix containing each reflecting interface, for each source/receiver pair, with the source function. Figure 7 shows the resulting receiver responses for the geometry shown in Figure 6.

![Receiver Responses](image)

**Figure 7.** Receiver responses generated through amplitude versus offset modeling. The colored lines match the layer interfaces shown in Figure 6. Two-way travel time is on the y-axis (seconds) and offset is presented on the x-axis (km).

Figure 7 shows where the p-wave reflections from each layer should be located in terms of time and offset. The model reveals that for the layers within the crust, significant move-out is visible as distance between the source and receiver is increased. However, the p-wave reflections
from the top and bottom of the transition zone show very little move out as source/receiver distance is increased.

Further modeling showed that when mutually uncorrelated sources were distributed below the deepest reflector of interest, the autocorrelation of the transmission response eliminated any time shift in the receiver response. In this scenario, all p-wave reflections appear horizontal and travel time is reduced to nearly half of the travel times presented on the y-axis of Figure 7. These observations are consistent with the relationships derived by Wapenaar (2003). Considering these observations, another model was created with mutually uncorrelated sources located near the surface. Although seismic interferometry theory states that the ambient noise sources should be located below the deepest reflector of interest, in practical scenarios most ambient noise sources originate at or near the surface (Zhan, et.al., 2010). Hence, the sources were randomly distributed in the crust and are shown in Figure 8.

Figure 8. Geometry and ray tracing of a hypothetical geologic model showing layers of the Earth for western Montana up to the bottom of the transition zone. P-wave velocities are shown for each respective layer in the model and are associated with the color bar at the bottom of the figure. Here distance and depth are shown in meters and velocity is in m/s.
The receiver response and resultant Green's functions are shown in Figure 9 which show similarities between the modeled receiver responses and the interferometric response. The Green's functions were calculated using the `xcorr` function in Matlab which calculates the cross-correlation between two sets of time series data. In this case, the time-series data are the receiver responses between all station pairs. The results from cross-correlations are then stacked based on inter-station distance. These concepts are discussed in greater detail in the *Generating Estimated Green's Functions* section in the method of this thesis. For a total of 8 receiver responses, 28 cross correlations are calculated and then stacked to produce 7 Green's functions at inter-station distances of about 50 km, 150 km, 225 km, 300 km, 375 km, 450 km, and 525 km.

![Figure 9](image)

*Figure 9. (a) The receiver response for the model shown in Figure 8. (b) The green's functions calculated through cross-correlation of receiver responses shown in (a). Time is shown on the y-axis (seconds) and offset is on the x-axis (km).*

To compare the modeled responses shown in Figure 9 (a) and the interferometric responses in Figure 9(b), the traces located at 300 km offset (or inter-station distance) were plotted on the same scale (Figure 10).
Figure 10. The interferometric trace at 300 km inter-station distance is shown in red and the directly modeled trace (at 300 km offset) is shown in blue.

Figure 10 shows good agreement between the direct and interferometric traces. The increased amplitude observed in the interferometric trace is caused by stacking the cross-correlation results based on inter-station distance. The change in amplitude can be eliminated by normalizing both the direct and interferometric traces with their own maximum. Then the scale between each trace would be the same, and the traces would show greater similarity. As expected, the interferometric traces are noisier than the directly modeled traces, which is caused by the cross-correlation of receiver responses and stacking.

3.2. Data Selection

Available ambient noise data are presented by IRIS through their Data Tool Matrix and subsequently their Seismiquery program which allows for a search of the DMC (Data Management Center) data holdings. Continuous ambient noise data are available for western Montana between the years of 2007 and 2010. In this time period, stations within Montana
appear to have the most widespread and continuous data in the year of 2009. The selected stations come from the Transportable Array (TA) network. The stations in this network have been installed across the United States between the years of 2002 and 2012. Figure 11 shows the available stations throughout the TA network, and their relative times of installation. On an average, each set of stations was active for about 2 years after installation.

Figure 11. Transportable Array (TA) network map including information on when each set of stations was installed. (Incorporated Research Institutions for Seismology, 2015)

Figure 12 shows selected stations used for the investigation of ambient seismic noise in western Montana. Some stations were not included in the station array for this project because they did not contain continuous data for the month of January in 2009. Data were extracted into Matlab via the IRISFetch software that is freely available through IRIS. This software allows for the direct retrieval of data from each of these stations, and then imports them into Matlab.
For each of the TA stations, data are available for three components at various sampling rates. The three components are radial, transverse, and vertical; where radial refers to a north/south orientation and transverse refers to a west/east orientation. Only the vertical component data at a sampling rate of 40 samples per second were used for this study. The type of instrument installed at each station is a Streckeisen STS-2 G3/Quanterra 330 Linear Phase Co. Data from each of these stations included the relevant poles and zero files such that the instrument response could be removed from the data.

3.3. Spectral analysis of Ambient Noise Data in western Montana

After the data were imported into Matlab, several modifications were made so that the data could be used for analysis. To understand the factors that contributed to the ambient noise observed at a particular station, I conducted a spectral analysis. This spectral analysis involves the generation of power spectral density (PSD) plots, and their modification to produce
probability density functions (PDF) for a particular station over a period of one month. The procedure used for this spectral analysis is similar to what the U.S. Geological Survey published as a stand-alone software package (McNamara & Boaz, 2006) for seismic noise analysis and follows the algorithm used by the Albuquerque Seismological Laboratory (ASL) in their definition of the NHNM and NLNM (Peterson, 1993).

The goal for spectral analysis in this study was to develop the tools for a standard method of analyzing ambient seismic background noise in Matlab. The tools used to generate results are comparable to the NHNM and NLNM, presented in the background section of this thesis.

3.3.1. Preprocessing

The first step in spectral analysis was to import the raw data. This was done for every station in Figure 10 for the entire month of January, 2009. In order to describe the steps involved in the processing of data, TA station F15A on the 15th of January, 2009, is used as an example. This station is located just south of Butte, Montana. On this day two earthquakes were recorded at every station used for this project. These earthquakes appear at about 8am and just after 6pm, and correspond to a magnitude 6.7 earthquake, which struck near the Loyalty Islands in the southwest Pacific basin, and the second was a magnitude 7.4 earthquake located near the Kuril Islands off the eastern coast of Russia (United States Geological Survey, 2015). Figure 13 below shows the extracted raw data from the IRIS DMC for station F15A.
The next step in preprocessing the day long time-series data was to parse the data into 1 hour long segments which overlap by 50%. Because the sampling rate for each station is 40 samples per second, one hour of broadband data contains 144 000 samples. Overlapping the time series decreases the variance observed in the final PSD estimate of the data. For an entire day of data, a total of 47 hour long segments are produced using this technique. Because the Fast Fourier Transforms (FFT) were used in processing, the number of samples in an hour long time-series is truncated to the next lowest power of two. This is determined by using the following formula where $N$ is the number of samples in the data and $p$ is a positive integer.
Equation 8

\[ 2^p \geq |N| \]

next lowest power of 2 for \( N \) = \( 2^{p-1} \)

Here the next lowest power of two is equal to \( 2^{17} \) which means that the new length of the data series for an hour long segment is 131,072 samples. This inherently changes the length of the time-series to 3276.8 seconds from 3600 seconds. In order to further reduce the variance, the hour long segments are divided again by one quarter in length and overlap by 75%. This means that each hour long time-series has 13 segments, 819.2 seconds in length, with a total of 32,768 samples per segment. The longest period which can be resolved by this method is approximately equal to the length of the time-series at this point divided by 10 which equals about 90 seconds. The shortest period which can be resolves is equal to the inverse of the Nyquist folding frequency (half of the sampling frequency) which is equal to 1/20 or 0.05 seconds. Readings at periods shorter than 0.05 seconds and longer than 90 seconds are not reliable due to resolution limitations. An example of the first segment of the time-series is presented in Figure 14.
Figure 14. First segment of raw data from TA station F15A on January 15, 2009. The total length of each time-series segment is 819.2 seconds and the y-axis represents the amplitude of the data in digital counts (unscaled displacement).

The next step in processing was to remove the mean from each segment of data making it zero mean data. In addition, the data have linear trends removed (using the "detrend" function in Matlab) and are bandpassed using a Butterworth filter between frequencies of 0.001 Hz and 19.5 Hz. The Streckeisen STS-2 has a lower frequency limit of about 120 seconds or about 0.008Hz for which the instrument has a flat response to velocity (at longer periods the sensitivity of the instrument is no longer constant and needs to be considered during removal of the instrument response) (Streckeisen, 1995). Bandpassing the data ensured that any long period trends associated with instrument glitches, were removed. The inclusion of long period trends can create large distortions in the frequency spectrums of the data by nullifying the estimation of low frequency spectral quantities. The bandpass filter applied in the Peterson (1993) and Berger &
Davis (2004) models was between 0.01 Hz and 20 Hz. This implies that some additional low frequency information, specifically between 100 and 120 second periods, from ambient seismic noise in western Montana was retained in my spectral analysis. The Butterworth filter was chosen to remove the long period trends because it can be used to generate transfer function coefficients. Those coefficients were then inserted into the "filt" function in Matlab, to return the discrete time transfer function, which was applied directly to the time-series data. Details regarding the use of the "detrend", "butter", and "filt" functions can be found on the Mathworks website (MathWorks, 2015). Relevant formulas involved with the preprocessing up to this point include:

**Equation 9**

\[
\bar{u}_{\text{mean}} = \frac{1}{N} \sum_{n=1}^{N} u_n,
\]

Each sample in the time series (\(u_n\) for \(n = 1, 2, 3, \ldots, N\)) is added together and then this summation is divided by the total number of samples in the time series, \(N\). This produces the mean value for the time series, \(\bar{u}_{\text{mean}}\). Next, Equation 10 shows the mean is removed from each sample in the time series \(u(t)\), producing zero-mean data \(x(t)\).

**Equation 10**

\[
x(t) = u(t) - \bar{u}_{\text{mean}}
\]

Figure 15 below shows the first segment of data after the mean and linear trends were removed, and the data was bandpassed between 0.001 Hz and 19.5 Hz.
Figure 15. Data from TA station F15A on January 15, 2009, after the mean and linear trends are removed and the data are bandpass filtered between 19.5 and 1000Hz. The length of the time-series is 819.2 seconds and the amplitude of the data are presented on the y-axis in digital counts which is an unscaled measure of displacement.

Lastly, a 10% cosine taper was applied using a Tukey window in order to reduce the spectral leakage caused by the next step when a FFT is applied. The windows for tapering were calculated using the following formula:

Equation 11

$$w(x) = \begin{cases} \frac{1}{2}\left\{1 + \cos\left(\frac{2\pi}{r}\left[x - \frac{r}{2}\right]\right)\right\}, & 0 \leq x < \frac{r}{2} \\ 1, & \frac{r}{2} \leq x < 1 - \frac{r}{2} \\ \frac{1}{2}\left\{1 + \cos\left(\frac{2\pi}{r}\left[x - 1 + \frac{r}{2}\right]\right)\right\}, & 1 - \frac{r}{2} \leq x \leq 1 \end{cases}$$

Because I chose to use a 10% cosine taper on the data, the variable $r$ in Equation 11 must be equal to the decimal value of this percentage (i.e. 0.1). Variable $x$ in Equation 11 represents an
evenly spaced vector ranging between 0 and 1 with $N$ entries, and was used to properly allocate the tapering coefficients to each sample in the original time series. The taper coefficients are stored in the variable $w(x)$, which is used to normalize each entry in the time series $x(t)$ through multiplication. The tapered data are presented in Figure 16, where ends of the data converge to zero.

![Data after 10% Cosine Taper](image)

**Figure 16.** Data from TA station F15A on January 15, 2009, after preprocessing and a 10% cosine taper is applied. The length of the time-series is 819.2 seconds and the amplitudes of the data are presented on the y-axis in digital counts which is an unscaled measure of displacement.

Tapering the time-series has the effect of smoothing the FFT, and decreasing the effects of the discontinuity between the beginning and end of the time-series (McNamara & Boaz, 2006). The reduction in variance is quantified as the ratio of the untapered waveform to the tapered waveform, and is typically $>1$. This ratio was later used to correct the absolute power in the final spectrum for each segment.
3.3.2. Power Spectral Density

To compute the power spectral density, each segment of data are preprocessed to the point shown in Figure 14 then the FFT is computed and normalized by the inverse of the sampling frequency \( d_t = 0.025s \). Here McNamara (2006) applies a normalization factor of \( 2 \times d_t/N \) to the square of the amplitude spectrum; in this thesis the normalization factor is applied prior to transformation to power units. Additional formulas and information regarding the above described procedure are available in Appendix A. Figure 17 corresponds to the raw frequency spectrum generated from the preprocessed data in Figure 16.

![Amplitude Spectrum of a Segment of Data in Digital Counts](image)

Figure 17. The causal part of the frequency spectrum for a preprocessed segment of the data. The y axis is presented in digital counts per Hz, and the x axis shows frequency up to Nyquist.

The above figure shows that the frequency spectrum of the preprocessed data has majority of the signal occurring at less than 1 Hz. The high amplitude of the spectrum at less
than 1 Hz makes it difficult to resolve higher frequency amplitudes because they are present at much smaller amplitudes. The data are transformed from units of displacement to velocity, and velocity to acceleration via integration (multiplication in the frequency domain). Please refer to Appendix A for formulas regarding integration in the frequency domain. The data are presented in units of acceleration because the microseism peaks are most flat in this domain making other features in the frequency spectrum more visible. In addition, transformation to acceleration amplifies the high frequency noise and is presented in Figure 18. The amplitude will not be in true units of acceleration until the instrument response is removed.

Figure 18. Amplitude spectrum of the data after transformation to acceleration in the frequency domain. The y axis shows the amplitude in digital counts transformed into acceleration as a function of frequency.
The above processes were repeated for each of the 13 segments composing each hour of data. Then, the amplitude spectrums for each of the segments, $P_{k,q}$, were averaged using the following equation:

**Equation 12**

$$A_k = \frac{1}{q} \left( P_{k,1} + P_{k,2} + \cdots + P_{k,q} \right),$$

where $k$, increases from 1 to the total number of samples in each segment minus 1 and $A_k$ is the raw estimate of the amplitude spectrum at the frequency $f_k$ for the segment of interest. The total number of segments in each hour long time-series is $q=13$. The segment averaging implicated above causes the estimate to have a 26 degrees of freedom producing a 95% level of confidence that each spectral point will fall within the range of -2.14dB and +2.87dB of the estimate calculated in the PSD (Peterson, 1993). At this stage the data were corrected for the effects of the cosine taper by multiplying the ratio described in the last part of the preprocessing section. Next, the instrument response was removed in the frequency domain using a subroutine transfer program. The instrument transfer function is specific to each station and depends on the pole and zero arrays as well as the constant provided by the RESP (SEED instrument response) file for each dataset. The instrument transfer function was deconvolved from the data by dividing the data by the transfer function in the frequency domain. This puts the data into true acceleration as a function of frequency. The causal part of the data in the frequency domain with instrument response removed are presented in the Figure 19.
Figure 19. Amplitude spectrum of a one hour segment of the fully preprocessed data with the instrument response removed. The y axis presents units of acceleration as a function of frequency and the x axis shows the frequency (Hz) up to Nyquist.

In the above calculation both the real and complex portions of the data are preserved through the use of complex division. Next, the power spectrum was calculated using Equation 13; here the normalization factor of $2*\Delta t/N$ was applied to the square of the fully preprocessed amplitude spectrum $D$.

Equation 13

$$Power = 10 \ast \log_{10} \left( (\text{real}(D) \ast \text{real}(D)) + (\text{imaginary}(D) \ast \text{imaginary}(D)) \right) \ast \left(\frac{2 \ast \Delta t}{N}\right)$$
The resulting power spectrum for an hour long segment is shown in Figure 20. In order to resolve all frequencies of the spectrum, the x axis is on a logarithmic scale. Also, this axis was transformed from frequency to period for easier comparison to the Peterson ambient seismic noise model. The y axis represents the power at each period in the spectrum in decibels (dB). The sudden drop observed in the data at about 0.05 seconds (outlined in black) is caused by the bandpass filter applied earlier.

![Power Spectrum of 1 Hour of Data](image)

Figure 20. The power spectrum of a one hour segment of data. The x axis is on a logarithmic scale in seconds and the y axis is power in dB. The black box shows a sudden drop in power caused by the preceding application of a bandpass filter.

In order to accurately sample the PSD, full octave averages were taken in 1/8th octave intervals. This means that values of the PSD are averaged between short periods and long periods. The first short period is equal to $T_s = d_t \times 2$ and the first long period is equal to $T_l = 2 \times T_s$. The relationship between $T_l$ and $T_s$ remains constant until the maximum frequency of about 200
seconds was reached. The values of the short period were incremented by multiplying the previous short period value by a factor of $2^{1/8}$. The central period of each octave $T_c = \sqrt{T_s \cdot T_b}$ was stored and used as the relative x values for the PDF, which was segmented into 97 different periods (or frequencies) due to the octave averaging. This reduces the length of the data from $N=16\,368$ to 97. Each hour of data was processed in this way and Figure 21 shows the smooth PSD estimates for the data collected from TA station F15A on January 15, 2009.

![PSD diagram Containing 1 Day of Data Composed of 1 Hour Segments with 50% Overlap](image)

Figure 21. Smooth PSD estimates for TA station F15A on January 15, 2009. The x axis represents the relative period in seconds on a logarithmic scale and the y axis shows the power in dB. Each line in the figure represents a different hour long segment of the data.

The PSD estimates for the whole month were then rounded to whole numbers and then separated into 1 dB bins ranging from -210 dB to an upper limit of -100 dB for TA station F15A.
The number of values which fall into each bin at each central period were recorded and then frequency distribution plots (histograms) were generated.

### 3.3.3. Probability Density Function

Finally, the distribution of powers were plotted as a function of period using the probability density function. The PDF at a particular central frequency is equal to the ratio of the number of spectral estimates that fall into a 1 dB bin at that central frequency, to the total number of spectral estimates of all powers with the same central frequency (i.e. the total number of hours in the dataset). The frequency distribution plots and final PDF for TA station F15A and all stations in the study (Figure 12) are shown in Figures 26-30 and are interpreted in the results section of this paper.

### 3.4. Preprocessing of Ambient Noise Data

I developed Matlab programming which follows a very similar procedure for preprocessing of ambient noise data to that of Bensen, et al., (2007). This procedure is also followed by IRIS for the generation of their cross-correlation data product. The procedure includes the removal of the data mean and linear trends, as well as the transformation in the frequency domain from digital counts at each station, to velocity. Data in the frequency domain were not halved and the entire frequency spectrum (both acausal and causal) are included in preprocessing. The final task in the first phase of preprocessing was the removal of the instrument response which places the data into true ground motion velocity units of m/s as a function of frequency. The first phase of preprocessing involved the same steps and formulas as outlined above in the preprocessing and PSD sections of the spectral analysis. Key differences in these two procedures are that the data were not segmented into hour long sections, an entire day
of data were processed at once with no overlapping portions, and the data were not transformed into acceleration.

Unlike in the spectral analysis portion of this study, the data were next transformed back into the time domain using the inverse fast Fourier transform function. The resulting time-series shows the data as a function of time and true ground motion velocity in Figure 22.

![Data with instrument response removed](image)

**Figure 22.** Data from TA station F15A on January 15, 2009 after removal of the instrument response function and transformation back into the time domain. Plotted as velocity versus time.

The next phase of preprocessing is temporal normalization, which was used to identify and effectively remove contamination of earthquakes. The method for temporal normalization which is promoted by Bensen et al. (2007) is called running-absolute-mean-normalization. This method computes the running average of the absolute waveform bandpassed in the earthquake band (15s-50s). This was done because in certain cases the effects of earthquakes will barely be
visible above background noise in ambient seismic data without any bandpass. This is particularly true when teleseisms are low magnitude and at great distances from the station of interest. The running average of the waveform is calculated within a normalization time window of fixed length. For this project, the chosen normalization time window was equal to half of the maximum period of the pass band applied to the data. In an attempt to retrieve reliable reflections and surface/body wave phases, the data was processed in two distinct frequency bands. The high frequency band between 0.1 and 1 Hz which was investigated primarily on a local scale (Roux (2005), Poli (2012), and Zhan(2010)) and a low frequency band between about 0.01 and 0.2 Hz (Bensen (2007), Wang(2014), Nishida (2013), Poli (2012) and Lin (2008)). These two bands were investigated because of the short duration for which data are processed (January 2009) and the complex geologic structure in western Montana. High and low frequency bands were processed in an attempt to identify the magnitude of waveform attenuation at different frequencies for the station density and distribution in this study. Furthermore, earthquake signatures can be observed in two distinct frequency bands depending on their magnitude. Large magnitudes typically appear at periods between 15-50 seconds known as the "earthquake band", and low magnitude earthquakes can be observed at periods between 1 and 10 seconds. An investigation of two different frequency bands shows whether passive seismic interferometry could be used on a regional scale of western Montana to resolve body waves and/or surface waves at particular frequencies. The weight of the normalization factor, \( \hat{\omega}_n \), was determined using the Equation 14 and define using the preprocessed data bandpassed in the earthquake band, \( \hat{d}_j \).

Equation 14

\[
\hat{\omega}_n = \frac{1}{2N+1} \sum_{j=n-N}^{n+N} |\hat{d}_j|,
\]
the normalization window was $N=50$ seconds which means that 1728 different normalization factors were applied to one day of data such that the effect of regional/ global seismicity could be removed from the ambient noise data. The next step was to normalize the preprocessed data by the inverse of the normalization weight. Since two large scale earthquakes were recorded on January 15, 2009, data for that day were a good example to show the effects of running-absolute-mean normalization on the data. Figure 23 (a) and (b) show the preprocessed data after temporal normalization and the bandpassed data with earthquakes removed from background noise.
Figure 23. (a) Preprocessed data after temporal normalization using the running absolute mean method where the weights are defined on the bandpassed data in the earthquake band. (b) Normalized data from TA station F15A plotted as rescaled acceleration versus time bandpassed between 15 and 50 seconds.
Figure 23 shows that the two earthquakes were effectively removed from the ambient noise data. Next, spectral whitening was applied to the data. Ambient noise data is not flat in the frequency domain as shown in the spectral analysis part of this paper. Typically ambient noise data exhibits peaks near the primary and secondary microseisms caused by oceanic wave oscillation, and also increases at periods greater than 50 seconds due to a signal now referred to as "seismic hum" (Bensen, et al., 2007). An example of the amplitude spectrum after temporal normalization is shown in Figure 24. (a) and is followed by the same signal after spectral whitening in Figure 24 (b). The primary and secondary microseism peaks are clearly visible in Figure 24 (a) at approximately 0.06 HZ and 0.16 Hz. Spectral whitening was completed in the frequency domain using the multitaper Thompson algorithm which generates a smoothed version of the magnitude of the frequency spectrum. More information on the multitaper Thomson algorithm can be found on the Mathworks website under the function name `pmtm` (MathWorks, 2015). Next, both the original and smoothed amplitude spectrums were normalized by dividing by their respective maxima. Finally the original amplitude spectrum was divided by the smoothed spectrum to return spectrally white data in the frequency domain.
Figure 24. (a) Amplitude spectrum of the data in the pass band of interest (5s to 100s) prior to spectral whitening as a function of frequency. (b) Amplitude spectrum of the data after spectral whitening. The tapering at both ends is cause by the bandpass filter between 5 and 100 seconds.
Finally, the data were transformed back into the time domain using an inverse fast Fourier transform in preparation for cross-correlation. An example of an entire month of data after being processed through spectral whitening is shown in Figure 25.

![Preprocessed Seismic Data Prior to Cross-Correlation](image)

Figure 25. All preprocessed data for TA station F15A in January 2009. Plotted as rescaled velocity as a function of time.

The above figure shows that the preprocessed data for the entire month does not contain significant earthquake contamination because of temporal normalization. The large spikes appearing above background noise are most likely caused by one of two possible sources. First, absolute-running-mean normalization cannot surgically remove narrow data glitches, as it will inevitable down-weight a broad time interval around the glitch. Therefore, large spikes could be representative of data glitches (Bensen, et.al., 2007). Second, surface waves from very large magnitude earthquakes (≥9) can circle the Earth multiple times and can have periods of around
100 s (Stickney, 2015). Because of the long period of these surface waves, they would not be removed in absolute-running-mean normalization where the normalization weights are defined on data bandpassed in the earthquake band. Earthquake signals which pass through the temporal normalization process tend to appear on cross-correlations as high amplitude spurious precursor arrivals (Bensen, et.al.,2007). When I cross-correlated the data without clipping the large spikes visible in Figure 25, the results showed abundant precursor events in the first 40 seconds of lag-time as well as other horizontal reflection events which do not correspond to any commonly observed reflective phases (Appendix B, Figure 34).

To remove the large spikes shown in Figure 25, the standard deviation was calculated for each day long time-series. Then, any sample in the day long time-series which had amplitude greater than 3 times the standard deviation was clipped to equal 3 times the standard deviation. I assumed that majority of the signal for each day was composed of ambient seismic noise rather than transient seismic noise caused by earthquakes. In addition to data glitches and earthquake signals not removed by temporal normalization, transformations to and from the frequency domain, as well as cosine tapering can all reduce the signal to noise ratio of the data.

### 3.5. Generating Estimated Green's Functions

After each daily time-series of data were preprocessed for each station, the daily segments for each station pair were bandpassed in the frequency band of interest and cross-correlated in the frequency domain. Cross-correlation is a process which measures the similarity between two signals. Although cross-correlation and convolution are very similar processes, cross-correlation is more commonly used in SI because the results can be directly presented as a function of lag-time and no reversal of time series is necessary (as in convolution). Hence, cross-correlation cuts down the number of operations necessary in processing. However, this does not
significantly influence processing time but was convenient. Cross-correlation for two station pairs, \( x(n) \) and \( y(n) \), is computed for every value of \( n \) from 1 (the first sample in each time series) to \( N \) (the last sample in the time series). The result is a two-sided correlated noise function (\( EGF \)) with a total of \( 2N+1 \) entries. The central value is the DC component of the data and does not contribute to the actual cross-correlated signal.

**Equation 15**

\[
EGF = \sum_{n=1}^{n=N} x(n) * (n)
\]

Energy levels for daily time-series vary significantly both on a daily basis and between stations in this study. Normalized cross-correlation was necessary and is summarized in Equation 16 for two time series, \( x(n) \) and \( y(n) \).

**Equation 16**

\[
EGF_{norm} = \frac{\sum_{n=1}^{n=N} x(n) * (n)}{\sqrt{\sum_{n=1}^{n=N} x^2(n) * \sum_{n=1}^{n=N} y^2(n)}}
\]

the numerator is equal to the basic equation for cross correlation (Equation 15), and the denominator scales the result of the cross-correlation by a factor that is related to the energy of each signal in cross-correlation (\( x(n) \) and \( y(n) \)). Both real and imaginary parts of the signal were considered in all calculations. For the 26 stations used in the study, a total of \( n*(n-1)/2 \) (where \( n \) is the number of stations) cross-correlations were calculated for each day. Then the results of cross-correlations for each day were stacked together, resulting in over 10 000 cross-correlations that were used to represent the EGF for one month of data. The cross-correlations are presented as a function of lag-time and inter-stations distance.
4. Results

4.1. Spectral Analysis

Spectral analysis provides the ability to characterize sources of ambient noise. The PSD’s for all hour long segments with a 50% overlap for TA station F15A were computed for the month of January. A frequency distribution plot was created from this information and the results for four different period bands (0.1s, 1.6s, 12.8s and 102.5s) are presented below in Figure 26.

![Frequency Distribution Plots Using 1dB Bins, at 4 Different Period Bands](image)

Figure 26. A frequency distribution plot (histogram) for TA station F15A in January 2009. The red, yellow, green, and blue plots correspond to periods of 0.1, 1.6, 12.8, and 102.5 seconds respectively. The number of occurrences in each 1 dB plot for each of the 4 period bands listed above are plotted for bins between -210 and -100 dB.

The above plot shows that much of the data with periods between 0.1 and about 15 seconds exhibits amplitudes between -140 dB and -160 dB. Data with periods of about 100 seconds typically have lower power levels with amplitudes between -170 dB and -190 dB. Data
with long periods such as those around 100 seconds contribute to the signal known as "seismic hum", and has fairly low amplitude characteristics until the periods become longer than 100 seconds. At periods longer than 100 seconds, the strength of the seismic hum signal increases significantly but cannot be effectively resolved by the spectral analysis since the resolution is limited to about 90 seconds. Figure 27 shows the PSD and corresponding PDF for TA station F15A through the month of January, 2009.

![PSD and Probability Density Function For Station F15A from January 1-31, 2009](image)

**Figure 27.** PSD and Probability Density Function for TA station F15A for the month of January 2009. Period on the x-axis is plotted on a logarithmic scale. Power on the y axis is reported in decibels. The data is binned into 1 dB segments for periods ranging between 0.025s and 200 seconds. Plotted with the Peterson NHNM and NLNM.

The most common source of seismic noise is referred to as cultural noise and is cause by human actions at or near the surface (McNamara & Buland, 2003). This type of noise is typically
composed of high frequency (>1-10 Hz) surface waves, and attenuates within several kilometers of distance and depth. The large spread between transportable array stations means that this type of noise, and this frequency band, are not suitable for cross-correlations and estimation of Green's functions for this study. Cultural noise exhibits large diurnal changes and can strongly be influenced by the type of disturbance. During the day when vehicles are on the roads nearby a station, the station exhibits up to 30 dB increase in power. These diurnal variations can be seen in Figure 27 as a ~15 dB increase in power between 0.1 and 1 seconds. Station A15A exhibits very high power levels between 0.1 and 1 seconds because of the presence of pump jacks continuously operating in the area. Small pump jacks, visible along the I-15 South from the Alberta/Montana border, are likely producing from the relatively shallow Kevin or Sunburst fields near the towns of the same name (north of Shelby) (Paukert, 2015). Therefore, the power level for this station approaches the NHNM at higher frequencies (shorter periods) that are associated with this disturbance. Figure 28 shows the PDF for Station A15A, and points out the increase in cultural noise caused by the operating machinery.
Figure 28. Probability Density Function for TA station A15A for the month of January 2009. Period on the x-axis is plotted on a logarithmic scale. Power on the y-axis is reported in decibels. The data are binned into 1 dB segments for periods ranging between 0.025 s and 200 s. Plotted with the Peterson NHNM and NLMN. Also shown is the relative power increase caused by constantly operating machinery when compared to the PDF from Station F15A (Figure 25).

The frequency distribution plot and PDF model for all of the stations in this study are shown in Figures 29 and 30, respectively. The frequency distribution plot shows that the noise typically characterized as cultural noise has a bimodal distribution. This modality is caused by the influence of different types of cultural noise. Typically vehicles and wind turbulence between topographical irregularities, as well as the coupling of tree motion relative to tree roots generate high-frequency noise (McNamara & Boaz, 2006). Operating machinery caused high amplitude high-frequency noise, and thus created the second mode (~130 dB) for the high frequency noise at around 10 Hz (0.1 s).
Figure 29. A frequency distribution plot (histogram) for all stations in this study for January 2009. The red, yellow, green, and blue plots correspond to periods of 0.1, 1.6, 12.8, and 102.5 seconds respectively. The number of occurrences in each 1 dB plot for each of the 4 period bands listed above are plotted for bins between -210 and -80 dB.

Two dominant peaks are visible in the power spectral density plots for every station. These peaks are representative of the primary and secondary microseisms. The primary microseism found at a longer period is typically between 10 and 16 seconds, and is known as the single frequency peak. In this study, the single frequency peak was located at about 12 seconds or more towards the shorter period end of the expected values. This microseism is generated by vertical pressure variations or waves crashing on the shoreline. These events originate in shallow coastal waters where oceanic wave energy is transformed directly into seismic energy (Hasselmann, 1963). The secondary microseism shows a high amplitude and high frequency signature, and is known as the double frequency peak. This peak is generated by the
superposition of oceanic waves of equal period traveling in opposite directions. This motion generates standing gravity waves of half the period of standard water waves. Gravity waves are waves generated in a fluid medium or at the interface between two media when the force of gravity or buoyancy tries to restore equilibrium (Gill, 1982). These standing gravity waves cause perturbations in the water column which propagate to the ocean floor (McNamara & Buland, 2003). Seismic stations in western Montana show that the secondary microseism falls between about 5 and 6 second periods with power levels ranging from -135 dB to approximately -120 dB with a mean around -130 dB. The power of the secondary microseism exhibits significant diurnal and seasonal variations caused by the presence of oceanic storms. Large oceanic storms can cause up to 20 dB increases in power (McNamara & Boaz, 2006).

Considering the fact that the processed ambient noise data is originated in the winter, the results are consistent to that found by McNamara and Buland (2003) during their seasonal investigation of ambient noise. They mentioned that during the winter months, the power increase for the secondary microseism peak can increase between 15 to 20 dB, and shift to slightly longer periods. This is caused by the increased intensity of the Atlantic and Pacific storms during the fall and winter. Alternately, at long periods (50-100s), noise increases during the spring and summer months and decreases during the winter. This is consistent with the low power levels observed in this study at these periods. The increase in noise during the summer could be attributed to large variations in daily temperature. The power of the secondary microseism is also strongly influenced by proximity to the coastline. The power level for the secondary microseism in western Montana during the month of January is expected to be lower than the power of this microseism in Hawaii along the coastline. The smearing of the secondary microseism peak is
caused by both averaging techniques used in the spectral analysis and diurnal/seasonal variations.

Unlike the Peterson model, in this spectral analysis no effort was made to eliminate body and surface waves which originated from earthquakes. As a result, power levels were significantly increased throughout the spectrum when influenced by a teleseismic event. Large teleseismic events are usually dominated by surface waves at periods greater than ~15 seconds. Smaller (local) earthquakes, and the resulting p and s-waves, dominated shorter periods (at less than 1s). This energy can easily be misinterpreted as cultural noise. The effects of teleseismic events are clearly identified in Figure 30.
Figure 30. Probability Density Function for all stations in this study for January 2009. Period on the x-axis is plotted on a logarithmic scale. Power on the y axis is reported in decibels. The data are binned into 1 dB segments for periods ranging between 0.025s and 200s. Plotted with the Peterson NHNM and NLNM. Defines the relative sources associated with particular frequency and power levels in western Montana.

The data in this model was screened to ensure that all data used was recorded continuously through the month of January. Therefore, the data do not exhibit the effects of telemetry drop-outs which appear as high power linear events typically above the NHNM.

Since the Peterson models, advances in instrumentation have reduced noise levels in long-period bands (> 10 s), while changing local noise sources such as roads and population density can increase shorter period (< 1 s) noise levels. Ambient noise levels are also strongly affected by geographic location mainly due to proximity to coastlines and population centers (McNamara & Buland, 2003). Overall, western Montana exhibits low spectral power at all
frequencies when compared to the rest of the U.S.. The data from western Montana (Figure 28) is compared to McNamara and Buland's study (2003) involving the geographic variations in ambient seismic noise throughout the U.S.. This model falls nicely within all of their estimates for the Montana region at all frequency bands. Figures associated with the McNamara and Buland study (2003) including geographic variations are included in Appendix C of this thesis.

According to the probability density function (Figure 28), the highest probabilities fall between 1 and 100 second periods. This is the most stable part of the spectrum. Thus, the frequency bands between 1-10 seconds and 20 - 100 seconds were chosen to be the bands of interest for cross-correlation of ambient noise data for this study.

4.2. Green's Functions Estimation

After preprocessing day long segments of ambient noise data for each station shown in Figure 12, the data were cross-correlated in the frequency band of interest defined by spectral analysis and previous works (Wang (2014), Nishida (2013), Poli (2012), Roux (2005), and Zhan (2010)). The goal of this process was to reveal body waves through temporal stacking of month long cross-correlations of ambient noise from the transportable array seismographic network in western Montana. The resulting cross-correlations are then plotted as a function of inter-station distance and lag-time. Due to limitation of computational power and extended processing time, only one month of data were cross-correlated, and then were stacked to represent a month long time-series. This study is used as a test to verify if one month of data and standard preprocessing in Matlab can effectively resolve subsurface features (p-wave reflections) in western Montana. The previously mentioned subsurface features include the Mohorovičić discontinuity and other major components in the Earth's interior such as the transition zone between the upper and lower mantle as well as the core phases.
The results from the cross-correlation of 26 station pairs for the frequency band between 20 and 100 seconds are presented in Figure 31 for lag-times between 0 and 250 seconds.

![Graph showing cross-correlation results](image)

**Figure 31.** Estimated Green's functions plotted as a function of lag-time (s) and inter-station distance (km). Two coherent phases are observed, one highlighted in yellow corresponds to the direct p-wave reflected from the moho and the one highlighted in green shows arrivals with travel times very similar to those exhibited by Rayleigh waves.

The estimated Green's functions in Figure 31 show reflections with travel times which correspond well to those presented for the p-waves reflection from the moho (PmP phase) in the synthetic model (Figure 7). Also, the Rayleigh wave phase is visible at slower speeds than the p-waves (Figure 31, highlighted in Green). The Rayleigh waves show similar arrival times to those observed by Lin et.al., 2008, for the vertical component of data in the frequency band between 10 and 50 seconds. These reflections are not well resolved at inter-station distances that are greater than 450 km, mostly due to the lack of station pairs with distances greater than 450 km. Additional reflection events are visible between 90 and 175 seconds lag-time, although they are neither laterally continuous nor coherent at inter-station distances greater than 150 km. In
addition, these events appear to have the same move-out as the Rayleigh waves, and therefore could be multiples cause by scattering of the surface waves due to structural irregularities. Extraction of body waves from ambient seismic noise, without using transient sources (earthquakes) is very difficult, and the reflections were hard to identify because of large amounts of noise in the data. The high amount of noise in the cross-correlations could be caused by the fact that the noise distribution is not completely uniform in time and space. Temporal normalization was used to remove the effects of large seismic events. Otherwise, these events would dominate the arrival-time structure of the noise correlation functions. Since there is high level of noise in the results, refining the normalization procedure may increase the signal to noise ratio of the data (Sabra, et.al., 2005). The causal side of the cross-correlated data, up to a lag-time of 5000 seconds is presented in Figure 32. The increased lag-time reveals no coherent phases. If longer time series data was used then I would expect to resolve nearly horizontal reflection events with little to no move-out, at approximately 900 s, 2400 s and 3000 s corresponding to the ScS, PKIKP, and PcPK phases, respectively (Wang, et.al., 2014)
Figure 32. Estimated Green's functions stacked for the month of January and plotted as a function of inter-station distance and lag time up to 5000 s. The data were bandpassed between 20 s and 100 s. No coherent phases are visible.

When the data were spatially stacked into 5 km bins for inter-station distances ranging between 50 and 500 km, no coherent waveforms were observed. This is likely due to the fact that many of the cross-correlations have such low signal-to-noise ratios that they completely mask the signal that is present in the data prior to spatial stacking. Attempts were made to generate EGF's using data that was segmented into hour long sections. However, no reflection event was observed and the results were inconclusive. Thus, this type of temporal stacking is ineffective when trying to produce cross-correlation functions for one month of data. The preprocessing and cross-correlations were repeated for the frequency band between 1 and 10
second periods, and the cross-correlation is presented in Figure 33.

Figure 33. Cross-correlations for the vertical components of 26 transportable array stations in western Montana bandpassed between 1 and 10 seconds.

For the data bandpassed between 1 and 10 seconds, no coherent body waves are visible. Also, the Rayleigh waves propagating across the virtual seismogram are weakly resolved and not coherent at inter-station distances greater than 300 km. Coherent reflection events in this frequency band are likely masked by noise in the cross-correlations originating from earthquake contamination, which was not removed during temporal normalization defined on the earthquake band (15-50s). Many earthquakes both local and regional can produce signals with periods between 1 s-10 s which may have remained, if signals in the 15 s-50 s period band were not present at the same time. Poli et.al., (2012b) was able to resolve the SmS (surface wave reflected from the Mohorovičić discontinuity) phase from the acausal part of the cross-correlation of the vertical component of ambient noise in the frequency band between 1 and 2 seconds. The SmS
phase is not stable in tectonic regions, and therefore may be hard to resolve in western Montana (Zhan, et.al., 2010).
5. Conclusions

Programming in Matlab was an effective way to produce synthetic seismograms which summarize the theory behind the seismic interferometry process, and present accurate travel times for body wave reflections at horizontal reflective boundaries in the Earth. Modeling the direct and interferometric traces for specific receiver geometries and velocity models provides an estimation of noise correlation functions (EGF’s). Processing of ambient noise data and subsequent spectral analysis were useful in presenting the ambient noise wavefield, and understanding its power sources in western Montana. Spectral analysis was effective in identifying appropriate frequency bands which can be used in the seismic interferometry process. This analysis indicated that the highest values from the probability density function are located between 1 and 100 seconds, and that the ambient noise in this frequency band is composed of both P and S-waves. Spectral analysis indicates that ambient noise in western Montana has low power levels compared to some regions in the United States. Further, the primary and secondary ocean microseisms are found at approximately 12 seconds and 5 seconds (in western Montana during the winter).

The Matlab code developed in this study can be applied to ambient noise data from any seismographic network available through the IRIS Data management center. Transformation of data into this format provides the ability to use the data in conjunction with the TauP tool kit available in Matlab. It is a powerful earthquake data processing tool that can be used for comparison of earthquake generated responses through passive seismic interferometry.

Ambient seismic noise in the frequency bands between 1 and 10 seconds and 20 and 100 seconds should be used for seismic interferometry processes on regional scales. However, for the longer period band, the array used for this study was not large enough to identify all body wave
phases reflected from the earth's interior, which are more clear at inter-station distances greater than 500 km. The transportable array has potential to resolve crustal body wave reflections, but a much longer period of recorded data must be used in temporal stacking. Additional processing steps can be taken to reduce the signal-to-noise ratio of the data (please see the Future Work section for more details). The retrieval of reflections in the crust of western Montana using seismic interferometry was a challenge considering the complex geologic structure in the area. Successful extraction of body waves in the crust/mantle has been achieved in areas with simpler crustal structure (i.e. Northern Finland and the Canadian Shield). However, cross-correlations of data (bandpassed between 20 and 100s) show coherent reflections which are associated with P-waves reflected from the Mohorovičić discontinuity and Rayleigh waves.
6. Future Work

In order to effectively resolve reflection events in the crust for the state of Montana, I would recommend using a seismographic network with a much larger station density. Increasing station density would allow for the investigation of high frequency noise which attenuates rapidly and could not be investigated in my thesis due to the large inter-station distances between TA stations (~70 km). Further, improvements can be made on the cross-correlation results obtained from this study. Future work can include the removal of surface waves in the resulting cross-correlations as well as improvements to signal-to-noise ratio of the data.

First, longer time-series data in conjunction with higher computing power are needed to reveal coherent body waves in spatially stacked synthetic seismograms. Second, the coherence of the prestacked cross-correlations should be investigated, and signals with very low signal-to-noise ratios should be eliminated before spatial stacking. In order to better resolve body waves, the strong amplitudes in the virtual seismogram can be reduced to twice the peak of the signal window of the PKIKPPIKP phase (provided that it is coherent). Furthermore, the time window corresponding to the surface waves can be zeroed out, or wavenumber-frequency filtering can be used to minimize surface waves and improve the SNR of body waves (Nishida, 2013; Xu et al., 2012; Draganov et al. 2009). The use of the transportable array seismographic network in identifying body waves within the earth is feasible. However, larger inter-station distances should be included (>1000 km) such that critical distances and wavefield directionality can be considered.

An investigation of the radial and transverse (horizontal) components of ambient noise can be effective in identifying p-wave and s-wave phases. Then, these phases can be subject to polarization analysis (frequency-time analysis) which would compare the particle motion of
different observed phases to that of a synthetic seismogram created at similar inter-station distance. This type of analysis was beyond the scope of this study but would be an effective way to confirm the observed phases in cross-correlations.
7. References


The method used to compute the PSD is commonly referred to as the direct Fourier transform or the Cooley-Tukey method (Cooley & Tukey, 1965); the PSD was calculated using a finite range FFT summarized by the following equation for the periodic time-series of each segment \( y(t) \).

**Equation 17**

\[
Y(f,T) = \int_{0}^{T_r} y(t)e^{-i2\pi ft} dt
\]

Here, \( Y(f,T) \) is the continuous time-series, \( t \) is the continuous flow of time from zero to \( T_r \) which is 819.2 seconds and \( f \) are the frequencies within the data. However, because our data is discretized in the frequency domain with \( f = f_k = k/N \Delta t \) for \( k=1,2,3...,N-1 \), the Fourier components are derived as follows:

**Equation 18**

\[
Y_k = \frac{Y(f_k,T)}{\Delta t}
\]

Here \( \Delta t \) is equal to the inverse of the sampling frequency. The two above equations are simultaneously solved when the "fft" function is used in MATLAB. The next step according to McNamara and Boaz (2006) was to apply a normalization factor of \( 2\Delta t/N \) to the square of the amplitude spectrum retrieved from the FFT. In this study the normalization factor was applied at a later time such that it does not conflict with the removal of the instrument response from the data. At this point, half of the amplitude spectrum was removed such that only the causal part of the spectrum remains. In this process the number of samples was reduced by half leaving \( N=16385 \). The next step in PSD processing was the transformation from distance into velocity and velocity into acceleration so that the instrument response could be removed. Then the data
was transformed into actual units of acceleration (m/s²). This transformation needs to be completed in the frequency domain, hence it was not applied to the data prior to the application of the FFT. This transformation was computed by integrating the causal part of the data which in the frequency domain results in the following equations:

**Equation 19**
*For each sample in the amplitude spectrum:*

\[
\Omega(n) = 2 \pi \left( \frac{\text{sampling frequency}}{N} \right) \frac{2}{n} \]  
\text{for } n = 1,2,3,\ldots,N

\[
vel(n) = \text{real}(Y(n)) \cdot \Omega(n) \cdot i - \text{imaginary}(Y(n)) \cdot \Omega(n)
\]

The above equations must first be applied to the causal frequency spectrum to generate the frequency spectrum in units of velocity/Hz and then again to generate the frequency spectrum in terms of acceleration/Hz.
9. Appendix B: Preliminary Cross Correlation Result

Figure 34. Estimated greens function for cross-correlations of all station pairs plotted as a function of inter-station distance. Data bandpassed between 7 and 150 seconds plotted to a lag-time of 200 seconds. The red box shows a reflection event which could be representative of the top of the transition zone between the upper and lower mantle located at 410 km in depth. Green box shows high amplitude precursory arrivals caused by earthquake signals not removed during temporal normalization.
10. Appendix C: Support Figures

Figure 35. Map of the 118 GSN stations use in the Berger study of ambient earth noise.
Figure 36. PDF mode noise levels above the NLNM mapped across the US in 3 separate period bands (McNamara and Buland, 2003).
Figure 37. (a) The global paths of selected body-wave arrivals (1. direct [P/S], blue line; 2. twice reflected [PP/SS], blue line; 3. outer-core reflected [PcP/Scs], red line; 4. twice inner core passing [PKIKPPKIKP], green line; 5. outer core and twice surface reflected [PcPPKPKP], purple line); (b) predicted travel time curves based on the AK135 spherically symmetric Earth Model, color of each line is the same as that of the corresponding phase defined in (a) (Wang, 2014).

Figure 38. Virtual seismograms of spatially stacked cross-correlations by 50 km stacking distance bins. Each trace is made by stacking all cross-correlations in the same distance bin and normalizing by stacking fold. For a better view of the body waves, strong amplitudes are cut to twice the peak of the signal window of PKIKPPKIKP. The plotted phases and colors correspond to those defined in Figure 36 (Wang, 2014).
11. **Appendix D: Matlab Code**

Table 1 summarizes the Matlab functions which originated from external sources and were implemented in some form to this thesis. Many of these codes have been modified to achieve the desired results.

**Table 1. Matlab functions and sources implemented into the Matlab code used in this thesis. Many of these functions were modified to achieve results specific to this project.**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>model.m</td>
<td>(Nainggolan, 2011)</td>
</tr>
<tr>
<td>logsyn.m</td>
<td></td>
</tr>
<tr>
<td>shooting_one_source.m</td>
<td></td>
</tr>
<tr>
<td>shuey.m</td>
<td></td>
</tr>
<tr>
<td>thomsen.m</td>
<td></td>
</tr>
<tr>
<td>zoepprtiz.m</td>
<td></td>
</tr>
<tr>
<td>bp_bu_co.m</td>
<td>(Porritt, 2013)</td>
</tr>
<tr>
<td>complex_amplitude.m</td>
<td></td>
</tr>
<tr>
<td>cos_taper.m</td>
<td></td>
</tr>
<tr>
<td>cross_correlate.m</td>
<td></td>
</tr>
<tr>
<td>freq_differentiate.m</td>
<td></td>
</tr>
<tr>
<td>freq_integrate.m</td>
<td></td>
</tr>
<tr>
<td>generate_response.m</td>
<td></td>
</tr>
<tr>
<td>interpbl.m</td>
<td></td>
</tr>
<tr>
<td>sinci.m</td>
<td></td>
</tr>
<tr>
<td>wig.m</td>
<td></td>
</tr>
<tr>
<td>zdiv.m</td>
<td></td>
</tr>
<tr>
<td>between.m</td>
<td>(CREWES Project, 2015)</td>
</tr>
<tr>
<td>fffrl.m</td>
<td></td>
</tr>
<tr>
<td>filt.m</td>
<td></td>
</tr>
<tr>
<td>findex.m</td>
<td></td>
</tr>
<tr>
<td>hmask.m</td>
<td></td>
</tr>
<tr>
<td>ifffrl.m</td>
<td></td>
</tr>
<tr>
<td>interpextrap.m</td>
<td></td>
</tr>
<tr>
<td>isearthobj.m</td>
<td></td>
</tr>
<tr>
<td>lint.m</td>
<td></td>
</tr>
<tr>
<td>logtotime.m</td>
<td></td>
</tr>
<tr>
<td>near.m</td>
<td></td>
</tr>
<tr>
<td>objget.m</td>
<td></td>
</tr>
<tr>
<td>objset.m</td>
<td></td>
</tr>
<tr>
<td>ocget.m</td>
<td></td>
</tr>
<tr>
<td>oget.m</td>
<td></td>
</tr>
<tr>
<td>padpow2.m</td>
<td></td>
</tr>
<tr>
<td>randobj.m</td>
<td>resamp.m</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>randobj.m</td>
<td>resamp.m</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>seis2z.m</td>
<td></td>
</tr>
<tr>
<td>whitening.m</td>
<td></td>
</tr>
</tbody>
</table>

(Mahmoudian, 2006)

(Daphne, 2010)
11.1. Concepts of Seismic Interferometry Used on Synthetic Data

% model.m
%============================================================================
% MAIN PROGRAM
%============================================================================
clc; clear all; close all;
format short g
%============================================================================
% Define Geometry
%============================================================================
fprintf('--- Defining the interfaces ...
');
% Input geometry
xmin = 0; xmax = 600000;
zmin = 0; zmax = 5155000;
% Make strata layer
zlayer = [0;40000;410000;660000;2885000;5155000];
%zlayer = [0;2885000;5155000];%for 1 layer, make sure you correct the vp values to %one layer in the "create elastic parameters" section.
nlayer = length(zlayer);
layer = 1:1:nlayer;
thick = abs(diff(zlayer));
x = [xmin xmax]; z = [zlayer zlayer];
dg = 10;
xx = xmin:dg:xmax; nx = length(xx);
zz = repmat(zlayer,1,nx);
fprintf(' Geometry has been defined...[OK]
');
%============================================================================
% Source-Receiver Groups
%============================================================================
fprintf('--- Setting source-receiver configuration ...
');
% Receiver Interval
dr = 75000;
%100 Sources and 100 Receivers-----------------------------
%Source
xs=[0:20000:600000];
%xs=40000;
%xs=[550];% for one source, this is a good way to test if the cross-
correlation will return the same result as the directly modeled response.
ns=length(xs);
random_z= rand ([1 ns]);% randomizes the depth assigned to the 'ns' % sources.
% This next part of the code places all of the sources anywhere between % 2000 m and 1250 m (note the 750 m difference) in depth.
zs= (random_z*750)+1250;
zs= (random_z*410000);% use this line for sources distributed randomly %throughout the subsurface not just in the bottom layer.
%NOTE IF THE SOURCES ARE LOCATED DIRECTLY BENEATH THE RECIEVER THE CODE %WILL NOT WORK.
% RECEIVERS
xr=[1:dr:1:600000];
% 100 receivers with a 20m receiver interval spread between 1 and 2000 m
% spread
nr = length(xr);
zr = zeros(1,nr);
nray = ns*nr;
fprintf(' Source-Receiver Groups have been setted...[OK]\n');

%============================================================================
================
% Create Elastic Parameter
%============================================================================
================
fprintf('--- > Creating elastic parameter ...
');
% Create synthetic Vp, Vs, and Density
vlayer = [6000;7500;9000;10500;11500;8000]; % Velocity
% vlayer = [6000;11500;8000];%for 1 layer
vlayer = vlayer(1:length(zlayer));
vel = [vlayer vlayer];
vp = vlayer;
% P wave velocity
vs=1.09913326*(vp.^0.9238115336);%S wave Velocity calculated based on
Carroll's rule
vs = (vp-1360)/1.16; % S wave velocity based on Castagna's rule
ro = 0.31.*(vp).^0.25; % Density based on Gardner's rule
pois = (vs.^2-0.5*vp.^2)./(vs.^2-vp.^2); % Poisson ratio
fprintf(' No.Layer Depth(m) Vp(m/s) Vs(m/s) Density(kg/m^3) Poisson Ratio
');
disp([layer' zlayer vp vs ro pois]);
fprintf('Elastic parameters have been created...[OK]\n');

figure;
set(gcf,'color','white');
% Plot P wave velocity
subplot(1,3,1);
stairs(vp,zlayer,'LineWidth',2.5,'Color',[0.07843 0.1686 0.549]);
ylabel('Depth (m)', 'FontWeight','bold','Color','black');
title('V_p (m/s)', 'FontWeight','bold');
set(gca,'XMinorGrid','on','YMinorGrid','on','YDir','reverse','YColor',[0.0431 4 0.5176 0.7804],...
'XAxisLocation','top','XColor',[0.0431 4 0.5176 0.7804]
0.7804),'MinorGridLineStyle','-','FontWeight','demi','FontAngle','italic');

% Plot S wave velocity
subplot(1,3,2);
stairs(vs,zlayer,'LineWidth',2.5,'Color',[1 0 0]);
title('V_s (m/s)', 'FontWeight','bold');
set(gca,'XMinorGrid','on','YMinorGrid','on','YDir','reverse','YColor',[0.0431 4 0.5176 0.7804],...
'XAxisLocation','top','XColor',[0.0431 4 0.5176 0.7804]
0.7804),'MinorGridLineStyle','-','FontWeight','demi','FontAngle','italic');

% Plot Density
subplot(1,3,3);
stairs(ro,zlayer,'LineWidth',2.5,'Color',[0.07059 0.6392 0.07059]);
title('Density (kg/m^3)', 'FontWeight','bold');
set(gca,'XMinorGrid','on','YMinorGrid','on','YDir','reverse','YColor',[0.0431 4 0.5176 0.7804],...
% Plot Geology Model
figure;
set(gcf, 'color', 'white');
pcolor(xx,zz,repmat(vlayer,1,nx)); shading flat; hold on
colormap(hsv); cc=colorbar ('horz'); cc.Label.String='Velocity (m/s)';
axis([0 xmax zmin -0.03*zmax zmax])

% Plot Source-Receiver Group
plot(xs,zs,'r*','markersize',12); hold on
plot(xr,zr,'sk','markersize',4,'markerfacecolor','c'); hold on
xlabel('Distance (m)','FontWeight','bold','Color','black');
ylabel('Depth (m)','FontWeight','bold','Color','black');

%============================================================================
% Run Ray Tracing
%============================================================================
fprintf('--- Starting ray tracing ...
');
wat = waitbar(0,'Raytracing is being processed, please wait...');
%
% Loop over for number of sources
 tic
   for i=1:ns
     % Loop over for number of receivers
     for j=1:nr
       % Loop over for number of layer
       for k=1:nlayer
         % Declare reflection boundary
         if and(zr(j) < zlayer(k),zs(i) < zlayer(k))
            zm = zz(k,:); zf = min(zm)-100000*eps;
         end
         % Downgoing path
         d = find(zlayer > zs(i));
         if(isempty(d)); sdown = length(zlayer); else sdown = d(1)-1; end
         d = find(zlayer > zf);
         if(isempty(d)); edown = length(zlayer); else edown = d(1)-1; end
         zd = [zs(i);zlayer(sdown+1:edown)]; nd = length(zd);
         % Upgoing path
         u = find(zlayer > zr(j));
         if(isempty(u)); sup = length(zlayer); else sup = u(1)-1; end
         u = find(zlayer > zf);
         if(isempty(u)); eup = length(zlayer); else eup = u(1)-1; end
         zu = [zr(j);zlayer(sup:eup)]; nu = length(zu);
         zn = [zd;(flipud(zu))]; nrefl = length(zn)-1;
         % Declare elastic parameter
% Downgoing elastic parameter
vpd = [vp(sdown:edown);vp(edown)];
vsd = [vs(sdown:edown);vs(edown)];
rod = [ro(sdown:edown);ro(edown)];

% Upgoing elastic parameter
vpu = [vp(sup:eup);vp(eup)];
vsu = [vs(sup:eup);vs(eup)];
rou = [ro(sup:eup);ro(eup)];

% Combine model elastic parameter
vpp = [vpd(1:end-1);flipud(vpu(1:end-1))];

vss = [vsd(1:end-1);flipud(vsu(1:end-1))];

vps = [vpd(1:end-1);flipud(vsu(1:end-1))];
rho = [rod(1:end-1);flipud(rou(1:end-1))];

%============================================================================

===
% Start Raytracing (P-P, S-S, or P-S mode)
ops = 1; % ops=1 for PP mode; ops=2 for PS mode
[xh,zh,vh,pp,teta,tt,time] =
shooting_one_source(vpp,vps,zn,xx,xs(i),xr(j),ops);

if zs>=(zlayer(nlayer-1))
layers=nlayer-1;
twt(:,j,i)=tt;
else
layers=nlayer;
twt(k,j,i)=time;
end
theta = abs(teta);

% Plot Ray
if ops == 1
plot(xh,zh,'k-');
title('Seismic Raytracing (P-P mode)','FontWeight','bold');
elseif ops == 2
xd = xh(1:nd+1); xu = xh(nd+1:end);
zd = zh(1:nd+1); zu = zh(nd+1:end);
plot(xd,zd,'k-',xu,zu,'r-');
title('Seismic Raytracing (P-S mode)','FontWeight','bold');
end

%============================================================================

=====
% Compute Reflection Coefficient (Downgoing-Upgoing)
for c=1:nrefl-1
% Reflection Coefficient of Zoeppritz Approximation
[rcl,teta] = zoeppritz(pp,theta(c),vpp(c),vss(c),rho(c),theta(c+1),...
vpp(c+1),vss(c+1),rho(c+1),ops);
rclz(c,j,i) = rcl;
% Reflection Coefficient of Shuey Approximation
[A,B,rc2,teta] = shuey(theta(c),vpp(c),vss(c),rho(c),theta(c+1),...
vpp(c+1),vss(c+1),rho(c+1),ops);
rchs(c,j,i) = rc2; AA(c,j,i) = A; BB(c,j,i) = B;
% Reflection Coefficient of Thomsen Approximation
[rc3,teta] = thomsen(theta(c),vpp(c),vss(c),rho(c),theta(c+1),...
vpp(c+1),vss(c+1),rho(c+1));
rcts(c,j,i) = rc3;
angle(c,j,i) = teta.*(180/pi);
end

%============================================================================

====
end % for horizon/reflector end
xoff = [xoff xr(j)];
waitbar(j/nray,wat)
end % for receiver end
% Save Data
save(['time_shot',num2str(i),'.mat'],'twt');
save(['reflz_shot',num2str(i),'.mat'],'rcz');
save(['refls_shot',num2str(i),'.mat'],'rcs');
save(['reflt_shot',num2str(i),'.mat'],'rct');
save(['teta_shot',num2str(i),'.mat'],'angle');
save(['intercept_shot',num2str(i),'.mat'],'AA');
save(['gradient_shot',num2str(i),'.mat'],'BB');
waitbar(i/ns,wat)
end % for sources end
%============================================================================
============
=======
close(wat);
toc
fprintf(' Ray tracing has succesfully finished...[OK]\n');
xx = repmat(xr,nlayer,1);
% Plot Traveltime
figure;
for i=1:ns
tm = load(['time_shot',num2str(i),'.mat']); tt = tm.twt;
plot(xx(2:nlayer,:),tt(2:nlayer,:,i),'b.'); hold on
xlabel('Horizontal Position (m)','FontWeight','bold','FontAngle','normal','Color','black');
ylabel('Time (s)','FontWeight','bold','FontAngle','normal','Color','black');
title('Traveltime','FontWeight','bold');
set(gca,'Ydir','reverse');
set(gcf,'color','white');
end % Plot Reflection Coefficient
figure;
for i=1:ns
reflz = load(['reflz_shot',num2str(i),'.mat']); reflz = reflz.rcz;
refls = load(['refls_shot',num2str(i),'.mat']); refls = refls.rcs;
reflt = load(['reflt_shot',num2str(i),'.mat']); reflt = reflt.rct;
theta = load(['teta_shot',num2str(i),'.mat']); angle = theta.angle;
%Reflection Coefficient of Zoeppritz
subplot(1,3,1)
plot(abs(angle(1:layers,:,i)),reflz(1:layers,:,i),'r.'); hold on
xlabel('Incidence Angles (degree)','FontWeight','bold','Color','black');
ylabel('Reflection Coefficient','FontWeight','bold','Color','black');
grid on; title('Rpp Zoeppritz','FontWeight','bold','Color','black')
set(gca,'YColor',[0.04314 0.5176 0.7804],'XColor',[0.04314 0.5176 0.7804]);
hold on
%Reflection Coefficient of Shuey
subplot(1,3,2)
plot(abs(angle(1:layers,:,i)),refls(1:layers,:,i),'g.'); hold on
xlabel('Incidence Angles (degree)','FontWeight','bold','Color','black');
grid on; title('Rpp Shuey','FontWeight','bold','Color','black')
set(gca,'YColor',[0.04314 0.5176 0.7804],'XColor',[0.04314 0.5176 0.7804]);
hold on
%Reflection Coefficient of Thomsen
subplot(1,3,3)
plot(abs(angle(1:layers,:,:)), ref(1:layers,:,:), 'b.'); hold on
xlabel('Incidence Angles (degree)', 'FontWeight', 'bold', 'Color', 'black');
grid on; title('Rpp Thomsen', 'FontWeight', 'bold', 'Color', 'black');
set(gca, 'YColor', [0.04314 0.5176 0.7804], 'XColor', [0.04314 0.5176 0.7804]);
hold on
set(gcf, 'color', 'white');
end
for i=1:ns
refls = load(['refls_shot', num2str(i), '.mat']); refls = refls.rcs;
theta = load(['teta_shot', num2str(i), '.mat']); angle = theta.angle;
Rt = load(['intercept_shot', num2str(i), '.mat']); Rt = Rt.AA;
Gt = load(['gradient_shot', num2str(i), '.mat']); Gt = Gt.BB;
Ro(1:layers,:,i) = Rt(1:layers,:,i);
Go(1:layers,:,i) = Gt(1:layers,:,i);
Rc(1:layers,:,i) = refls(1:layers,:,i);
inc(1:layers,:,i) = angle(1:layers,:,i);
end
Rp = reshape(Ro,nr*ns*(layers),1);
G = reshape(Go,nr*ns*(layers),1);
R = reshape(Rc,nr*ns*(layers),1);
teta = reshape(inc,nr*ns*(layers),1);

fprintf('---> Starting AVO Modelling ...
');

% AVO Modelling
fprintf('================================================================================
');
fprintf('================================================================================
');
fprintf('================================================================================
');
fprintf('--- AVO Modelling: Amplitude vs. offset...
');

% Make wavelet ricker (f = 8 Hz)
dt = 0.025; f = 8;
[w, tw] = ricker(dt, f);

% Create zeros matrix for spike's location
tmax = max(tt(:));
tr = 0:dt:tmax; nt = length(tr);
t = tt(2:nlayer,:);

fprintf('--- AVO Modelling: Amplitude vs. offset...
');
%using alternate code (S1R1Layers5.mat)
spikes = zeros(nt,nray);
rcz = real(reflz(1:nlayer-1,:));
for k=1:(nlayer-1)
    for j=1:nray
        ir(k,j) = fix(t(k,j)/dt+0.1)+1;
        spikes(ir(k,j),j) = spikes(ir(k,j),j) + rcz(k,j);
    end
end
% spikes = zeros(nt,nray);
% rcz = real(reflz(1:layers,:));
% for k=1:nlayer
%     for j=1:nray
%         if any(zs)<(zlayer(2))|| any(zs)>=(zlayer(nlayer-1))
%             ir(k,j) = fix(t(k,j)/dt+0.1);
%         else
%             ir(k,j) = fix(t(k,j)/dt+0.1)+1;
%         end
% The +one needed to be taken out because the index exceeded the matrix
% dimensions, this happens when the source is
% located below the lowest reflector or above the first.
% spikes(ir(k,j),j) = spikes(ir(k,j),j) + rcz(k,j);
% end
% end
% Convolve spikes with wavelet ricker
for j=1:nray
    seisz(:,j) = conv2(spikes(:,j),w,'same');
end
ampz=reshape(seisz,length(tr),nr,ns);
for i=1:ns
    ampz_shot(:,:,i) = ampz(:,:,i);
    save(['ampz_shot',num2str(i),'.mat'],'ampz_shot');
end
% CMP Stacked
Stacked = zeros(nt,nr);
for i=1:ns
    files = load(['ampz_shot',num2str(i),'.mat']);
    files = files.ampz_shot;
    Stacked = Stacked + files(:,:,i);
end
% Save Seismic Data
save(['seis','.mat'],'Stacked');
% Plot AVO
figure;
wig(xr./1000,tr,Stacked,'black'); hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Time (s)','FontWeight','bold','Color','black');
title('Receiver Responses','FontWeight','bold','Color','black');
axis tight
set(gca,'YColor',[0.04314 0.5176 0.7804],'XColor',[0.04314 0.5176 0.7804]);
hold on
set(gcf,'color','white');
fprintf(' AVO Modelling has been computed...[OK]\n');
%%
%CROSS-CORRELATION
totalxcorr= ((nr)*(nr-1))/2;
% above calculates all the combinations of receivers needed for the cross
% correlation of all receivers.
crosscorrelation= zeros((length(Stacked)*2)-1,totalxcorrs);
% The size of the cross-correlation matrix is defined by 2 times the length
% of the Stacked data minus 1. The number of columns is defined by the
xc=0;
for station_1=1:nr-1
    for station_2=1:nr-station_1
        xc=xc+1;
        crosscorrelation(:,xc) = 
        xcorr(Stacked(:,station_1),Stacked(:,station_1+station_2));
    end
end
% calculate the distance between each station pair
xc=0;
for station_1=1:nr-1
    for station_2=1:nr-station_1
        xc=xc+1;
        dist(xc)=abs(xr(station_1)-xr(station_1+station_2));
    end
end

% The next task is to sum (Stack) all of the cross-correlations which
% correspond to the same virtual source position. This means that with 100
% receivers there should be 99 cross-correlated traces to sum together for
% each virtual source (i.e. receiver position).
% virtualsource_traces= zeros((length(Stacked)*2)-1,(nr-1),nr);
% for i = 1:(nr)
%     virtualsource_traces(:,:,i)=crosscorrelation(:,1+((nr-1)*(i-1)):((nr-1)*i));
% end
% % Now the traces which must be summed are defined by each number in the
% % third dimension of the virtualsource_traces matrix.
% % A sum will be executed along each row for each virtual source to get the
% % GF.
% StackedGF_vs=zeros((length(Stacked)*2)-1,nr);
% for i = 1:nr
%     for j = 1:(length(Stacked)*2)-1
%         StackedGF_vs(j,i)=sum(virtualsource_traces(j,:,i));
%     end
% end
% % the length is equal to the number of traces times the number of time
% % segments (0:dt:tmax). Where tmax is equal to the largest number in the
% % time vector for the travel times calculated in ray tracing.
% y_axis=linspace(-tmax,tmax,length(crosscorrelation));
% limit=length(find(y_axis>=0.025));
% % figure;
wig(dist./1000,y_axis,crosscorrelation,'blue'); hold on
xlabel('Horizontal Distance (km)', 'FontWeight', 'bold', 'Color', 'black');
ylabel('Time (s)', 'FontWeight', 'bold', 'Color', 'black');
title('GF Obtained Through Cross-Correlation', 'FontWeight', 'bold', 'Color', 'black');
axis tight
figure;
wig(dist./1000,y_axis(limit:end),crosscorrelation(limit:end,:),'blue'); hold on
xlabel('Horizontal Distance (km)','FontWeight','bold','Color','black');
ylabel('Time (s)','FontWeight','bold','Color','black');
title('Casual GF Obtained Through Cross-Correlation','FontWeight','bold','Color','black');
axis tight
11.2. Data Retrieval

%% For data retrieval
close all
clear
clc

yr=2009; % pick the year and month you would like to gather data for
month=[01 02 03 04 05 06 07 08 09 10 11 12];
days_in_month=[eomday(yr,1) eomday(yr,2) eomday(yr,3) eomday(yr,4)
eomday(yr,5) eomday(yr,6) eomday(yr,7) eomday(yr,8) eomday(yr,9)
eomday(yr,10) eomday(yr,11) eomday(yr,12)];
tic; % start the timer
day=1;
mn=1; % is january
for j = 1:days_in_month(mn) % here the index for days matches the month of
data you are retrieving
    ii=1;
    lenn=1;
    hr=[00 12 24];
    for i=1:2 % number of segments which compose one day of data ie. 00-12 and
        12-24 = 2 segments
            start_time=[ yr month(mn) day hr(ii) 00 00];
            end_time]=[ yr month(mn) day hr(ii+1) 00 00];
            t1=datenum(start_time);
            t2=datenum(end_time);
            % retrieve data from IRIS fetch
            mytrace=irisFetch.Traces('TA','G18A','*','BHZ',t1,t2,'includePZ');
            del=isempty(mytrace);
            if del==0
                data_trace=mytrace.data;
                dtt_trace=data_trace; % duplicate the data for use in the length
                variable
                day_trace(lenn:lenn+length(dtt_trace)-1)=data_trace;
                lenn=length(dtt_trace)+1;
                % get the PZ information for each station
                zz=mytrace.sacpz.zeros;% locate the zeros
                pp=mytrace.sacpz.poles;% locate the poles
                constant=mytrace.sacpz.constant;% determine the constant
                ii=ii+1; % increment to the next segment of time to be added to the
                day long trace.
            else
                data_trace=[];
                pp=[];
                zz=[];
                constant=[];
            end
        end
    end
end
% cut all data to the same length for one day = 24*60*60*40
len_dtrace=length(day_trace);
if len_dtrace>=24*3600*40
    cut_dtrace= day_trace(1:(24*3600*40));
else
    cut_dtrace=[];
    display(['empty day number=', num2str(j)])
end
file_name=['STATION_DATA\G18A_', num2str(yr), '_', num2str(month(mn)), '_', num2str(day), '.mat'];
    save(file_name, 'cut_dtrace', 'pp', 'constant', 'zz')

display(['day number=', num2str(j)]);
display(['data size = ', num2str(len_dtrace)]);
clear day_trace file_name
day=day+1;
end
toc;
11.3. PSD

close all
clear
clc

yr=2009; % pick the year and month you would like to gather data for
month= [01 02 03 04 05 06 07 08 09 10 11 12];
days_in_month=[eomday(yr,1) eomday(yr,2) eomday(yr,3) eomday(yr,4)
eomday(yr,5) eomday(yr,6) eomday(yr,7) eomday(yr,8) eomday(yr,9)
eomday(yr,10) eomday(yr,11) eomday(yr,12)];

mn=1; %ie. January
npts=24*3600*40;
Fs=40; %samples per second = 40Hz, sample rate and sample frequency are the
same thing
overhr=47;

%Create a matrix containing sequentially ordered vectors of length equal to
%the stacking time window.
tw=3600; %length of stacking window in time (s)
N=tw*Fs;
oct_n=97;

monthaverage=zeros(oct_n,overhr,days_in_month(mn));
tic
for day=1:days_in_month(mn)
    close all
    file_name=['STATION_DATA\F17A_2009_1_',num2str(day),'.mat'];
    trace=load(file_name);
    data=trace.cut_dtrace;
    pp=trace.pp;
    zz=trace.zz;
    constant=trace.constant;

    hr= [00 24];
    start_time=[ yr month(mn) day hr(1) 00 00];
    end_time = [yr month(mn) day hr(2) 00 00];
    t1=datenum(start_time);
    t2=datenum(end_time);
    sampletimes = linspace(t1,t2,length(data));
    plot(sampletimes, data)
    datetick;
    ylabel('Digital Counts');
    xlabel('Time (hh:mm)')
    title(['Raw Seismogram for Transportable Array station F15A, Starting
at:', datestr(t1)]);

    %Adjusting the record length
    npts=length(data);
tw=3600; %length of stacking window in time (s)
N=tw*Fs;

    %split the data into hour long segments that overlap by 50%
    rl=zeros(N,((npts)/(N/2))-1);
    for n=1:((npts)/(N/2))-1
if n==1
    rl(:,n)=data(1:N);
else
    rl(:,n)=data((N/2)*(n-1)+1:((N/2)*(n-1))+N));
end
end

% Preprocessing

np = nextpow2(N);
% hi = 2^np;
lo = 2^(np-1);

% new number for operations
op=lo;

% truncate the data to a new length of op
for n=1:((npts)/(N/2))-1
    rlt(:,n)=rl(1:op,n);
end

% separate the data into 15 minute segments that overlap by 75%
NN=op/4;
mm=NN/4;
for n=1:((npts)/(N/2))-1
    for j=1:(op/mm)-3
        if j==1
            segments(:,j,n)=rlt(1:NN,n);
        else
            segments(:,j,n)=rlt((mm)*(j-1)+1:(((mm)*(j-1))+NN),n);
        end
    end
end

Th=op/Fs;
Tr=Th/4;
Nt=Fs/2;
dt=1/Fs;
dt2=1/Nt;

% figure
% plot(0:1/Fs:Tr-(1/Fs),segments(:,1,1))
% xlabel('Seconds (s)')
% ylabel('Digital Counts')
% title('Raw Data after Record length adjustment')
% axis tight

% PROCESSING

for hh=1:((npts)/(N/2))-1
    for j=1:(op/mm)-3
        % generate the mean for each segment
        smean(j,hh)=(1/NN)*sum(segments(:,j,hh));
        X(:,j,hh)=segments(:,j,hh)-smean(j,hh);
    end
end
for hh=1:((npts)/(N/2)) - 1
    for j=1:(op/mm)-3
        X(:,j,hh)=detrend(X(:,j,hh));
    end
end
%Bandpass the data
bp=bp_bu_co(X,1/1000,19.5,Fs,2,1);
% figure
% plot(0:1/Fs:Tr-(1/Fs),bp(:,l,1))
% xlabel('Seconds (s)')
% ylabel('Digital Counts')
% title('Data after Mean and Trends Removed with a Bandpass Filter')
% axis tight
%apply a Tukey window to apply a 10% cosine Taper
window=tukeywin(NN,0.1);
for hh=1:((npts)/(N/2)) - 1
    for j=1:(op/mm)-3
        tap_X(:,j,hh)=bp(:,j,hh).*window;
    end
end
% figure
% plot(0:1/Fs:Tr-(1/Fs),tap_X(:,l,1))
% xlabel('Seconds (s)')
% ylabel('Digital Counts')
% title('Data after 10% Cosine Taper')
% axis tight

%Calculate the power ratio of the raw data to the tapered data
for hh=1:((npts)/(N/2)) - 1
    BP1=mean(bandpower(X(:,:,hh)));
    BP2=mean(bandpower(tap_X(:,:,hh)));
    ratio(hh)=BP1/BP2;
end
%show what the data looks like in acceleration and use and alternate method
%for calculating the PSD
%%
%PSD
%1. Calculate the fourier transform
for hh=1:((npts)/(N/2)) - 1
    for j=1:(op/mm)-3
        FC(:,j,hh)=fft(tap_X(:,j,hh))./(dt*4); %FC(:,j,hh)=fft(tap_X(:,j,hh));
    end
end
%kill half the frequency spectrum
FC2=FC(1:NN/2,:,:);
%define new number of points in the frequency domain
NF=length(FC2);

% define the frequency axis
for n=1:NF
freq(n) = n/(NN*dt);
end
% define the time axis
time = 1./freq;
% figure
% plot(freq, abs(FC2(:,1,1)))
% xlabel('Frequency (Hz)')
% ylabel('Amplitude in Digital Counts (m)/Hz')
% title('Amplitude Spectrum of a Segment of Data in Digital Counts')
% axis tight
% change the data into velocity
for hh = 1:((npts)/(N/2))-1
    for j = 1:(op/mm)-3
        for n = 1:NF
            omega = 2*pi*(Fs/2/NF) * n;
            vel_f(n,j,hh) = real(FC2(n,j,hh)) * omega * 1i - imag(FC2(n,j,hh)) * omega;
        end
    end
end
% change the data into acceleration
for hh = 1:((npts)/(N/2))-1
    for j = 1:(op/mm)-3
        for n = 1:NF
            omega = 2*pi*(Fs/2/NF) * n;
            acc_f(n,j,hh) = real(vel_f(n,j,hh)) * omega * 1i - imag(vel_f(n,j,hh)) * omega;
        end
    end
end
% figure
% plot(freq, abs(acc_f(:,1,1)))
% xlabel('Frequency (Hz)')
% ylabel('Amplitude in Digital Counts *(m/s^2)/Hz')
% title('Amplitude Spectrum of a Segment of Data in Acceleration')
% axis tight
% Normalize the frequency by taking the square of it's absolute value and
% multiplying it by 2*dt/N
for hh = 1:((npts)/(N/2))-1
    % sum the power series to one hour in length, Stacking with
    % normalization proportional to the number of units used for stacking
    % (ie. 13) for 1 hour of data
    ss = acc_f(:, :, hh);
    hourstack = sum(ss, 2);
    hourstack = hourstack';
    power(:, hh) = (1/((op/mm)-3)).*hourstack;
end
% multiply the power by the ratio
for hh = 1:((npts)/(N/2))-1
    PSD(:, hh) = power(:, hh).*ratio(hh);
end
% 3] remove instrument response using the pz file from the data
for hh = 1:((npts)/(N/2))-1
for j=1:NF
    freq2 = (j) / (dt * NF); % try NF here if results are weird
    tmp = generate_response(zz, pp, constant, freq2);
    noinst_PSD(j, hh) = zdiv(PSD(j, hh), tmp);
end

% figure
% plot(freq, abs(noinst_PSD(:,1)))
% xlabel('Frequency (Hz)')
% ylabel('Amplitude (m/s^2)/Hz')
% title('Amplitude Spectrum of a One Hour Segment of Data After Instrument Response Is Removed')
% axis tight

% display the time-series for each hour long segment following an inverse fourier transform
for hh=1:(npts)/((N/2))-1
    inv(:, hh) = ifft(noinst_PSD(:, hh));
    win = tukeywin(length(inv(:, hh)), 0.1);
    abs_inv(:, hh) = abs(inv(:, hh));
    win_inv(:, hh) = abs_inv(:, hh) .* win;
end

% figure
% plot(win_inv)
% Calculate the real power values
PSD_real = 10*log10((real(noinst_PSD) .* real(noinst_PSD) + imag(noinst_PSD) .* imag(noinst_PSD)) * dt / 2 / length(noinst_PSD));
% PSD = 10*log10(noinst_PSD);

% figure
% semilogx(time, PSD_real(:,1))
% xlabel('Period (s)')
% ylabel('Power 10*log10 (m/s^2)/Hz or dB')
% title('Power Spectrum of 1 Hour of Data')
% axis tight

%%
% PDF

% Average the power over full octaves in 1/8 octave increments
time = fliplr(time);

% calculate the vectors for the short period corner
for o=1:NF
    if o == 1
        ts(o) = dt2;
        tl(o) = 2 * ts(o);
        tc(o) = sqrt(ts(o) * tl(o));
    else
        ts(o) = ts(o-1) * (2^(1/8));
        tl(o) = 2 * ts(o);
    end
tc(o)=\sqrt{ts(o)\times tl(o)};
\%
if \quad ts(o) \geq (Tr/9)
   if \quad ts(o) \geq Tr/4
      break
   end
end
end

% match where the times correspond between the PSD vector and the octaves
PSD_flip=flipud(PSD_real);
oct_n=length(ts);
for hh=1:((npts)/(N/2))-1
for n=1:oct_n
   start=NF-length(find(time>=ts(n)))+1;
   finish=length(find(time<=tl(n)));
   duration=length(start:finish);
   average(n,hh)=(sum(PSD_flip(start:finish,hh)))/duration;
end
end

% figure
% semilogx(tc,average)
% xlabel('Period (s)')
% ylabel('Power 10*\log_{10} (m/s^2)/Hz or dB')
% title('PSD diagram Containing 1 Day of Data Composed of 1 Hour Segments with 50% Overlap')
% axis tight
monthaverage(:,:,day)=average(:,:,);
clear abs_inv acc_f astack average bp BP1 BP2 constant data dt dt2 duration
end_time FC FC2 file_name fish freq freq2 HH hi hourstack hr inv j lo mm n N
NF NN noinst_PSD np Nt o omega op power pp PSD PSD_flip PSD_real ratio rl
rlt sampletimes segments smean ss start_time t1 t2 tap_X Th time tl tmp
Tr trace ts tw vel_f win win_inv window X zz
display(['day number=', num2str(day)]);
toc
end
tot_h=days_in_month(mn)*overhr;
monthPSD=reshape(monthaverage,97,tot_h);
save('pdf_F17.mat','tc','monthPSD')
11.4. PDF

close all
clear
clc

% To compute the PDF
% load all of the data from the PDF_complete.m file
tic
tic
tic
tic
tic
tic
tic
tic
tic
tic
tic
tic
% load all the station data
a14=load('pdf_A14.mat');
b14=load('pdf_B14.mat');
e14=load('pdf_E14.mat');
f14=load('pdf_F14.mat');
g14=load('pdf_G14.mat');
toc
a15=load('pdf_A15.mat');
c15=load('pdf_C15.mat');
e15=load('pdf_E15.mat');
f15=load('pdf_F15.mat');
g15=load('pdf_G15.mat');
toc
b16=load('pdf_B16.mat');
d16=load('pdf_D16.mat');
e16=load('pdf_E16.mat');
g16=load('pdf_G16.mat');
h16=load('pdf_H16.mat');
toc
b17=load('pdf_B17.mat');
c17=load('pdf_C17.mat');
d17=load('pdf_D17.mat');
f17=load('pdf_F17.mat');
g17=load('pdf_G17.mat');
toc
a18=load('pdf_A18.mat');
b18=load('pdf_B18.mat');
d18=load('pdf_D18.mat');
e18=load('pdf_E18.mat');
f18=load('pdf_F18.mat');
g18=load('pdf_G18.mat');

% put all PSD's into one big matrix
monthPSD(:,:,1)=a14.monthPSD;
monthPSD(:,:,2)=a15.monthPSD;
monthPSD(:,:,3)=a18.monthPSD;
monthPSD(:,:,4)=b14.monthPSD;
monthPSD(:,:,5)=b16.monthPSD;
monthPSD(:,:,6)=b17.monthPSD;
monthPSD(:,:,7)=b18.monthPSD;
monthPSD(:,:,8)=c15.monthPSD;
monthPSD(:,:,9)=c17.monthPSD;
monthPSD(:,:,10)=d16.monthPSD;
monthPSD(:,:,11)=d17.monthPSD;
monthPSD(:,:,12)=d18.monthPSD;
monthPSD(:,:,13)=e14.monthPSD;
monthPSD(:,:,14)=e15.monthPSD;
monthPSD(:,:,15)=e16.monthPSD;
monthPSD(:,:,16)=e18.monthPSD;
monthPSD(:,:,17)=f14.monthPSD;
monthPSD(:,:,18)=f15.monthPSD;
monthPSD(:,:,19)=f17.monthPSD;
monthPSD(:,:,20)=f18.monthPSD;
monthPSD(:,:,21)=g14.monthPSD;
monthPSD(:,:,22)=g15.monthPSD;
monthPSD(:,:,23)=g16.monthPSD;
monthPSD(:,:,24)=g17.monthPSD;
monthPSD(:,:,25)=g18.monthPSD;
monthPSD(:,:,26)=h16.monthPSD;

tot_h=47*31*26;
monthPSD=reshape(monthPSD,97,tot_h);

%%
% BINNING THE DATA INTO 1 DB SEGMENTS
% Define bin edges
binedges=linspace(-210,-80,130+1); %(deine the low power and high power, then the difference between the two)
% round each of the power values to a whole number
for h=1:tot_h
    round_PSDs(:,:,h)=round(monthPSD(:,:,h));
end
[numcentral hours]=size(round_PSDs);
%determine the number of values in each hour long segment which fall into each 1Db bin. Simultaneously determines which bin each value fell into.
for n=1:numcentral
    [Np,whichbin]=histc(round_PSDs(n,:),binedges);
    % make a matrix out of the quantities pertaining to the number of estimates which fall into each bin
    quant_tc(:,:,n)=Np;
    whichbin=whichbin;
    % create the matrix which will be a function of period and power with weight based on the power density function.
    pdf_matrix(:,:,n)=quant_tc(:,:,n)./tot_h;
end
% load in one of the central period files. (Assuming all of them are the same)
time=a14.tc;
% plot some sample periods
figure
plot(binedges,quant_tc(:,:,5),'r',binedges,quant_tc(:,:,37),'y',binedges,quant_tc(:,:,61),'g',binedges,quant_tc(:,:,85))
% hold on
% plot(binedges,quant_tc(:,:,37))
% plot(binedges,quant_tc(:,:,61))
% plot(binedges,quant tc(:,:,93))
xlabel('Power [10log10 (m/s^2)^2] dB')
ylabel('Number of Occurrences')
title('Frequency Distribution Plots Using 1dB Bins, at 4 Different Period Bands')
legend('Period =0.1s','Period=1.6s','Period=12.8s','Period=102.5s','Location','northeast')
axis tight
% Plot the contour map of the PDF with respect the period on the x axis and range in dB on the y axis (number of bins and = to 'binedges' vector)
figure
probability=linspace(0.00,0.3,10000);
ticks=(0:0.02:0.3);
contour(time,binedges,pdf_matrix,probability);
%change the x axis to log scale.
%generate color map
C=[101,200,182;
104,201,179;
106,202,177;
108,203,174;
111,204,172;
113,204,170;
116,205,167;
118,206,165;
121,207,163;
123,208,160;
125,209,158;
128,210,155;
130,210,153;
133,211,151;
135,212,148;
138,213,146;
140,214,144;
142,215,141;
145,216,139;
147,216,136;
150,217,134;
152,218,132;
155,219,129;
157,220,127;
159,221,125;
162,222,122;
164,223,120;
167,223,117;
169,224,115;
172,225,113;
174,226,110;
176,227,108;
179,228,106;
181,229,103;
184,229,101;
186,230,98;
189,231,96;
191,232,94;
193,233,91;
196,234,89;
198,235,87;
201,235,84;
203,236,82;
205,237,79;
208,238,77;
210,239,75;
213,240,72;
215,241,70;
218,241,68;
220,242,65;
222,243,63;
225,244,60;
227, 245, 58;  
230, 246, 56;  
232, 247, 53;  
235, 247, 51;  
237, 248, 49;  
239, 249, 46;  
242, 250, 44;  
244, 251, 41;  
247, 252, 39;  
249, 253, 37;  
252, 253, 34;  
254, 254, 32;  
255, 254, 30;  
255, 252, 30;  
255, 250, 29;  
255, 248, 29;  
255, 246, 28;  
255, 244, 28;  
255, 242, 27;  
255, 240, 27;  
255, 238, 26;  
255, 236, 26;  
255, 234, 25;  
255, 232, 25;  
255, 230, 24;  
255, 228, 24;  
255, 226, 23;  
255, 224, 23;  
255, 222, 22;  
255, 219, 22;  
255, 217, 22;  
255, 215, 21;  
255, 213, 21;  
255, 211, 20;  
255, 209, 20;  
255, 207, 19;  
255, 205, 19;  
255, 203, 18;  
255, 201, 18;  
255, 199, 17;  
255, 197, 17;  
255, 195, 16;  
255, 193, 16;  
255, 191, 15;  
255, 189, 15;  
255, 186, 14;  
255, 184, 14;  
255, 182, 13;  
255, 180, 13;  
255, 178, 13;  
255, 176, 12;  
255, 174, 12;  
255, 172, 11;  
255, 170, 11;  
255, 168, 10;  
255, 166, 10;  
255, 164, 9;
<table>
<thead>
<tr>
<th>Value</th>
<th>X-Position</th>
<th>Y-Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>176</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>173</td>
<td>47</td>
<td>0</td>
</tr>
<tr>
<td>171</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>169</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>167</td>
<td>41</td>
<td>0</td>
</tr>
<tr>
<td>165</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>163</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>161</td>
<td>35</td>
<td>0</td>
</tr>
<tr>
<td>159</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>157</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>155</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>153</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>148</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>146</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>144</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>142</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>140</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>138</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>136</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>134</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>132</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>130</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>128</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>125</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>123</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>121</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>119</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>118</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>116</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>114</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>108</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>104</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>102</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>98</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>96</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>94</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>93</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>85</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>83</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>81</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>79</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>77</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>73</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>66</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
axis xy
set(gca,'XScale','log')
colormap(C/255)
cc=colorbar('YTick',ticks,'YTickLabel',ticks);
cc.Label.String = 'Probability';
cc.Label.FontSize = 16;
%set(get(cc,'title'),'String','Probability');
colormap(C/255);
caxis([probability(1) probability(length(probability))]);
xlabel('Period (s)')
ylabel('Power [10log10 (m/s^2)^2 ] dB')
title('PSD and Probability Density Function For All Stations from January 1-31, 2009')
grid on
11.5. Preprocessing for Ambient Noise Cross-correlations

% this is the code for preprocessing the data retrieved using data_get and % calculating the Power Spectral Density

close all
clear
clc

yr=2009; % pick the year and month you would like to gather data for
month= [01 02 03 04 05 06 07 08 09 10 11 12];
days_in_month=[eomday(yr,1) eomday(yr,2) eomday(yr,3) eomday(yr,4)
eomday(yr,5) eomday(yr,6) eomday(yr,7) eomday(yr,8) eomday(yr,9)
eomday(yr,10) eomday(yr,11) eomday(yr,12)];
mn=1;%ie. January
npts=24*3600*40;
Fs=40;%samples per second = 40Hz, sample rate and sample frequency are the same thing

monthdata=zeros(npts,days_in_month(mn));
tic
for day=15:15%days_in_month(mn)
close all
    file_name=['STATION_DATA\G14A_2009_1_',num2str(day),'.mat'];
    trace=load(file_name);
data=trace.cut_dtrace;
    pp=trace.pp;
    zz=trace.zz;
    constant=trace.constant;

    hr= [00 24];
    start_time=[ yr month(mn) day hr(1) 00 00];
    end_time = [yr month(mn) day hr(2) 00 00];
    t1=datenum(start_time);
    t2=datenum(end_time);
    sampletimes = linspace(t1,t2,length(data));
    plot(sampletimes, data)
datetick;
ylabel('Digital Counts/Instrument Sensitivity');
xlabel('Time (hh:mm)');
title(['Raw Seismogram for Transportable Array station F15A, Starting at:', datestr(t1)]);

%Adjusting the record length
npts=length(data);

%1. Remove the mean from each hour long segment
smean=(1/npts)*sum(data);
meanless=data-smean;

%2. Detrend the data
trendless=detrend(meanless);

%3. apply a Tukey window to apply a 10% cosine Taper
window=tukeywin(npts,0.1);
tapered=trendless.*window';

% figure
% plot(sampletimes,tapered)
% datetick
% xlabel('Time (hh:mm)')
% ylabel('Digital Counts')
% title('Raw Data processing up to instrument response removal')
% axis tight

%define the time change between samples
dt=1/Fs;

% 4. Calculate the fourier transform
FC=fft(tapered);

% define the frequency axis
freq=zeros(npts,1);
for j=1:npts
    freq(j)=j/(npts*dt);
end

% %define the time axis
time=1./freq;

% To change the data into acceleration you need to integrate twice in the
% frequency domain

%change data from distance into velocity
omega=zeros(npts,1);
vel_f=zeros(npts,1);
for j=1:npts
    omega(j) = 2*pi*(Fs/2/npts) * j;
    vel_f(j) = (real(FC(j)) * omega(j) * 1i - imag(FC(j)) * omega(j));
end

%change data from velocity to acceleration
omega=zeros(npts,1);
acc_f=zeros(npts,1);
for j=1:npts
    omega(j) = 2*pi*(Fs/2/npts) * j;
    acc_f(j) = (real(vel_f(j)) * omega(j) * 1i - imag(vel_f(j)) * omega(j));
end

figure
plot(freq(1:npts/2),abs(acc_f(1:npts/2,1)))
xlabel('Frequency (Hz)')
ylabel('Digital Counts (m/s^2)/Hz')
title('Amplitude spectrum of Data after transformaiton to Acceleration')
axis tight
% remove instrument response using the pz file from the data
noinst=zeros(npts,1);
for j=1:npts
    freq2 = (j) / (dt * npts);
    tmp=generate_response(zz,pp,constant,freq2);
    noinst(j) = (zdiv(acc_f(j),tmp));
end
toc

figure
plot(freq(1:npts/2),abs(noinst(1:npts/2)))
xlabel('Frequency (Hz)')
ylabel('Acceleration (m/s^2)^2/Hz')
title('Amplitude spectrum of Data after instrument response is removed')
axis tight

% phasel_data=ifft(noinst,npts,'symmetric');
figure
plot(sampletimes,phasel_data)
datetick
xlabel('Time (hh:mm)')
ylabel('Acceleration (m/s^2)')
title('Data With instrument response removed')

% bp=bp_bu_co(phasel_data,1/150,1/7,Fs,2,1);
figure
plot(sampletimes,bp)
datetick
xlabel('Time (hh:mm)')
ylabel('Acceleration (m/s^2)')
title('Bandpassed Data to Be Used In Temporal Normalization')

%%
% temporal normalization

% calculate the running average of the wavelet in the normalization window
N=75; % half of the max period of the bandpass which is applied to the
data prior to cross-correlation. (7s-150s)
% tw=80; % normalization time increment window, 80 seconds according to
IRIS

numnorm=npts/(N*Fs);
weight=zeros(numnorm,1);
for j=1:numnorm
    bp2sum=bp(((j-1)*(N*Fs))+1:(j*(N*Fs)));
    weight(j)=(1/((2*N)+1))*sum(abs(bp2sum)); % calculated the weighted
    average for each 80 second long time window starting at the first sample.
end

normdatamatrix=zeros(N*Fs,numnorm);
for j=1:numnorm
    windata=phasel_data(((j-1)*(N*Fs))+1:(j*(N*Fs)));
    normdatamatrix(:,j)=windata./weight(j);
end
% reshape the normalized data into a vector
normdata=reshape(normdatamatrix,npts,1);
normdata=normdata.*window;

figure
plot(sampletimes,normdata)
datetick
title('Normalized Data')
xlabel('Time (hh:mm)');
ylabel('Acceleration (m/s^2)');

% bpnorm=bp_bu_co(normdata,1/150,1/7,Fs,2,1);
figure
plot(sampletimes,bpnorm)
datetick
title('Bandpassed Normalized Data')
xlabel('Time (hh:mm)');
ylabel('Acceleration (m/s^2)');

%% SPECTRAL WHITENING
spectdata=bp_bu_co(normdata,1/150,1/7,Fs,2,1);

% df=1 / (1*(dt)*(npts-1)); % Hz frequency increments
Y=(fft(spectdata,npts)); % Fourire Transform to the frequency domain

% plot the amplitude spectrum from 0 to 5 seconds or 0 to 0.2 Hz
limit=length(find(freq<=1/5));
figure
plot(freq(1:limit-1),abs(Y(2:limit)));
title('Amplitude Spectrum of Bandpassed Data used for Spectral Whitening')
xlabel('Frequency (Hz)');
ylabel('Amplitude');

% SMOOTHING SECTION WITH PMTM (multitaper Thomson algorithm)
% [pxx,w] = pmtm(x,nw,nfft) returns the normalized frequency vector, w.
% If pxx is %a one-sided PSD estimate, w spans the interval [0,?] if nfft is even
% and % [0,?] if nfft is odd. If pxx is a two-sided PSD estimate, w spans the % interval [0,2?).Here 'x' is the data, 'nw'=Time-halfbandwidth product, % % specified as a positive scalar.'nfft' is the number of dft points.
Pxx = pmtm(spectdata,4,npts);
figure
plot(freq(1:limit-1),Pxx(2:limit))
title('Partial Spectral Density Estimate Calculated by Multitaper Thomson Algorithm')
xlabel('Frequency (Hz)');
ylabel('Amplitude');
% toc
Pxx=Pxx(2:end)';
Pxx_left = fliplr(Pxx(1:length(Pxx)));  
Pxx_total = [Pxx_left Pxx];  

Pxx_phase = angle(Y);  % note: phase doesn't change with smoothing  
Pxx_mag = sqrt(ifftshift(Pxx_total));  % THIS IS THE SMOOTHED magnitude  

%%% SCALE THE ORIGINAL AND SMOOTHED AMPLITUDES  
Y = Y / max(Y);  
Pxx_mag = Pxx_mag / max(Pxx_mag);  

%%% WHITENING SECTION: we divide the original magnitude with the smoothed one  
white_mag = Y ./ Pxx_mag;  
white_freq = white_mag .* exp(1i * Pxx_phase);  % this is the 'prewhitened' signal in frequency space  
white_time = ifft(white_freq, npts, 'symmetric');  % this is the prewhitened signal in time domain  

figure  
plot(freq(1:limit-1), abs(white_freq(2:limit)));  
title('Amplitude Spectrum of Bandpassed Data After Spectral Whitening')  
xlabel('Frequency (Hz)')  
ylabel('Amplitude')  

%% Observe the new amplitude spectrum after whitening  
bpwhite = bp_bu_co(white_time, 1/150, 1/7, Fs, 2, 1);  
B = fft(bpwhite, npts);  
figure  
plot(freq(1:limit-1), abs(B(2:limit)));  
title('Amplitude Spectrum of Bandpassed Data After Spectral Whitening')  
xlabel('Frequency (Hz)')  
ylabel('Amplitude')  

% taper the final preprocessed data  
finaldata = white_time .* window;  
figure  
plot(sampletimes, white_time)  
datetick  
title('Preprocessed Seismic Data Prior to Cross-Correlation')  
xlabel('Time (hh:mm)')  
ylabel('Ground Motion Acceleration (m/s^2)')  

%%  

% Calculate the real power values  
PSD_real = 10*log10((real(noinst).*real(noinst) + imag(noinst).*imag(noinst)) * dt / 2 / length(noinst));  

figure  
semilogx(time(1:npts/2), PSD_real(1:npts/2))  
xlabel('Period (s)')  
ylabel('Power 10*log10 (m/s^2)^2/Hz')  
title('Power Spectrum of Preprocessed Data')  
axis tight  

% save the data for each day  
monthdata(:, day) = white_time;  
display([day number = ', num2str(day)]);
%clear acc_f B bp bp2sum bpnorm bpwhite constant data df dt end_time FC
file_name finaldata freq freq2 hr j limit meanless N noinst normdata
normdatamatrix numnorm omega phasel_data pp PSD_real Pxx Pxx_left Pxx_mag
Pxx_phase Pxx_total sampletimes smean spectdata start_time t1 t2 tapered time
tmp trace trendless vel_f weight white_freq white_mag white_time windata
window Y zz
toc
doc
end
11.6. Cross-correlations and Stacking

% close all
% clear
% clc

yr=2009; % pick the year and month you would like to gather data for
month= [01 02 03 04 05 06 07 08 09 10 11 12];
days_in_month=[eomday(yr,1) eomday(yr,2) eomday(yr,3) eomday(yr,4)
eomday(yr,5) eomday(yr,6) eomday(yr,7) eomday(yr,8) eomday(yr,9)
eomday(yr,10) eomday(yr,11) eomday(yr,12)];
mn=1;

tic; % start the timer

% load all the station data
a14=load('fullA14.mat');
b14=load('fullB14.mat');
e14=load('fullE14.mat');
f14=load('fullF14.mat');
g14=load('fullG14.mat');
toc
a15=load('fullA15.mat');
c15=load('fullC15.mat');
e15=load('fullE15.mat');
f15=load('fullF15.mat');
g15=load('fullG15.mat');
toc
a16=load('fullA16.mat');
b16=load('fullB16.mat');
c16=load('fullC16.mat');
e16=load('fullE16.mat');
g16=load('fullG16.mat');
toc
a17=load('fullA17.mat');
c17=load('fullC17.mat');
b17=load('fullB17.mat');
e17=load('fullE17.mat');
g17=load('fullG17.mat');
toc
a18=load('fullA18.mat');
b18=load('fullB18.mat');
e18=load('fullE18.mat');
f18=load('fullF18.mat');
g18=load('fullG18.mat');
toc

% define the sampling frequency for all the stations
Fs=40;
% define the lag time you would like to view
lagtime=5000;
% get the distance between each pair of receivers
stationlocations = 'stationlocations.xlsx';
loc = xlsread(stationlocations);
numxcorr=sum(1:length(loc)-1); % determine how many cross-correlations will be performed.
dist=zeros(numxcorr,1);
xc=0;
for stlnum=1:length(loc)-1
  for st2num=1:length(loc)-stlnum
xc=xc+1;

dist(xc)=lat_longd(loc(st1num,1),loc(st1num,2),loc(st1num+st2num,1),loc(st1num+st2num,2));

end

% seems like the most resonable presentation is in velocity with normalized % values
npts=40*3600*24;

%Make a matrix of data for each station in the same order as the distances %listed according to the lat long file.
tic
for day = 13:days_in_month(mn)
    stationacc=zeros(npts,length(loc));
    stationacc(:,1)=a14.monthdata(:,day);
    stationacc(:,2)=b14.monthdata(:,day);
    stationacc(:,3)=e14.monthdata(:,day);
    stationacc(:,4)=f14.monthdata(:,day);
    stationacc(:,5)=g14.monthdata(:,day);
    stationacc(:,6)=a15.monthdata(:,day);
    stationacc(:,7)=c15.monthdata(:,day);
    stationacc(:,8)=e15.monthdata(:,day);
    stationacc(:,9)=f15.monthdata(:,day);
    stationacc(:,10)=g15.monthdata(:,day);
    stationacc(:,11)=b16.monthdata(:,day);
    stationacc(:,12)=d16.monthdata(:,day);
    stationacc(:,13)=e16.monthdata(:,day);
    stationacc(:,14)=g16.monthdata(:,day);
    stationacc(:,15)=h16.monthdata(:,day);
    stationacc(:,16)=b17.monthdata(:,day);
    stationacc(:,17)=c17.monthdata(:,day);
    stationacc(:,18)=d17.monthdata(:,day);
    stationacc(:,19)=e17.monthdata(:,day);
    stationacc(:,20)=f17.monthdata(:,day);
    stationacc(:,21)=g17.monthdata(:,day);
    stationacc(:,22)=b18.monthdata(:,day);
    stationacc(:,23)=d18.monthdata(:,day);
    stationacc(:,24)=e18.monthdata(:,day);
    stationacc(:,25)=f18.monthdata(:,day);
    stationacc(:,26)=g18.monthdata(:,day);
    toc
end
% %transform data into velocity
vel=zeros(npts,26);
for jj=1:26
    vel(:,jj)=freq_integrate(stationacc(:,jj),Fs);
end
clear stationacc
%bandpass the data between 40 and 100 s
bp(:,:,1)=bp_bu_co(vel(:,:,1),1/50,1/15,Fs,2,1);

clear vel
%bp_MONTH=zeros(length(-lagtime:lagtime),numxcorr,days_in_month(mn));
bpcross=zeros(length(-lagtime:lagtime),numxcorr);
xc=0;
for station_1=1:length(loc)-1
    for station_2=1:length(loc)-station_1
        xc=xc+1;
        bpcross(:,xc)=cross_correlate(bp(:,station_1),bp(:,station_1+station_2),Fs,Fs
            ,lagtime,'norm');
    end
end
display(['day number=',num2str(day)]);
file_name=[STATION_DATA\Cross_50_15s',num2str(day),'.mat'];
save(file_name,'bpcross')
clear bpcross bp
toc
end

close all
clear
clc
tic
yr=2009; % pick the year and month you would like to gather data for
month=[01 02 03 04 05 06 07 08 09 10 11 12];
days_in_month=[eomday(yr,1) eomday(yr,2) eomday(yr,3) eomday(yr,4)
    eomday(yr,5) eomday(yr,6) eomday(yr,7) eomday(yr,8) eomday(yr,9)
    eomday(yr,10) eomday(yr,11) eomday(yr,12)];
mn=1;

% define the sampling frequency for all the stations
Fs=40;
% define the lag time you would like to view
lagtime=5000;
% get the distance between each pair of receivers
stationlocations = 'stationlocations.xlsx';
loc = xlsread(stationlocations);
numxcorr=sum(1:length(loc)-1);% determine how many cross-correlations will be
    performed.
dist=zeros(numxcorr,1);
xc=0;
for st1num=1:length(loc)-1
    for st2num=1:length(loc)-st1num
        xc=xc+1;
        dist(xc)=lat_longd(loc(st1num,1),loc(st1num,2),loc(st1num+st2num,1),loc(st1num+st2num,2));
    end
end
% crossed=zeros(length(-lagtime:lagtime),numxcorr,days_in_month(mn));
for day=1:days_in_month(mn)
    filename=STATION_DATA\Cross_50_15s',num2str(day),'.mat';
    data=load(filename);
crossed(:,day)=data.bpcross;
clear filename data
toc
end
tic
stack=zeros(length(-lagtime:lagtime),numxcorr);
for xc=1:numxcorr
    A=crossed(:,xc,:);
    B=reshape(A,length(-lagtime:lagtime),days_in_month(mn));
    stack(:,xc)=sum(B,2);
end
figure;
wig(dist,1:lagtime,stack(5002:end,:),''black''); hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Lag Time (s)','FontWeight','bold','Color','black');
title('EGF for January 2009 Bandpassed Between 50s and 15s','FontWeight','bold','Color','black');
axis tight
toc
figure;
wig(dist,1:250,(stack(5002:5251,:)),''black''); hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Lag Time (s)','FontWeight','bold','Color','black');
title('EGF for January 2009 Bandpassed Between 50s and 15s','FontWeight','bold','Color','black');
axis tight
figure;
wig(dist,-lagtime:lagtime,stack,''black''); hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Lag Time (s)','FontWeight','bold','Color','black');
title('EGF for January 2009 Bandpassed Between 50s and 15s','FontWeight','bold','Color','black');
axis tight

%Calculate the signal to noise ratio for each cross-correlation

sig2noi_a=zeros(xc,1);
for n=1:xc
    sig2noi_a(n)=snr(stack(:,n));
end

% kill the xcorrs with signal to noise ratios less than 5

killcorr=find((sig2noi_a)<=-20);
%delete the corresponding columns
stacksnr=stack;
stacksnr(:,killcorr)=[ ];
distsnr=dist;
distsnr(killcorr)=[ ];
figure;
wig(distsnr,1:250,(stacksnr(5002:5251,:)),''black''); hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Lag Time (s)','FontWeight','bold','Color','black');
title('EGF for January, 2009 Bandpassed Between 50s and 15s','FontWeight','bold','Color','black');
axis tight
bins=[50:5:505];
for pp = 1:length(bins)-1
    bin=find(dist>=bins(pp) & dist<=bins(pp+1));
    check=isempty(bin);
    if check==1
        display(['Empty bin number number=', num2str(pp)]);
    else
        for bb=1:length(bin)
            ss=stack(:,bin(bb));
            end
        stackbin(:,pp)=sum(ss,2)/length(bin);
        end
    end
newdist=[52.5:5:502.5];
figure;
wig(newdist,10:lagtime,stackbin(((length(stackbin)-1)/2)+11:end,:),'black');
hold on
xlabel('Offset (km)','FontWeight','bold','Color','black');
ylabel('Lag Time (s)','FontWeight','bold','Color','black');
title('Receiver Responses in Acceleration','FontWeight','bold','Color','black');
axis tight
SIGNATURE PAGE

This is to certify that the thesis prepared by Natalia Krzywosz entitled “Investigation of Ambient Seismic Noise Using Seismic Interferometry in Western Montana” has been examined and approved for acceptance by the Department of Geophysical Engineering, Montana Tech of The University of Montana, on this 15th day of January, 2016.

Khalid Miah, Ph.D., Assistant Professor
Department of Geophysical Engineering
Chair, Examination Committee

Marvin Speece, Ph.D., Professor and Chair
Department of Geophysical Engineering
Member, Examination Committee

Michael Stickney, Geologist and Director, Earthquake Studies
Montana Bureau of Mines and Geology, Research
Member, Examination Committee