TOWARDS AN IMPROVED DAS WORKFLOW FOR GEOTHERMAL RESOURCE DEVELOPMENT

A Dissertation

by

MILAN BRANKOVIC

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Mark Everett
Committee Members,	Richard Gibson
	Eduardo Gildin
	Hiroko Kitajima
Head of Department,	Julie Newman

December 2022

Major Subject: Geophysics

ABSTRACT

I have created a novel seismic data processing method and an algorithm that can drastically reduce computation times of finite difference methods (FDM) applied to acoustic wave equations. The former is a data decomposition method designed specifically so that its output can efficiently and accurately describe seismic signals. The method, referred to as shifted-matrix decomposition (SMD), was used to reduce the memory requirements of seismic data, improve signal-to-noise ratio (SNR), and detect seismic events. For compression and denoising, SMD was tested on marine seismic gathers, which contained a large number of reflected waves and noise with high coherence that resembled seismic signals. Shifted-matrix decomposition reduced the memory requirement by 80% and improved the visibility of weak reflections that were obscured by noise. For event detection, SMD was applied to detect microseismic events from distributed acoustic sensing (DAS) recordings during fracture stimulation at a geothermal experimental site.

Regarding acoustic wave equations, I developed an algorithm that can be applied to standard finite difference methods to decrease the computational cost of forward modeling. An important feature of the algorithm is the calculation, at each time step, of the pressure in only a moving subdomain which contains the grid-points across which waves are propagating. The computation is skipped on grid-points at which the waves are negligibly small or non-existent. The novelty in this work comes from flexibility of the subdomain, namely its ability to closely follow the developing wavefield. When applied to a standard 2D finite difference scheme it reduced the computation time for wave propagation simulations by over 50% while maintaining low errors.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Everett, and my committee members, Dr. Gibson, Dr. Gildin, and Dr. Kitajima for their wonderful guidance.

Thanks also to my wife for her patience, love and watching over the kids while I worked on my dissertation.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by Dr. Everett, Dr. Gibson, and Dr. Kitajima of the Department of Geology & Geophysics and by Dr. Gildin of the Department of Petroleum Engineering.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University and funding from Crisman Berg-Hughes.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

Geophysical exploration and fracture monitoring play important roles in hydrocarbon production. Currently fracture monitoring is mainly used to optimize production from unconventional reservoirs. The quantity of hydrocarbons produced depends on the distribution, density, and connectedness of the fractures, natural or induced by hydraulic stimulation, that are present in the reservoir surrounding the wellbore [1]. Therefore, fracture monitoring plays an integral role in reservoir stimulation operations and can be used to evaluate the production potential.

As the global industry shifts towards renewable energy, geophysical exploration maintains importance due to its applicability for critical mineral prospection and as a means to characterize geothermal energy sources. Geothermal energy differs from other sources of renewable energy such as solar [2] and wind [3] because of its ability to provide a consistent power output, independent of weather conditions [4]. Thus, adding a geothermal power source can bring consistency to a power grid that is reliant on renewables.

Fracture identification and monitoring may be performed by analyzing microseismic data [5]. The work in this dissertation is largely motivated by a recent development in reservoir monitoring which involves the use of fiber optic cables to record both low frequency strain and seismic waves in a technique called distributed acoustic sensing (DAS)[6]. DAS provides an unprecedented view of the reservoir via dense spatial sampling of cable strain at high cadence. High recording rates at numerous receiver positions greatly increase the amount of registered microseismic activity compared to conventional geophone monitoring.

While DAS is well-suited for monitoring unconventional reservoirs, recently it has seen increasing use outside geophysical exploration. Distributed acoustic sensing has been coupled to existing subsea cables to monitor ground motion signals from seismic events and to identify fault zones [7]. Because of its high spatial and temporal resolution, DAS is expected to see further use in earthquake monitoring, imaging of faults and other geologic structures, and natural hazard assessment [8]. Finally, DAS is also being used in the development of enhanced geothermal systems (EGS) [9].

Because of its ability to record the strain rate at a large number of locations at a high rate, and its ability to convert pre-existing cables into seismic antennas, DAS is expected to drastically increase the amount of seismic data recorded and the number of seismic events detected. To support these developments, the work in this thesis is oriented towards the development of efficient methods for data processing, seismic wave simulation, and event detection.

Recording data with DAS requires large amounts of computer memory. Furthermore, the recorded datasets often contain waves from weak seismic events that are obscured by background noise. Thus, data processing methods that compress the data and reduce its noise would complement the new instrumental developments in seismic monitoring. For that reason, in Chapter II I developed a data decomposition method designed specifically for seismic data collected by a large number of linearly distributed receivers. I refer to the method as shifted-matrix decomposition, or SMD. The output of SMD is a series of vectors that encode seismic information. In Chapter II I argue that this output can be used to reconstruct a denoised version of the original data and to reduce memory requirements of the data. In Chapter III, I show how the output of SMD can also be used for event detection. In Chapter IV of the dissertation, I develop a method for decreasing the computation time of finite difference simulations of acoustic wave propagation. In seismic monitoring, seismic simulations are used to estimate the location of the seismic sources, and sometimes additionally the source mechanism.

The novel aspects described in this dissertation are meant to improve different stages of the workflow for seismic monitoring of the subsurface. The main objective of the herein-developed methods is the reduction of computational cost. While geothermal energy sources have many attractive properties, the cost of EGS development can make it difficult to obtain investments necessary for initiating geothermal energy production. Distributed acoustic sensing, which is a potentially less-expensive alternative to downhole geophones, can reduce the cost and risk of initial investment in EGS. By building computationally inexpensive algorithms that support the integration of DAS I strive to make the development of EGS more affordable, and therefore, more widespread.

Chapters II and IV from this dissertation have already been published in [11] and [12], respectively.

CHAPTER II

A MACHINE LEARNING-BASED SEISMIC DATA COMPRESSION AND INTERPRETATION USING A NOVEL SHIFTED-MATRIC DECOMPOSITION ALGORITHM

Introduction

The last decade has seen a great increase in hydrocarbon production from unconventional reservoirs. However, there are still many challenges in predicting their production potential. The quantity of hydrocarbons produced depends on the distribution and quantity of fractures present in the reservoir. Fracture identification and monitoring can be done by analyzing microseismic data [1]. With the goal of improving the ability to track the fracture distribution, the amount of seismic data acquired during reservoir monitoring has been increasing.

Recent developments in reservoir monitoring use fiber optic cables to record both low frequency strain and seismic waves in a technique called distributed acoustic sensing (DAS) [2]. DAS provides a new and unique view of the reservoir by recording strain rate data with a high sample rate and dense spatial sampling. Higher recording rates and more receiver positions increase the amount of microseismic activity recorded while monitoring the reservoir. Recording microseismic events with DAS significantly increases the memory requirement of the monitoring data. Thus, data processing methods that can compress the data and reduce its noise would complement the new instrumental developments in reservoir monitoring.

Signal processing algorithms are often based on the transformation of signal into a new domain. Early examples of compression involve discrete cosine transform [3] and wavelet transforms [4–6] which use cosine functions or wavelets to represent the data in order to reduce the memory requirements. Reduced-rank methods, which approximate the noiseless seismic data using low rank matrices and tensors, have been used in noise reduction [7] while also reconstructing missing data [8,9]. In recent years, dictionary learning has seen wide application in seismic data. Because of its ability to provide a compact and informative representation of seismic signals, it has been used for both noise reduction [10–12] and data compression [13,14]. Unfortunately, the drawback of the dictionary learning applications is the computation time that comes with learning and updating the dictionary. There have been successful efforts to reduce the associated computational times [15], but the computational cost of applying dictionary learning to microseismic DAS recordings is still too great. Thus far, dictionary learning methods have been applied to data collected by geophones, which can be very large but are still much smaller than data obtained by a single fiber-optic cable which can sample strain thousands of times a second on hundreds, even thousands of receiver locations. Since DAS is used for microseismic monitoring, the great amounts of data would need to be processed in real time, which puts a constraint on the computation time of processing methods.

To achieve computationally efficient compression and denoising, we created a new data decomposition method by improving and further developing ideas developed as a part of the local SVD [16]. Similarly to the previously mentioned reduced-rank methods, local SVD applies SVD to a window in seismic data and represents the data using a small number of singular vectors. What makes local SVD unique is the process of shifting the columns of the window to maximize their correlation prior to applying SVD. This allows singular vectors to capture the signal in seismic data with high accuracy, while ignoring most of the noise. Once the data in the window is processed with SVD, the window is moved to the next location. This process is repeated until column shifting and SVD have been applied to every part of the data matrix. The path of the moving window as well as the number of singular vectors used at each location are predetermined. Local SVD can enhance a seismic data set even if it contains multiple wave arrivals. However, local SVD struggles to capture seismic signals if waves with different dips are interfering or present in the same window. The “dip” of the wave refers to the slope of the wave in the matrix, or how much the row position of a wave changes as we move from one column to its adjacent column.

Some of the problems encountered with local SVD were resolved with the development of structure-oriented singular value decomposition (SOSVD) [17]. By using plane wave destruction [18], SOSVD can identify several dominant slopes at each window location. While using plane wave destruction provides a noticeable improvement, numerical artifacts can still appear at the intersection of the waves with different slopes.

Our method is inspired by the SOSVD and local SVD, and it also shifts the columns of the matrix before applying SVD. However, in order to avoid numerical artifacts, we use different processes to determine how columns should be shifted. Specifically, we do not use a moving window with a predetermined path. Instead, we use a geometric mean filter that adaptively chooses which elements to use in the geometric mean. We apply the geometric mean filter to seismic data in order to highlight areas which contain wave arrivals. The windows from the matrix to which we apply SMD depend on the results of the geometric mean filter. This allows us to use more singular vectors to describe areas with multiple wave arrivals, and fewer singular vectors to describe areas dominated by noise.

Additionally, local SVD and SOSVD use SVD results solely to denoise seismic data. They use them to reconstruct denoised version of seismic data as soon as they obtain them and do not discuss the compression achieved by storing SVD results. In our work, instead of reconstructing the data immediately, we store the SVD results. The final product of our algorithm is the collection of SVD results, that can later be reconstructed into the denoised version of original data. By doing this, our algorithm can be used for data compression as well as noise suppression. We call our new method the shifted-matrix decomposition, or SMD for short.

It should also be noted that DAS has seen increasing use outside geophysical exploration. Distributed acoustic sensing has been used to record signals from earthquakes and volcanic events [19]. Because of its unprecedented spatial and temporal resolutions, DAS is expected to see increasing use in earthquake monitoring, imaging of faults and many other geologic formations, and hazard assessment [20]. The growing potential of DAS application outside of geophysical exploration, adds importance to our method, which we believe will play an integral role in the processing of DAS data.

We organize the paper as follows: First we give an overview of singular value decomposition and present two simple examples that show advantages and drawbacks of its application to seismic data. Next, we demonstrate the improvements achieved by shifting the traces before extracting singular vectors. In the following subsection, we describe in detail each step of the SMD algorithm. The SMD algorithm depends on several parameters. The optimal values of said parameters are determined in a machine learning stage following the algorithm description subsection. In the training stage we use seismic field data obtained from marine seismic gathers, which has a large amount of interference between coherent waves, and noise which can be difficult to differentiate from signal. After training on marine seismic gathers, SMD provides accurate results on other seismic data as well as marine seismic gathers, which allows us to skip the training stage in future applications. To confirm the accuracy of SMD, we reproduce synthetic data from [17], and compare results of SMD to results of local SVD and SOSVD. The SMD is then tested on real seismic data. While SMD is primarily developed for application to microseismic data recorded by DAS, we currently do not have access to such data. Instead, we apply SMD to field data obtained from marine seismic gathers [21]. The results of applying SMD to field data are used to reconstruct a denoised version of the data as well as to estimate the elastic wave velocity. Finally, we discuss possible future applications of SMD and how its results could be used in signal detection during seismic monitoring.

Materials and Methods

Singular Value Decomposition

Consider seismic data stored in a matrix $\boldsymbol{M} \in \mathbb{R}^{n_t, n_r}$, where n_r is the number of the receivers recording the data and n_t is the number of time samples. An element of the matrix $M_{i,j}$ describes the ground motion at the j-th receiver, and at the i-th time step. Singular value decomposition method is a common matrix decomposition method that can be used to express the matrix $\boldsymbol{M}$ of rank r as the product of matrices:

$$M = \boldsymbol{U}_r \boldsymbol{D}_r {\boldsymbol{V}_r}^T \tag{1.1}$$

where $\boldsymbol{U}_r = [\boldsymbol{u}^1, \boldsymbol{u}^2, \dots, \boldsymbol{u}^r]$ contains the r left singular vectors as columns, the diagonal matrix $\boldsymbol{D}_r = diag(\lambda^1, \lambda^2, \dots, \lambda^r)$ contains the r singular values, and $\boldsymbol{V}_r = [\boldsymbol{v}^1, \boldsymbol{v}^2, \dots, \boldsymbol{v}^r]$ contains the r right singular vectors as columns. In traditional SVD, the left singular vectors and right singular vectors are normalized. However, multiplying the right singular vector by the singular value, allows us to store the singular value in the right singular vector, which slightly reduces the memory requirements of SMD results. For that reason, from this point on, the “right singular vector” refers to the normalized right singular vector multiplied by the singular value. Equivalent to Equation (1.1), SVD can also be used to express the matrix $\boldsymbol{M}$ as a sum of outer products of left singular vectors and right singular vectors weighted by singular values [7]. However, due to the way we define the right singular vector in this work, we can express is as just a sum of outer products of left singular vectors and right singular vectors:

$$\boldsymbol{M} = \sum_{k=1}^{r} \boldsymbol{u}^k \boldsymbol{v}^{k^T}. \tag{1.2}$$

While Equation (1.2) may be less common, regarding SVD as sum of outer products is helpful for understanding the processes of SMD. In this work, we refer to any column from $\boldsymbol{U}$ as a column vector, and any column from $\boldsymbol{V}$ as a row vector. This is because in the outer product $\boldsymbol{u}^k \boldsymbol{v}^{k^T}$ all columns are multiples of the column vector $\boldsymbol{u}^k$, and all rows are multiples of the row vector $\boldsymbol{v}^{k^T}$.

Singular value decomposition can also be used as a data compression method. We assume singular values in $\boldsymbol{D}_r$ decrease from first to last row. When applying SVD to seismic data, the initial singular values are much greater than the average singular value in $\boldsymbol{D}_r$ and describe most of the signal. By neglecting the small singular values, the singular value decomposition can be used to approximate matrix $\boldsymbol{M}$ as

$$\boldsymbol{M} \approx \sum_{k=1}^{r_s} \boldsymbol{u}^k \boldsymbol{v}^{k^T}. \tag{1.3}$$

where ($r_s << r$). By using only the first few singular vectors to describe the matrix $\boldsymbol{M}$ we can significantly decrease the memory requirement of the seismic data. In some cases, a few pairs of singular vectors can very accurately describe the data stored in $\boldsymbol{M}$. For example, if a wave packet arrives at all receivers at the same time, all columns in $\boldsymbol{M}$ will contain the same pattern, and in that case, we can represent $\boldsymbol{M}$ using a single outer product (see Figure 1.1).

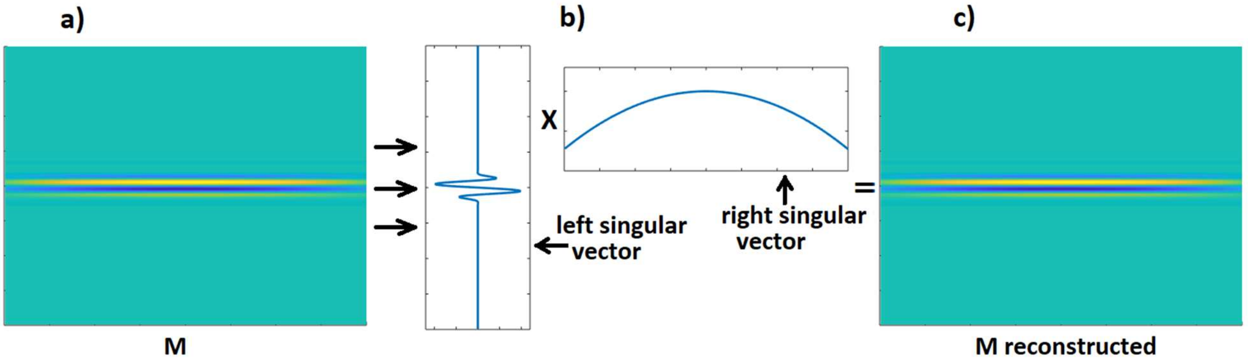

Figure 1.1. (**a**) A matrix showing a single wave arriving to all receivers at the same time. (**b**) A pair of singular vectors obtained by applying SVD to the matrix. (**c**) The outer product of the two singular vectors which matches the original matrix.

However, if the wave packet arrives at the receivers at different times, as in Figure 1.2, we can no longer represent the matrix ***M*** using a single outer product. Figure 1.2 shows an example of the errors that will occur if reconstruction of the data from a single outer product is attempted. Though in practice this would never be done, the example does show that greater data compression can be achieved in this framework if signals in data are aligned prior to a decomposition.

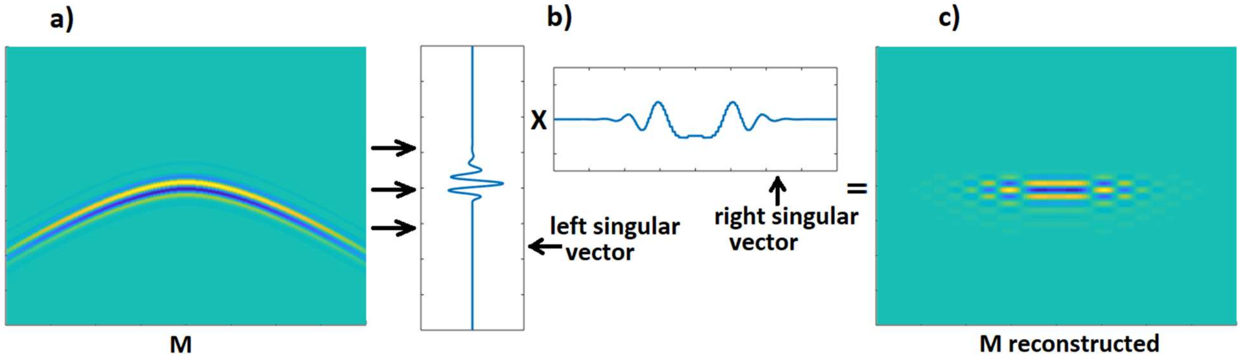

Figure 1.2. (**a**) A matrix showing a wave arriving to the receivers at different times (**b**) A pair of singular vectors obtained by applying SVD to the matrix. (**c**) The outer product of the pair of singular vectors which is very different from the original matrix.

The issues encountered with regular SVD (or PCA) were first addressed with the development of a local SVD by Bekara and Van der Baan [16], and later improvements led to the development of SOSVD by Gan et al. [17]. In both of these methods SVD is applied to a window $\boldsymbol{M}_w$ from matrix ***M*** with columns shifted in such a way to maximize the correlation between columns. The shape of the window doesn't change as the columns are shifted. This can be achieved in several ways, one of which is simply applying a wrap around condition so that values off the end of a matrix column, for example, are inserted at its beginning (and vice versa). An example of window shifting is described in Figure 3.

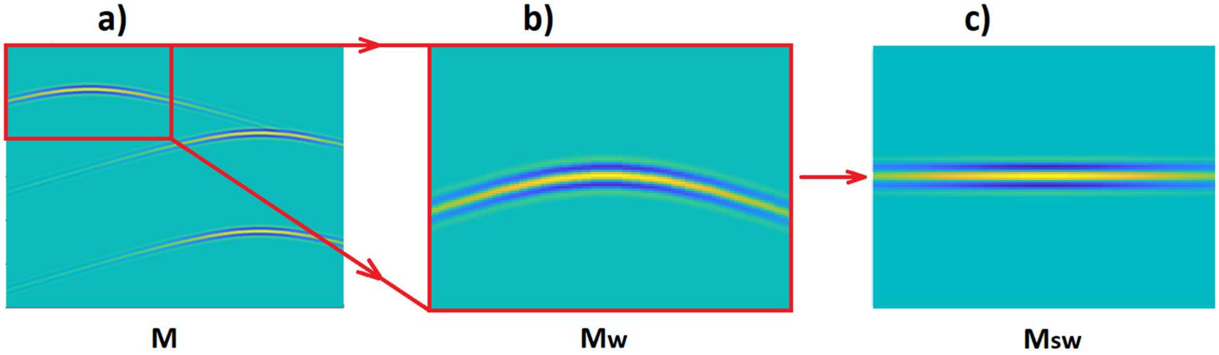

Figure 1.3. (**a**) The original data matrix ***M***. (**b**) A selected window $\boldsymbol{M}_w$ from the original data matrix ***M***. (**c**) The matrix $\boldsymbol{M}_{sw}$, obtained by shifting the columns of $\boldsymbol{M}_w$ to maximize their correlation.

The resulting shifted window ($\boldsymbol{M}_{sw}$) resembles the matrix from Figure 1.1, and can be described by the equation:

$$\boldsymbol{M}_{sw} = \chi(\boldsymbol{M}_w, \boldsymbol{s}) \tag{1.4}$$

where χ is the "shift" operator, $\boldsymbol{M}_w$ is the window subset from ***M*** and s is the shift vector. The "shift" operator takes a matrix and a shift vector as its arguments and shifts the columns of the matrix based on the values prescribed by the shift vector. For example, if the value of s for column *j* is *n*, then we would replace an element $M_{i,j}$ with the element $M_{i+n,j}$. The matrix decomposition used in local SVD and SOSVD can also be described with the Equation (1.5):

$$\boldsymbol{M}_w \approx \sum_k \chi(\boldsymbol{u}^k \boldsymbol{v}^{k^T}, -\boldsymbol{s}^k) \tag{1.5}$$

where $\boldsymbol{s}^k$ is the shift vector corresponding to the *k*-th pair of left and right singular vectors $\boldsymbol{u}^k$ and $\boldsymbol{v}^k$. By using Equation (1.5) and plugging in ***M*** for $\boldsymbol{M}_w$, the matrix ***M*** from the previous example can be presented with a single outer product coupled with a shift vector as shown in Figure 1.4. A simple quantitative example that shows the effectiveness of column-shifting with a wave similar to the one from Figure 1.4 can be found in the Appendix A.

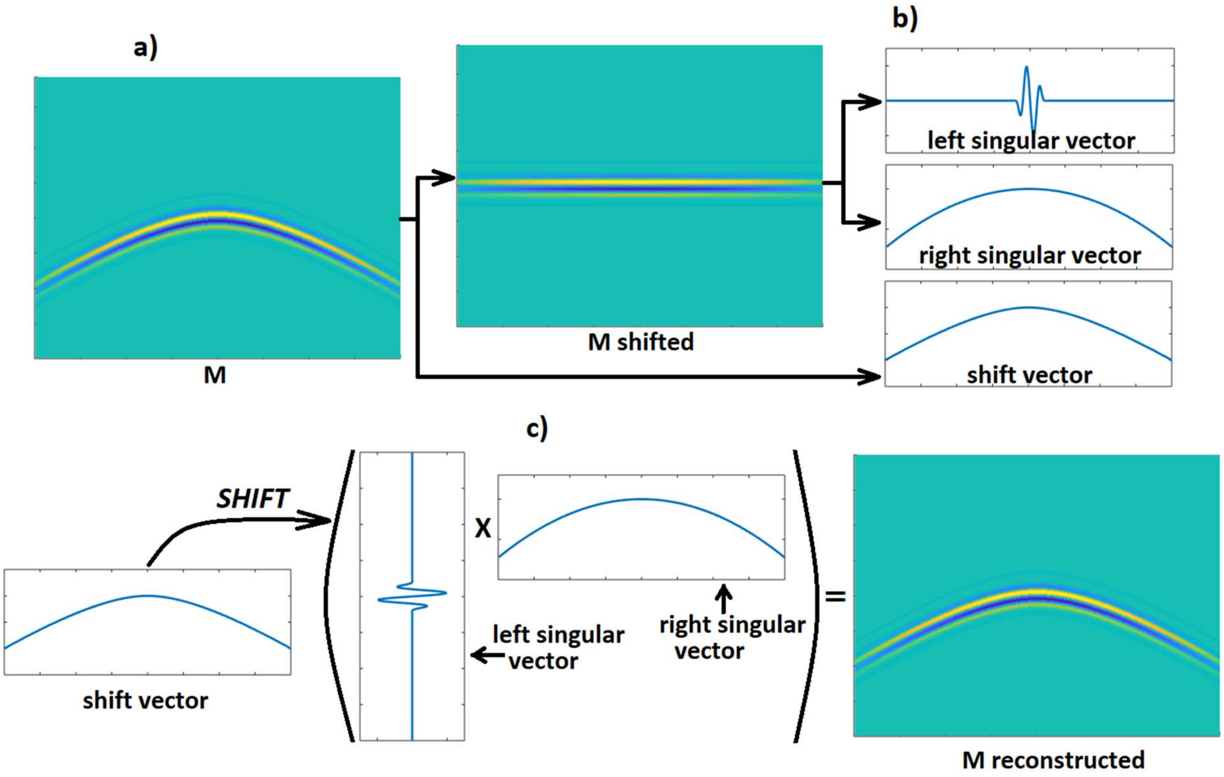

Figure 1.4. (**a**) A matrix showing a wave arriving to the receivers at different times. (**b**) The matrix being expressed as a pair of singular vectors and a shift vector. (**c**) The singular vectors and the shift vector being used to reconstruct a matrix that matches the original.

Unfortunately, both local SVD and SOSVD seem to fail to accurately denoise seismic data which contains interfering waves with different dips. In such scenarios, numerical artifacts appear around the area of intersection. We solve this problem by developing shifted-matrix decomposition (SMD) which uses a very different process for determining the shift vector. The following subsection will give a detailed description of the processes SMD uses to obtain the shift vectors and the pairs of singular vectors.

SMD Algorithm

The SMD algorithm can be described by the following steps:

1. Using a geometric mean filter, choose a specific point (row number and column number) in the data matrix $\boldsymbol{M}$ at which a displacement (or pressure) from a coherent wave was most likely recorded.

2. Using cross-correlation, find this wave in as many surrounding columns as possible. For each column, the relative row positions of the said wave are recorded in the shift vector.
3. Shift the columns of ***M*** using the shift vector s and record the row vector and the column vector. Subtract the shifted outer product of row vector and column vector from matrix ***M*** and shift the columns of ***M*** back to their original positions.
4. Repeat steps 1–3 until a certain performance criterion is satisfied.

A visual representation of the algorithm can also be found in Figure 1.5.

To provide a more in-depth understanding of the algorithm, in the following paragraphs we will give a detailed description of each of the four steps.

Step 1

To identify a point in matrix ***M*** at which a displacement (or pressure) from a coherent wave is likely recorded, we start by applying a geometric mean filter to the data matrix ***M***, to obtain the matrix ***E*** (Equation (1.6)):

$$E_{i,j} = | \prod_{q=-n_E}^{n_E} M_{\left(i+\delta_{i,j}(q)\right),(j+q)} |^{\frac{1}{2n_E+1}}. \tag{1.6}$$

For a position *(i, j)*, the parameter n_E indicates that n_E columns left of the position *(i, j)* and n_E columns right of the position *(i, j)* will be used when calculating $E_{i,j}$. If a position *(i, j)* is close to the first or the final column, fewer elements are used to calculate the geometric mean *Ei,j* as to avoid stepping outside the matrix boundaries. The row positions of elements used in the geometric mean are also constrained in order to always remain within the matrix boundaries.

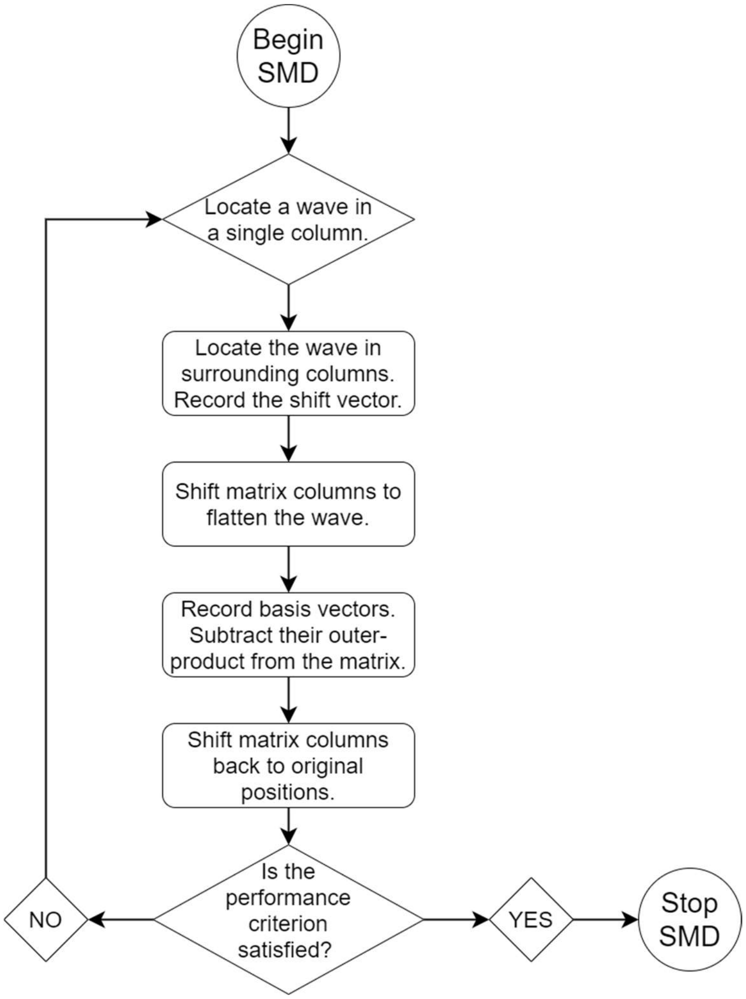
Begin SMD
Locate a wave in a single column.
Locate the wave in surrounding columns. Record the shift vector.
Shift matrix columns to flatten the wave.
Record basis vectors. Subtract their outer-product from the matrix.
Shift matrix columns back to original positions.
Is the performance criterion satisfied?
NO
YES
Stop SMD

Figure 1.5. A flowchart describing the SMD algorithm.

What makes this filter unique is the choice of elements which are used when calculating the geometric mean. From each of the $2n_E$ surrounding columns, only one element is used in the geometric mean. The variable $\boldsymbol{\delta}$ in Equation (1.6) indicates which element is used in the geometric mean from each column. For example, when calculating the value of the geometric mean in position *(i, j)*, from column $j+q$ we take the element at row position $i+\delta_{i,j}(q)$ to be used in the geometric mean. The process for determining the said set of elements is described with Figure 1.6.

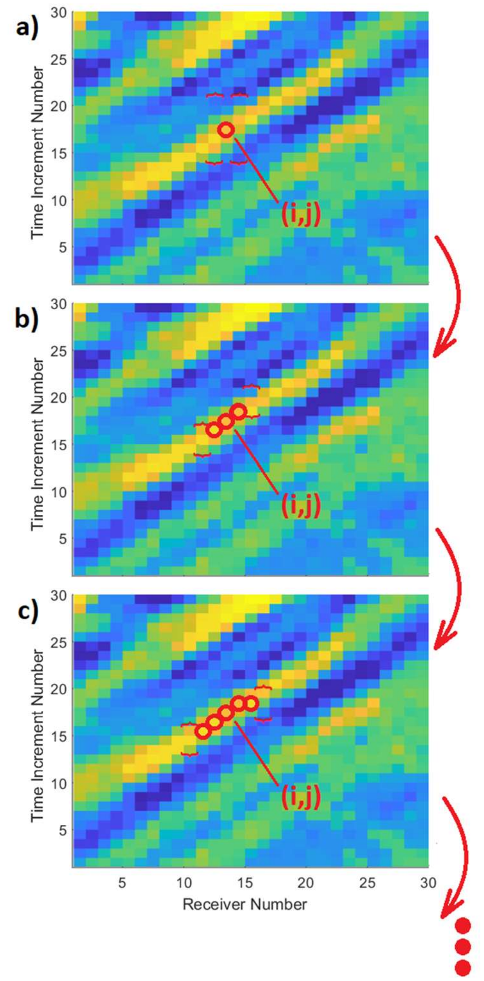

Figure 1.6. The process of determining the set of elements *($\boldsymbol{\delta}_{i,j}$)* for calculating the geometric mean at the position *(i, j)*. The curly braces indicate the set of elements from which the next element will be added to the geometric mean. (**a**) The first element, $M_{i,j}$ is automatically added to the set $\boldsymbol{\delta}_{i,j}$ and the algorithm searches for the following elements in columns adjacent to column *j*. (**b**) The two elements from the adjacent columns are added and the search is now on columns $j-2$ and $j+2$. (**c**) Two more elements are added to $\boldsymbol{\delta}_{i,j}$, from columns $j-2$ and $j+2$, and the search continues until $\boldsymbol{\delta}_{i,j}$ is complete.

Consider calculating the geometric mean in order to determine the value of $E_{i,j}$. Before calculating the geometric mean, we must determine the set of elements used in the mean (determine the values of $\boldsymbol{\delta}_{i,j}$). We start from the *j*-th column, from which we always use the element $M_{i,j}$ ($\delta_{i,j}(0) = 0$). Then, we proceed to find which elements to use from adjacent columns (Figure 1.6a). Assuming $M_{i,j}$ is positive, for column $j+1$, we pick the element between positions $(i-m, j+1)$ and $(i+m, j+1)$ which has the highest value. If the element $M_{i,j}$ were negative, we would pick the lowest value. The parameter m represents the maximum dip of the wave rounded up to the nearest integer. The same process is used to determine which element to use from column $j-1$. For any column $j+n_s$ where $1 < n_s \leq n_E$, for the geometric mean, we pick the element between positions $(i_s-1, j+n_s)$ and $(i_s+1, j+n_s)$ with the highest value (Figure 1.6b,c). The row number i_s is determined by following a linear trend based on the row position i in column j and the picked row position $i+\delta_{i,j}(n_s-1)$ in the column $j+n_s-1$. The same process is used to determine which element to use from columns below j − 1.

The geometric mean filter is designed to imitate the process humans use for recognizing coherent waves in the data. When differentiating signal from noise, the distribution of displacement (or pressure) along a long range of receivers plays an important role. A weak wave might have an amplitude that is even smaller than that of noise in few noisy areas. However, if the displacement (or pressure) from the wave is present in a long range of receivers, and its arrival time follows a hyperbolic trend, a data reviewer would certainly notice the wave. The goal of the geometric mean filter is to assign higher values to such a weak wave than to a noisy area with high amplitudes. By testing the geometric mean filter on a large number of examples, we found few cases in which the weak wave was not assigned higher values than some areas of the noise. However, even in such cases the slight increase in filter output due to the weak wave, followed a hyperbolic trend in time on a large range of receivers, while the higher increases due to randomness of noise did not. By applying the geometric mean filter again, to the results of the first application of the filter, we can bring out all noticeable waves, no matter how weak, to have higher values than any area containing only noise. As we have achieved the desired results with the second application of the filter, a third application is unnecessary and would only increase computation time. Therefore, once all the elements of the matrix ***E*** have been calculated, the filter is used again, but this time on the matrix ***E*** to obtain the matrix ***F***. When geometric mean filter is used to obtain ***F***, the parameter which controls the number of columns used in each geometric mean, n_F, is different than n_E which is used with the filter to obtain E. The optimal values of the parameters n_E and n_F are determined in a machine learning stage of the algorithm.

An example of the original data matrix ***M***, and of the filter's result ***F*** is given in the Figure 1.7.

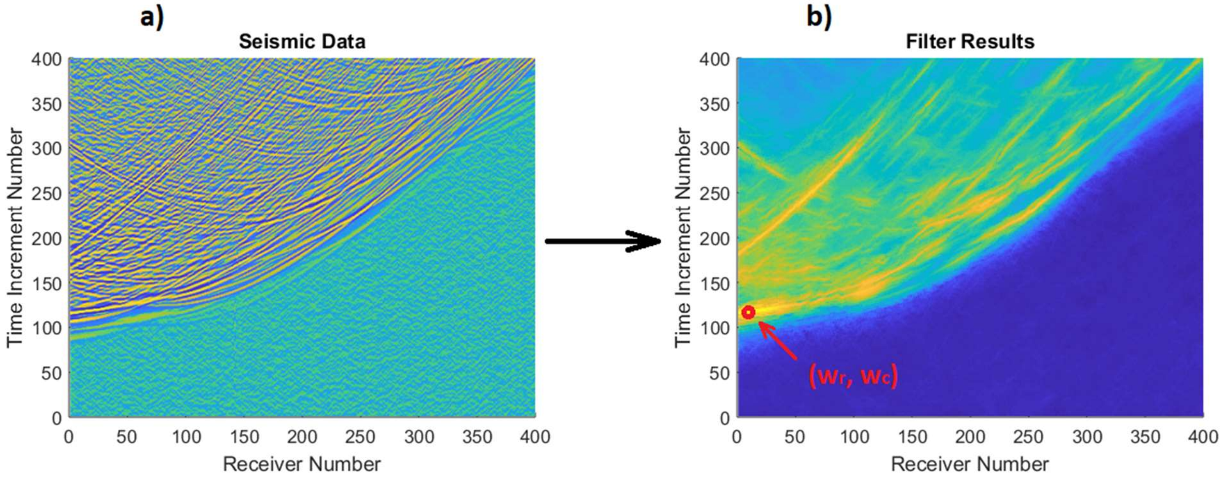

Figure 1.7. (**a**) The original data matrix ***M***. (**b**) The filter's result ***F*** with the indicated waveform row and waveform column.

Matrix ***F*** is designed to produce large values at points at which a displacement (or pressure) from a coherent wave was recorded. We select the element in ***F*** with the highest value as the most likely element in the matrix ***M*** that describes a coherent wave. The row and column of the selected point designates the position of the coherent wave and are termed respectively the waveform row and waveform column, or w_r and w_c in equations and Figure 1.7.

Step 2

In the first step we determine the position (w_r and w_c) of a specific point, or element, that describes a part of a coherent wave. Thus, we know the position of said wave in only one of the columns, namely the waveform column (w_c). In the second step, we find the row position of its waveform in the remaining columns of ***M***, and in doing so we build the shift vector. In this context, the shift vector records the row position of the waveform in each of the columns, relative to the position of the waveform row (w_r). Therefore, the waveform position in any column j is equal to the sum of the waveform row w_r and the j-th element of the shift vector $\boldsymbol{s}$. The location of the waveform in the remaining columns is found using a cross-correlation procedure.

We select a sequence ($\boldsymbol{\psi}$) from the waveform column that contains all the elements between rows w_r-w and w_r+w (Figure 1.8a,b). The parameter w represents the window size when determining the sequence ($\boldsymbol{\psi}$), and its value is determined in the machine learning stage. The sequence of elements $\boldsymbol{\psi}$ is supposed to represent the waveform, or the greater part of it, and we refer to it as the waveform sequence. The position of the waveform in other columns is then determined by finding in each column a sequence of elements that has the highest correlation with the waveform sequence (Figure 1.8c).

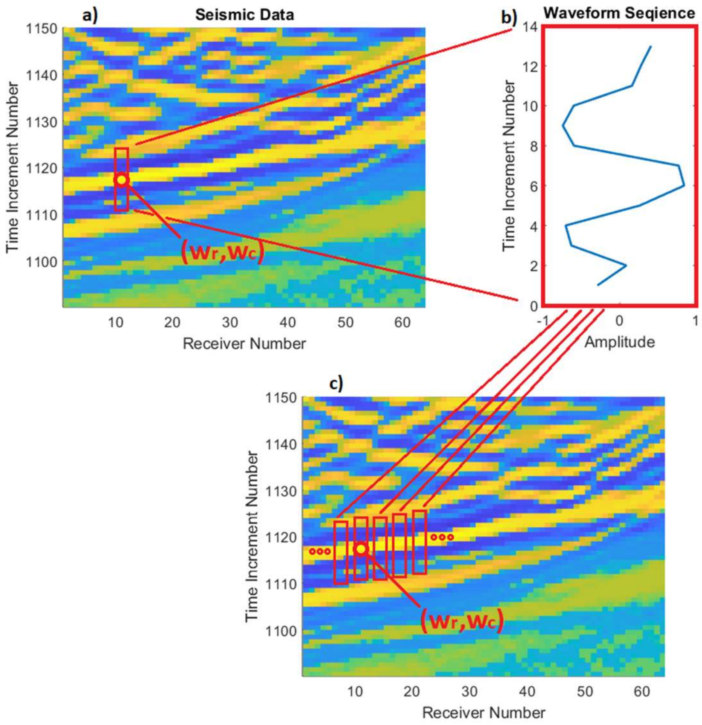

Figure 1.8. (**a**) Zoomed in area around position (w_r, w_c) from matrix ***M***. (**b**) The selected waveform sequence $\boldsymbol{\psi}$. (**c**) The location of $\boldsymbol{\psi}$ in the surrounding columns.

We start from the columns adjacent to the waveform column w_c. The waveform position in columns adjacent to w_c must be between w_r-m and w_r+m, where m is the maximum dip rounded up to the nearest integer. To reduce computation time, and to make sure to follow the same wave throughout all the columns, in adjacent columns we search for the $\boldsymbol{\psi}$ sequence from row w_r-m-w to row w_r+m+w. Initially, the range of rows in which we search for the $\boldsymbol{\psi}$ sequence is entirely dependent on the waveform position in the closest column. For any column position $j > w_c$, we search for the $\boldsymbol{\psi}$ sequence from row $w_r+s_{j-1}-m-w$ to row $w_r+s_{j-1}+m+w$.

Once we have determined the waveform position in columns ranging from position $wc-2l$ to position $wc+2l$, we narrow the search window for the following columns. The value of the parameter l is determined in the machine learning stage. At this point, for a column $j>w_c+2l$ we search for the waveform sequence from row i_e-1-w to i_e+1+w.

The row position i_e in column j is determined by following the parabolic trend of waveform positions in columns $j-1$, $j-1-l$, and $j-1-2l$. Fitting parabolas to shift vectors will also be used in the Discussion section, for estimating the velocities of the recorded waves. While arrival times are usually modeled with a hyperbola, we use a parabola because it requires fewer points on the shift vector to find its coefficients. Since most parts of the hyperbola can locally be fairly accurately described with a parabola, our parabolic estimates are sufficiently accurate. Narrowing the search window allows us to estimate the waveform position in each column more quickly and to make estimating waveform position more resilient to perturbations from noise and other waves. The optimal values for parameters w and l are determined in the machine learning stage. Moreover, if the maximum correlation with the waveform sequence in a certain column is below 0 the step terminates, as at that point we can be certain that the waveform is no longer present.

Step 3

Once the shift vector and all the waveform positions are determined, the shift vector and shift operator are used to shift the columns of the data matrix $\boldsymbol{M}$ so that the wave appears in the same set of rows in each column. At that point, the waveform is similar to the one shown in Figure 1.1, and it can be very accurately described as an outer product between two singular vectors. The number of rows storing the waveform pattern is much smaller than the total number of rows in $\boldsymbol{M}$. Therefore, most of the elements in the column vector are not describing the waveform, hence are not relevant, and can be ignored. The length of the column vector need not be as long as columns in $\boldsymbol{M}$, so it is reduced such that its first element specifies the number of zeros before the waveform and the last element specifies the number of zeros after the waveform. For example, the waveform row (w_r) would then be equal to the sum of the first element of the column vector (u_1) and half of the length of the recorded waveform (l_w). The optimal length of the waveform that is recorded in the column vector (l_w) (i.e., the number of elements in the column vector excluding the first and the last element), is determined in the machine learning stage. The row vectors and shift vectors are reduced in a similar manner to the column vector since the waveform is often contained in only a subset of columns of $\boldsymbol{M}$, i.e., the wave packet may not be recorded at a subset of receiver locations. The effect of these changes is to decrease the number of elements required to represent $\boldsymbol{M}$, thus decreasing the memory requirement of the SMD algorithm

Once the two singular vectors (the column vector $\boldsymbol{u}$ and the row vector $\boldsymbol{v}$) are extracted, their outer product is subtracted from the matrix $\boldsymbol{M}$. The columns of $\boldsymbol{M}$ are also shifted back to their original locations. The new matrix $\boldsymbol{M}_{new}$ can be described with the equation:

$$\boldsymbol{M}_{new} = M - \chi(\boldsymbol{u}\boldsymbol{v}^T, -\boldsymbol{s}). \tag{1.7}$$

As a result, the majority of the located wave disappears which allows the algorithm to focus on a potential second wave present in $\boldsymbol{M}$. At the end of this step of the algorithm, a wave has been subtracted from matrix $\boldsymbol{M}$ and added to the compressed data as a pair of singular vectors and a corresponding shift vector.

Step 4

The three steps described above are repeated until a certain performance criterion is met. Herein we define that criterion in terms of memory. Once the memory usage of the compressed data reaches 20% of the original data (80% compression), the SMD algorithm terminates. Specifically, once the number of elements used to describe all the shift vectors and singular vectors is equal to 20% of the number of elements in the matrix describing the raw seismic data, the performance criterion is met and the SMD algorithm terminates.

Machine Learning

Training Data and Scoring the Model

In the algorithm description we have introduced five parameters that influence the results of SMD (shifted-matrix decomposition). The five parameters are n_E and n_F from Step 1, w and l from Step 2, and l_w from Step 3. However, we do not have an intuitive understanding of how the results of SMD are affected by the changes to the parameters, and we have no analytical solutions for optimal values for the parameters. For that reason, we use supervised machine learning to find a set of parameters that provide consistently good SMD results. Once the parameters are optimized in the training stage, SMD can be applied to new seismic data without the need to run the training stage again. Moreover, the parameter m, which also affects the results of SMD, is predetermined and therefore cannot be optimized.

The first step is to define what is meant by a good SMD output. We desire the matrix reconstructed from SMD to closely resemble the original seismogram matrix $\boldsymbol{M}$. Specifically, with a predefined compression rate, we want to capture a big portion of the original data, and we want as little noise as possible to be recorded in the SMD results. We refer to the matrix reconstructed from decomposed data as $\boldsymbol{M}^r$. When we measured the resemblance between $\boldsymbol{M}$ and $\boldsymbol{M}^r$ as an L_1 or L_2 norm of the error $\boldsymbol{M}-\boldsymbol{M}^r$, we obtained the smallest error norms when the SMD captured a lot of the noise with a few pairs of singular vectors and shift vectors. For this reason, rather than minimizing a norm of the error, we decided to maximize the dot product of matrices $\boldsymbol{M}$ and $\boldsymbol{M}^r$:

$$\boldsymbol{M} \cdot \boldsymbol{M}^r = \sum_{i,j} M_{i,j} {m^r}_{i,j}. \tag{1.8}$$

The dot product was more sensitive to SMD results representing a weak coherent wave or a small remaining part of a large wave than it was to SMD results representing primarily noise. We believe this is because the dot product of the two matrices is proportional to their correlation. Recording a large area of noise with a pair of singular vectors and a shift vector could decrease the error $\boldsymbol{M}-\boldsymbol{M}^r$ by a fair amount. However, because of the randomness of noise, the reconstructed data will still have weak correlation with the original data in the noisy area. Recording any area with coherent waves with a pair of singular vectors and a shift vector usually provides high correlation between the original and the reconstructed data in said area.

Thus, we search for a set of parameters which maximize the dot product $\boldsymbol{M} \cdot \boldsymbol{M}^r$. For training data, we use recordings of reflected wave arrivals from 5 different shotgathers from marine seismic gathers. We use seismic field data obtained from marine seismic gathers because the data contain a large number of coherent waves with lots of interference, and noise with high coherence which can be difficult to differentiate from signal. To reduce the time spent in the training stage, from each shot-gather we only use 2 s recordings containing reflected waves and the preceding noise recorded by the first 200 receivers. More details on the field data will be provided in the following section. The five shot-gathers are labeled as $\boldsymbol{M}^{1} - \boldsymbol{M}^{5}$ and data reconstructed from decomposition of the five seismograms as $\boldsymbol{M}^{r,1} - \boldsymbol{M}^{r,5}$. Thus, the overall quality of SMD with a given set of parameters would be quantified as the SMD score (ζ):

$$\zeta = \sum_{k=1}^{5} \boldsymbol{M}^{k} \cdot \boldsymbol{M}^{r,k}. \tag{1.9}$$

The SMD score is only used in training stage to find an optimal set of parameters, and it will not be present in the following sections.

Derivative Free Optimization

There is no analytical formula directly relating the SMD score (Equation (1.9)) to the five input parameters. Therefore, we cannot use a gradient analysis to find optimal values for these parameters. Instead we use a derivative-free numerical optimization method. Specifically, we will use a pattern search. The five parameters to be optimized can only be natural numbers. Thus, the five parameters can be described by an element in $\mathbb{N}^5$. In this optimization, we take a subset $\boldsymbol{\Omega}$ of $\mathbb{N}^5$, to be the set of all the reasonable values of the five parameters. Specifically, we seek an element in $\boldsymbol{\Omega}$ for which SMD score has the highest value. This is done by performing a pattern search which takes the current position in $\boldsymbol{\Omega}$ and checks adjacent points to see if any of them yield a higher SMD score. An adjacent position is defined as a position that can be reached by changing only one of the five parameters by the minimum amount. If the adjacent element with the highest SMD score has a higher score than that of the current position, the optimizing position is moved to the adjacent element. This is done until a local maximum for the SMD score is found. To increase the likelihood of finding the global maximum, the pattern search is repeated multiple times with a random starting location.

Application

Once the training stage is finished, SMD can be applied to a new seismic data set even if it is not from a marine gather. Without going through the training stage again, SMD was successfully applied to new marine gathers, synthetic data from [17], and to field data collected by linearly distributed geophones. However, the discussions in the following sections are focused on data from marine seismic gathers, rather than data gathered by geophones. Because marine seismic gathers were obtained from a much larger number of receivers and contain greater volumes of data, they provide a more realistic example of applications of SMD.

Even when skipping the training stage, the parameters still need to be adjusted before SMD is applied to new seismic data, based on the dominant frequency and the maximum slope of the waves in the new dataset. Specifically, parameters w from Step 2 and l_w from Step 3 need to scale proportionally to the period of the dominant frequency multiplied by the sampling rate. Moreover, parameters n_E and n_F from Step 1 and l from Step 2 should scale proportionally to the period of the dominant frequency multiplied by the sampling rate and divided by the maximum slope of arriving waves. Therefore, when applying SMD to a seismic data set, one should also include information about the dominant frequency in the data set, the sampling rate of the receivers, and the maximum slope of the arriving waves. Knowing these values allows the user to apply SMD to new seismic data sets without going through the training stage. The flowchart in Figure 1.9 provides a visual description of how SMD is applied to a set of files containing seismic data.

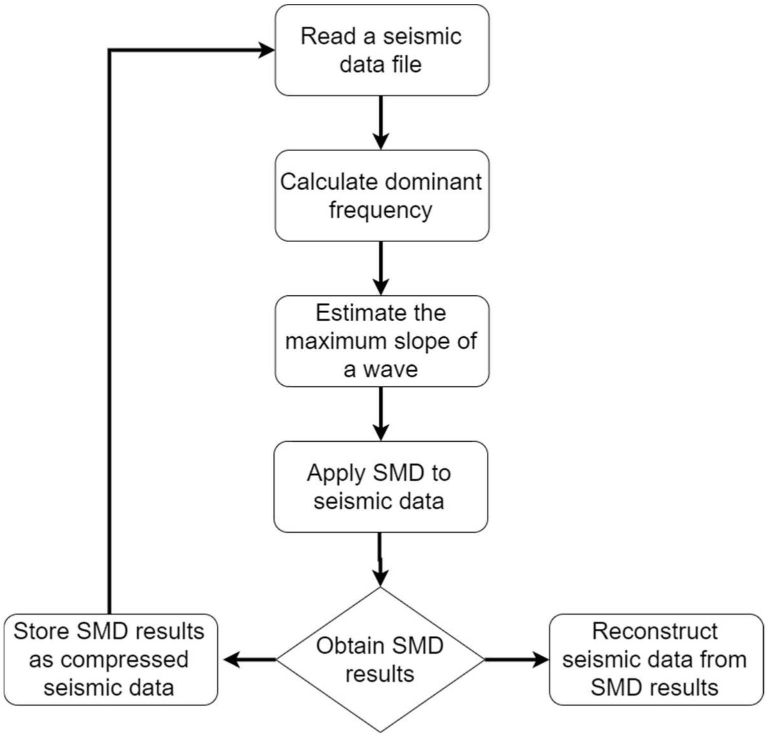
Read a seismic data file
Calculate dominant frequency
Estimate the maximum slope of a wave
Apply SMD to seismic data
Obtain SMD results
Store SMD results as compressed seismic data
Reconstruct seismic data from SMD results

Figure 1.9. Flowchart describing the process of applying SMD to seismic data. The fourth step from above is described by the flowchart presented in Figure 1.5.

Results

Once the SMD parameters have been optimized in the machine learning stage, its performance is tested on three datasets. The first one is a synthetic dataset similar to one from [17]. Structure-oriented SVD [17] was tested on several datasets and while it was successful at reducing noise in all examples, it produced numerical artifacts in one of them. In this work we reproduce that challenging dataset in order to demonstrate the improvements achieved with SMD. The other synthetic data test cases from [17] contain fewer coherent waves and don't show interference between waves of different slopes and are not as difficult for reliable compression and analysis. For that reason, the second and third data sets on which SMD is tested are from field data obtained during marine seismic gathers. The second dataset has many, interfering, strong arrivals which can together create a complicated signal. The third dataset has fewer arrivals, but the coherent waves are much weaker and difficult to differentiate from noise.

Synthetic Data

Figure 1.10 shows application of local SVD and SOSVD on the synthetic dataset from [17], while Figure 1.11 shows the application of SMD to a very similar data set. To create the synthetic data in Figure 1.11 we recreated the signal observed in Figure 1.10 and added random noise to it.

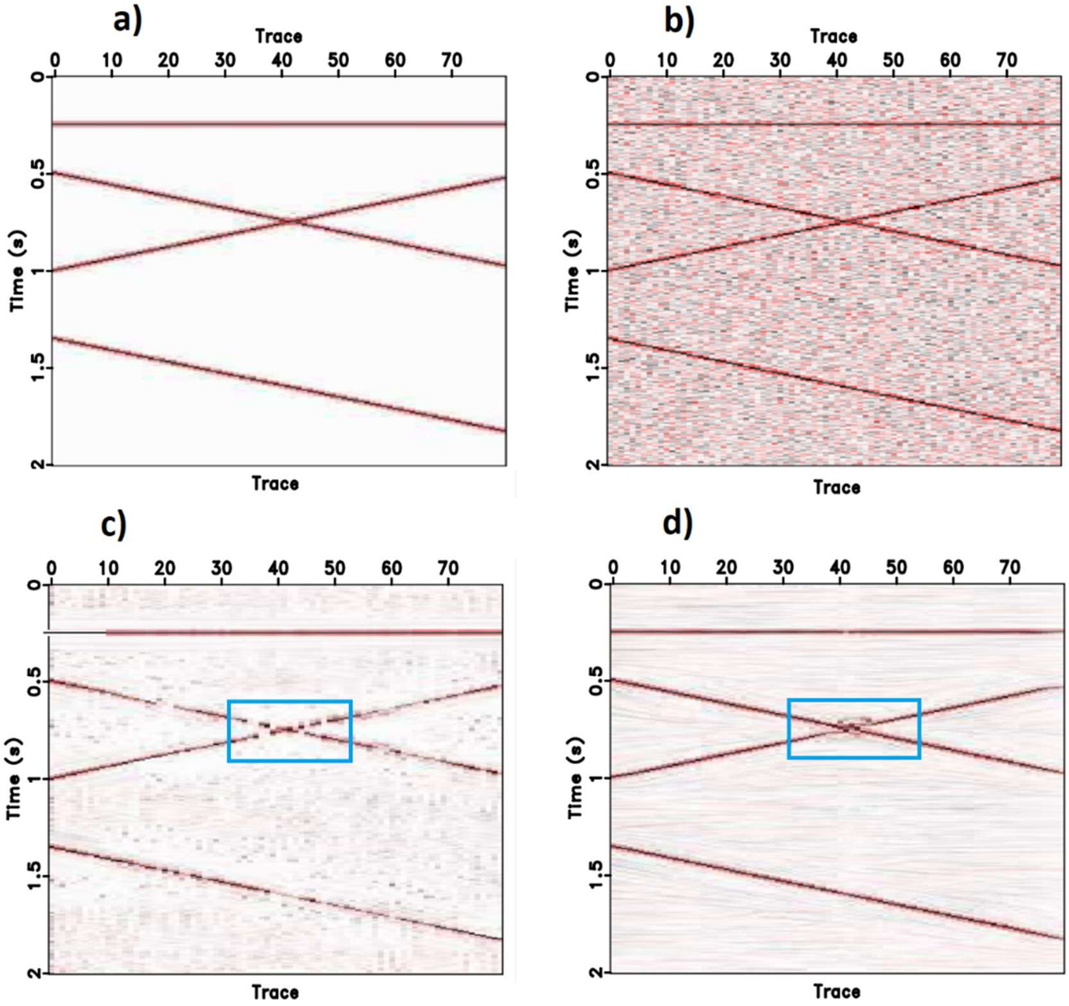

Figure 1.10. (**a**) Synthetic data showing several wave arrivals without noise. (**b**) The noisy data, created by adding noise to the synthetic data. (**c**) The noisy data filtered using local SVD. (**d**) The noisy data filtered with SOSVD. The blue rectangles highlight areas of interest in which local SVD and SOSVD produce numerical artifacts. This figure was modified from [17].

In Figure 1.10 we see that both local SVD and SOSVD are able to remove most of the noise. However, both local SVD and SOSVD struggle to properly reconstruct the signal in the area highlighted by the blue rectangle in Figure 1.10c,b. The highlighted area contains two waves of different slopes intersecting with each other.

Figure 1.11 shows results of applying SMD to the synthetic data set twice, with 95% compression (Figure 1.11c) and 80% compression (Figure 1.11d). In both SMD applications the coherent waves were reconstructed perfectly. We can see that there are no numerical artifacts in the area highlighted by the blue rectangle in Figure 1.11c,b, which contains intersecting waves with different slopes.

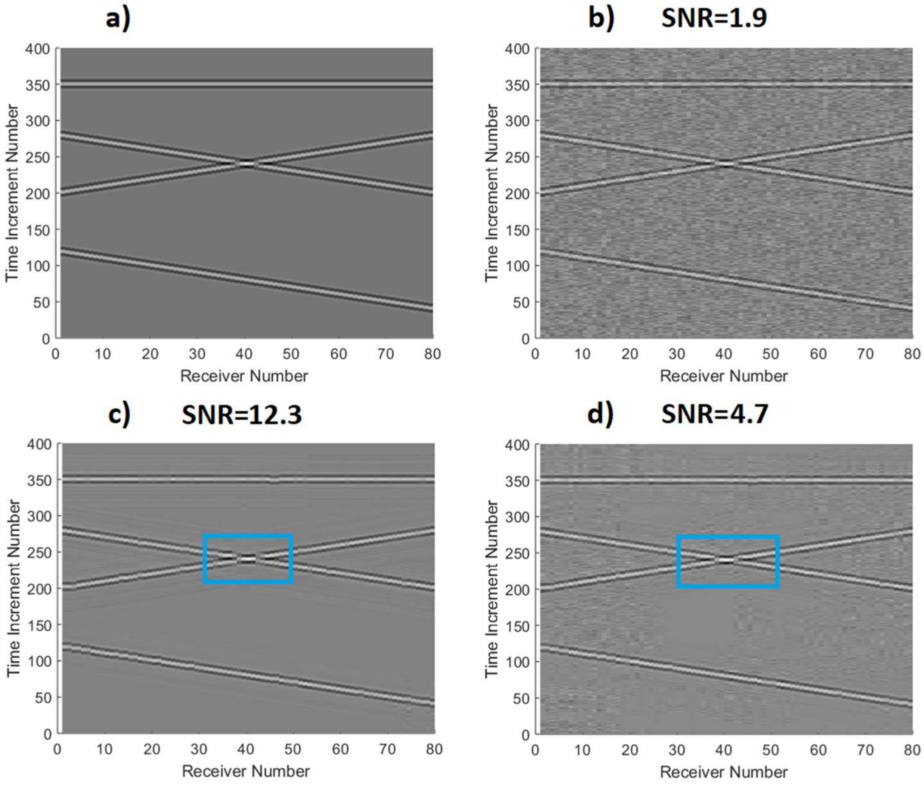

Figure 1.11. (**a**) Synthetic data showing several wave arrivals without noise. (**b**) The noisy data, created by adding noise to the synthetic data. (**c**) Data obtained by applying SMD to the noisy data with 95% compression. (**d**) Data obtained by applying SMD to the noisy data with 80% compression. The blue rectangles highlight areas of interest in which local SVD and SOSVD produce numerical artifacts.

While we can see significant noise reduction, we could not find specific information in [17] describing the amount of noise present in synthetic data in Figure 1.10. However, we can estimate the signal-to-noise ratio in the synthetic data on which SMD is tested. By measuring the root mean square value from all receivers in time increment range from 300 to 320, where only noise is present, we can estimate the amplitude of the noise. To estimate the amplitude of the signal, we measure the root mean square value from all receivers in time increment range from 340 to 360, from the data presented in Figure 1.11a, which contains only pure signal. The signal-to-noise (ρ) ratio is therefore calculated with the Equation (1.10):

$$\rho = \frac{\sqrt{\sum_{i=340}^{i=360} \sum_j {M_{i,j}^{p}}^2}}{\sqrt{\sum_{i=30}^{i=3} \ \sum_j {M_{i,j}^{n}}^2}} \tag{1.10}$$

where $\boldsymbol{M}^p$ represents the data containing only pure signal presented in Figure 1.11a, and $\boldsymbol{M}^n$ represents the data from Figure 1.11b,c, or d, which contains some amount of noise.

The signal-to-noise ratios in the original data, data reconstructed from SMD results after 80% compression, and after 95% compression were 1.9, 4.7, and 12.3, respectively. Notice that the signal-to-noise ratio is much larger when SMD is applied with 95% compression than when it is applied with 80% compression. This is because there are only a few coherent waves present in the data. These coherent waves are represented with first several singular vectors and shift vectors, while the following singular vectors and shift vectors are describing noise. However, SMD results represent noise a lot less efficiently than coherent waves, and the majority of the noise is still not recorded when SMD is applied with 80% compression. In conclusion, the SMD application with 80% reduction in memory requirements picked up some of the noise, while SMD application with 95% data compression ignored the noise almost completely.

Field Data

Finally, the SMD algorithm is tested on field data, collected between January and March 2016 during the CREST expedition, MGL1601, aboard the R/V Marcus G. Langseth. Pressure waves are generated by a tuned array of 36 air guns, towed at a depth of 6 m. The resulting acoustic waves are recorded using a 12,587.5 m hydrophone streamer, towed at the depth of 8 to 12 m, and carrying 1008 receivers. The receivers are spaced by 12.5 m, each of them recording pressure once every 4 ms. The maximum offset between two adjacent columns is a little greater than two rows, so we round it up to three rows. Further information regarding data acquisition can be found in [21]. Seismic data are also available at the NSF-sponsored Academic Seismic Portal hosted by the University of Texas Institute for Geophysics and can be accessed at https://www.marine-geo.org/tools/search/Files. php?data_set_uid=23597 (retrieved on 31 March 2021).

Unfortunately, when being applied to real seismic data such as in Figures 1.12 and 1.13 SMD with 95% compression cannot reconstruct the arriving waves properly. Due to multiple reflections and dispersion, the number of coherent waves is much larger in marine seismic gathers than in synthetic data. Furthermore, the noise has high coherence such that it can closely resemble wave arrivals. Using SMD with 80% compression ensures that the arriving waves will be properly reconstructed while also ensuring the noise is drastically reduced.

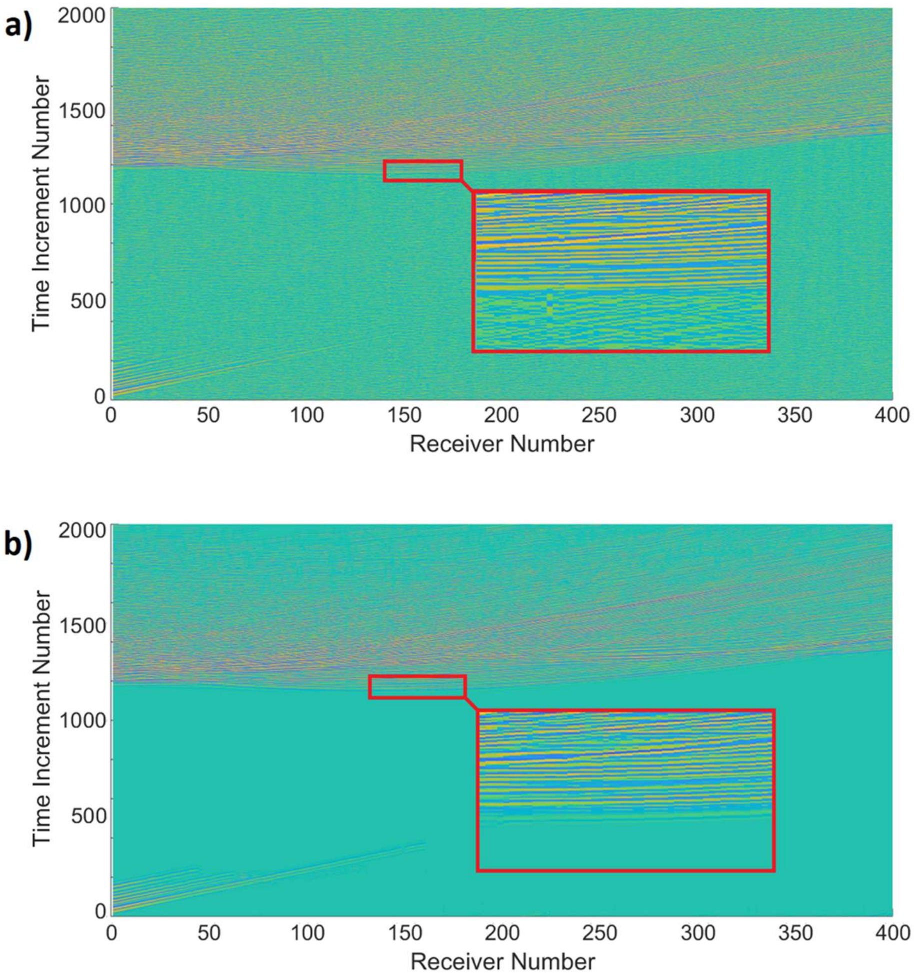

Figure 1.12. (**a**) The original data from ocean seismic gathers with strong reflections. (**b**) The data reconstructed after applying 80% compression with SMD to the original data. Areas highlighted in red are enlarged 3 times along x-axis and 6 times along y-axis.

The first data set (Figure 1.12) contains strong reflections arriving far after the direct wave which can be seen at early times on the few receivers close to the source. In this test, SMD was able to identify prominent waveform-related features while almost completely ignoring noise-dominated sections of data. The Figure 1.12 highlights in red an enlarged portion of the data (3 times along x-axis and 6 times along y-axis) in which we can see the first arriving reflections. In the original data (Figure 1.12a), at the lower half of the enlarged window we can also see the preceding noise. However, once the data is processed with SMD (Figure 1.12b) the preceding noise is no longer present. A principal drawback of SMD is that in some scenarios it will also ignore weaker incoming waves and fail to differentiate them from noise.

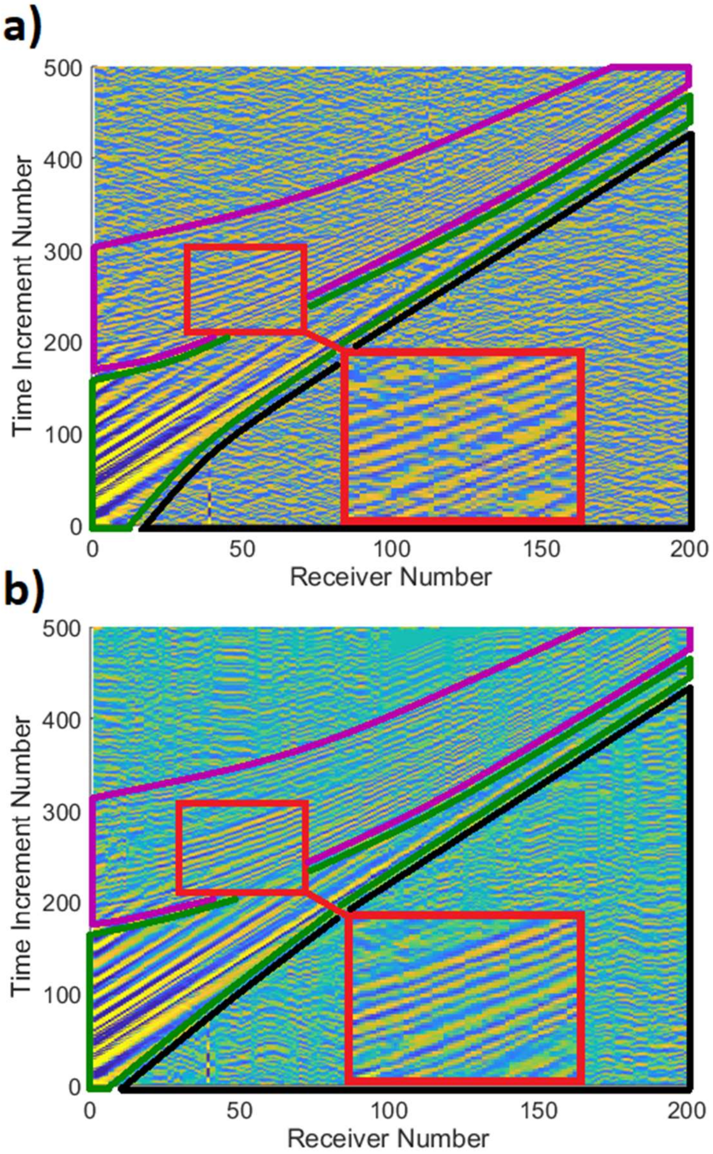

Figure 1.13. (**a**) The original data from ocean seismic gathers with weak reflections. (**b**) The data reconstructed after applying 80% compression with SMD to the original data. Areas highlighted in red are enlarged (2 times along the x- and y-axes). The reflected waves are circled with purple lines, direct wave with green lines, and the preceding noise with black lines.

The second data set (Figure 1.13) is from a different shot-gather in which we can observe weak reflections arriving closely after the much stronger direct wave. In Figure 1.13 the reflections are circled with purple lines, direct wave with green lines, and the preceding noise with black lines.

Red lines are used to highlight and enlarge (2 times along the x- and y-axes) an example of noise interfering with the reflected waves. The reflections are hard to differentiate from the more noticeable direct waves and because of their small amplitude, they are noticeably distorted by noise. In this test, SMD identified and properly reconstructed both direct waves and reflected waves, while reducing noise in all parts of the data. The preceding noise (circled in black) is noticeably weaker in the data reconstructed from SMD results (Figure 1.13b) than in the original data (Figure 1.13a). Furthermore, the enlarged window (circled in red) in the original data (Figure 1.13a) shows reflected waves distorted by noise. However, in the data reconstructed from SMD results (Figure 1.13b), the enlarged window (circled in red) shows a more clear, denoised version of the reflected waves. Because SMD reduces the noise everywhere in the data, the weak reflected waves can be seen more clearly in the data reconstructed from SMD results (Figure 1.13b).

The examples in Figures 1.12 and 1.13 show subsets of the entire data files which are too big to be presented in a single figure. However, the large data size provides a good opportunity for testing the efficiency of SMD. A single data file, which contains 12 s of recorded data, has 1008 traces, each trace containing 3000 pressure samples. Processing this amount of data on a laptop with an i5-6200U CPU and 8 GB of RAM, without parallelization, takes 4.2 s on average. Therefore, SMD is sufficiently fast when processing marine seismic gathers in real time. However, distributed acoustic sensing produces produces greater volumes of data than a long line of receivers during marine gathers.

We propose two solutions as we prepare SMD to future application on seismic data obtained by DAS. First, the current SMD algorithm is optimized to provide a best representation of the coherent waves, for a predefined memory requirement. To make SMD more applicable to new developments in the data acquisition (DAS), we could run a new training stage that considers computation time as well as the quality of the results. This could create a new version of SMD that is more applicable to data obtained by DAS. Second is the introduction of signal detection which we will discuss in the following section. The large data file that requires 4.2 s of computation time is heavily populated with signal. This is not the case during microseismic monitoring during which most of the files contain only noise. Running an initial test to check for the presence of coherent waves, prior to fully processing the data with SMD, may significantly reduce the time spent on processing data during seismic monitoring.

Discussion

In addition to seismic data compression and noise reduction, SMD also provides a new method of seismic data analysis. Rather than a list of displacements distributed in space and time to describe incoming waves, we have pairs of singular vectors paired with shift vectors. In an ideal scenario, for each arriving wave the column vector represents the waveform, the row vector represents the amplitudes at different receiver locations and the shift vector represents relative arrival times. Even though the ideal case is rarely achieved, we believe that the results of SMD provide an excellent advance in the realm of seismic analysis especially with its application of machine learning. With SMD, identifying features that can be used for building models is facilitated when noise-reduced data is represented in the SMD-compressed format.

The application of machine learning to data compression was explored in [24], which applied SVD to synthetic data and developed a model for estimating source location and orientation. However, SMD may provide greater opportunities for machine learning application.

It is instructive to provide an example of how the SMD algorithm can help estimate physical properties such as elastic wave velocities. To this end, herein we predict the average acoustic velocity (α) in our model (seawater) by analyzing the results of the SMD algorithm. Specifically, we use curvature of the shift vectors, and the zero offset time to estimate the average velocity of the reflected waves. Since we have a controlled source, a wave's zero offset time is simply it's row position in the first column of the data matrix multiplied by the time difference between consecutive recordings (t_d). Since the row position of a wave can be determined from the shift vector and the first element of the column vector, our wave velocity estimation is obtained solely from the data stored in SMD results.

To derive the formula for wave velocity we must make several assumptions about our surroundings. First, we assume that the reflecting surface (the ocean floor) is horizontal, and that the depth of the ocean floor (z_f) is significantly larger than the horizontal distance from source to the receiver ($x \ll z_f$). This gives us the expression for the total distance travel by the reflected waves (d):

$$d \approx 2z_f + \frac{x^2}{4z_f}. \tag{1.11}$$

Since we are considering the average acoustic velocity (α) in our model, we can rewrite the expression (1.11) in terms of the reflected wave travel-time (T):

$$T \approx 2z_f\alpha + \frac{x^2}{4z_f\alpha}. \tag{1.12}$$

The zero-offset time T_0 is defined as travel-time T at zero horizontal distance ($x = 0$):

$$T_0 = \frac{2z_f}{\alpha}. \tag{1.13}$$

Taking the second derivative of the expression (1.12) with respect to horizontal position x gives:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{1}{2z_f\alpha}. \tag{1.14}$$

Using expression (1.13) to substitute ($2z_f$) with ($T_0\alpha$) in expression (1.14) gives:

$$\alpha \approx (\frac{\partial^2 T}{\partial x^2} T_0)^{\frac{-1}{2}}. \tag{1.15}$$

Expression (1.15) can be used to calculate the acoustic velocity if we can express both $\frac{\partial^2 T}{\partial x^2}$ and *T₀* in terms of SMD results.

In subsection "SMD Algorithm, Step 2" it is explained that the row position (p_j) of a wave in any column *j* is equal to the sum of the *j*-th element of the shift vector (***s***) and the waveform row (w_r):

$$p_j = s_j + w_r. \tag{1.16}$$

In subsection "SMD Algorithm, Step 3" it is explained that the waveform row (w_r) can be obtained from the first element of the column vector (***u***) and the predetermined length of the recorded waveform (l_w):

$$w_r = u_1 + \frac{l_w - 1}{2}. \tag{1.17}$$

Using expressions (1.16) and (1.17), the zero-offset time (T_0) can be described with the row position of the wave in the first column (p_1) multiplied by the time difference between consecutive recordings (t_d):

$$T_0 = t_d \left(s_1 + u_1 + \frac{l_w - 1}{2} \right). \tag{1.18}$$

The expression (1.18) will be used to obtain the zero-offset time from SMD results. If we assume that the relative arrival times were accurately recorded by the shift vector, we can make the following substitution:

$$\frac{\partial^2 T}{\partial x^2} = \frac{s'' t_d}{{x_d}^2} \tag{1.19}$$

where x_d is the distance between adjacent receivers. In the expression (1.19), *s''* is the second derivative of an element in the shift vector ***s*** with respect to the position (*j*) of the element in the shift vector.

The value of s'' is determined by fitting a parabola to first n_r elements of the shift vector. The parameter n_r need not have an exact value. While we want to use enough elements from the shift vector to confidently fit a parabola, we also want to only use data from the receivers close to the source in order to follow the ($x \ll z_f$) condition. Therefore, the parameter n_r is set to 50. Once we fit the parabola, the value of s'' is estimated to be the quadratic coefficient c_2, multiplied by 2:

$$j \epsilon [1, n_r] : \; s_j \approx c_0 + c_1 j + c_2 j^2, \quad s'' = 2c_2. \tag{1.20}$$

By plugging the expressions (1.18)–(1.20) into expression (1.15) we estimate the acoustic velocity. However, the values of a shift vector can be affected by interfering waves, or noise. To minimize the error from those sources, we estimate the velocity based on the first n_s pairs of shift vectors and column vectors. Similar to n_r, the n_s parameter does not need to be set to any specific value. In this example, the parameter ns is set to 5. The average acoustic velocity α is estimated as the weighted average of the results from the first n_s extracted pairs of a shift vector and a column vector. Each term is weighted by the quality of the parabolic fit, which is equal to the inverse of the error norm e:

$$s^k \rightarrow \{c_0^k, c_1^k, c_2^k\},$$

$$e^k = \sum_{j=1}^{n_r} \left(s_j^k - \left(c_0^k + c_1^k j + c_2^k j^2 \right) \right)^2,$$

$$e_{sum} = \sum_{k=1}^{n_s} \frac{1}{e^k},$$

$$\alpha = \sum_{k=1}^{n_s} \left(\left(2c_2^k \right) \left(s_1 + u_1^k + \frac{l_w - 1}{2} \right) \frac{t_d^2}{x_d^2} \right)^{\frac{-1}{2}} \left(\frac{1}{e^k e_{sum}} \right). \tag{1.21}$$

The formula for α, defined in expression (1.21), was applied to 20 marine field data files, each recording a seismic response from a unique and controlled seismic source. For each file, we used the formula from expression (1.21) to estimate the velocity. The average estimated velocity was 1601 m/s and the standard deviation among the results was 236 m/s . Considering the ocean depth being about 3 km, the correct average acoustic velocity was likely about 1500 m/s . If we assume the correct average wave velocity experienced by the reflected waves was 1500 m/s , then the average and the median error from the 20 velocity estimates were 12.1% and 8.3%, respectively.

The experiment in this subsection proves that there is a strong correlation between the shift vector and relative arrival times of the wave at different receivers. We were able to use that correlation to estimate the velocity of the waves without relying on any information about the medium through which the waves were traveling. We believe that there is also a strong correlation between the column vector and the waveform, as well as between the row vector and the relative amplitudes of the waveform at different receivers. However, proving these correlations will require further testing.

In this example, SMD was applied to seismic data from a controlled source. In such cases we can always be certain that there are coherent waves present. However, we are planning a wide range of applications for SMD. During microseismic monitoring, often recorded by DAS, we might want to analyze coherent waves that are not coming from controlled sources. Therefore, when applying SMD to data from microseismic monitoring we do not know in advance whether the data contains signals of interest. For that reason, we developed a method for recognizing coherent signal in noisy data, that also relies solely on the results from SMD.

We created a method that can differentiate between noisy data and data containing signal, by measuring the curvature in the first several shift vectors. Applying SMD to noisy data without coherent waves returns a set of completely unrelated shift vectors. On the other hand, applying

SMD to data containing coherent waves, usually coming from the same source, returns a set of shift vectors most of which have similar curvatures. Therefore, the method differentiates between noisy data and data containing coherent waves, by calculating the standard deviation among the curvatures from the first 10 extracted shift vectors. To test this method, we applied SMD separately to two subsets of data from each of the 20 previously mentioned data files. The first subset contains only the recordings of noise, and the second subset contains recordings of both the coherent waves and the preceding noise (Figure 1.14).

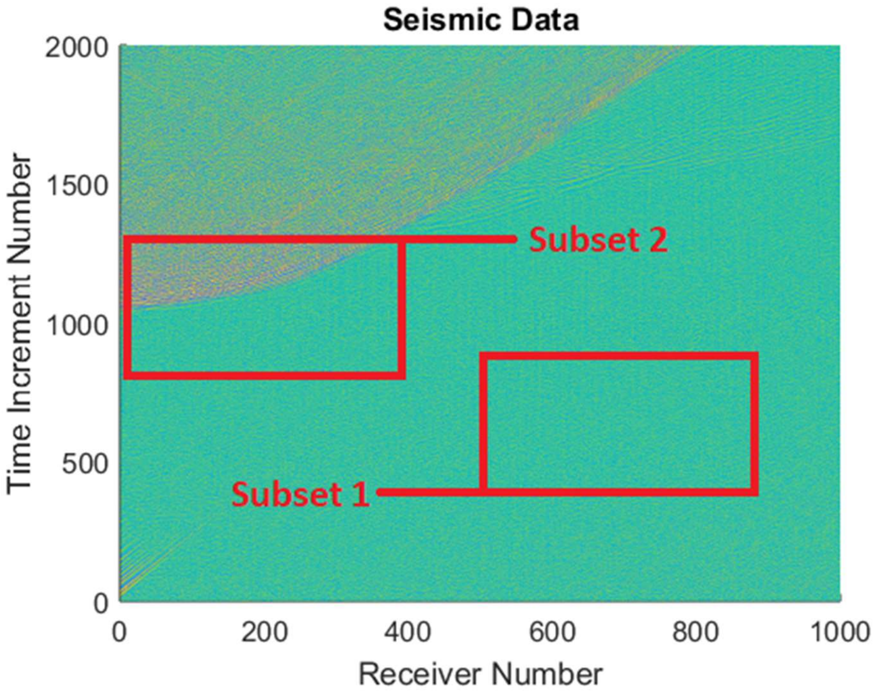

Figure 1.14. Two subsets from a marine field data file. The first subset contains only noise, and the second subset contains both a part of the coherent waves and some of the preceding noise.

For each subset, we use expression (1.20) to estimate the curvatures of the first 10 extracted shift vectors and then we compute the standard deviation of the 10 curvature values. The results are presented in Figure 1.15, in which the standard deviation for each subset without coherent waves is presented in blue, and the standard deviation for each subset containing coherent waves is presented in red. Due to shift vector curvature having random values when SMD results represent noise, the standard deviation among shift vector curvatures should be greater for the data subsets which contain only noise. As can be seen in Figure 1.15, the standard deviation among shift vectors is consistently lower for subsets containing coherent waves, which are presented in red. This confirms that we can differentiate between data containing only noise and data containing signal, by using our described method.

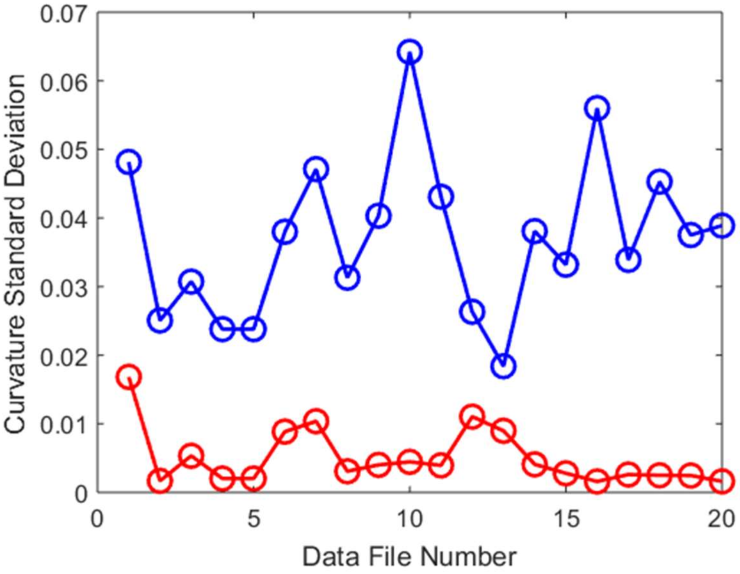

Figure 1.15. Standard deviation of shift vector curvature from each of the 20 data files. The results from subsets containing only noise are presented in blue and the results from subsets containing parts of coherent waves are presented in red.

It is important to emphasize that this signal detection method does not contain any information regarding the amplitude of the waves. It is entirely dependent on the curvature of the recorded shift vectors, which is a unique property. This means that this method, while accurate, can also be used to complement other signal detection methods, all of which rely on other properties in seismic data (such as STA/LTA for example, which relies on the amplitudes of the waves). Furthermore, this signal detection method only requires the first 10 shift vectors, which can be acquired very quickly compared to the to the processing time of the SMD. Therefore, when applying SMD to data obtained during seismic monitoring, we suggest first running a quick version of SMD that only extracts the first 10 shift vectors in order to run signal detection. If the presence of coherent waves is confirmed, SMD may continue to fully process the data.

The analysis of SMD results that was conducted in this subsection was not based on machine learning but instead on our understanding of the correlations between compressed data and attributes of the recorded waves. In the future, those strong correlations will be used with machine learning to train algorithms to accurately infer various properties of the source and the surrounding velocity model directly from compressed data. Additionally, application of data from marine surveys is difficult because of large amounts of interference between arriving waves. Thus, the SMD method may be more effective in different circumstances, such as in unconventional reservoirs monitored by distributed acoustic sensing.

Thus far, we only considered application of SMD to two-dimensional seismic data, obtained by a single line of receivers. However, receivers may often be distributed over an area, in a large number of lines. In such case, SMD may still be applied individually, to the data collected from each line. However, there may be a lot of redundancy between SMD results from each of the lines of receivers. Similar redundancies can also occur by applying SMD to multiple files from different times, obtained by a same set of receivers. In future work, we will take advantage of the fact that SMD results may be similar for batches of data collected from different receivers or during different times. Dictionary learning could be applied to SMD results to further compress the data, although we cannot yet recommend a method for learning the dictionary. Future work will therefore likely include further compression by applying dictionary learning to SMD results from multiple data files or multiple receiver lines.

Conclusions

Shifted-matrix decomposition is a powerful tool that can simultaneously compress and improve seismic data. This is done by converting the seismic data from a matrix into a set of pairs of singular vectors coupled with shift vectors which require less memory to store the seismic data. Furthermore, reconstructing the seismic data from compression results, creates a denoised version of the original data. Shifted-matrix decomposition provides an improvement to some of the existing denoising techniques such as SOSVD by avoiding numerical artifacts in areas with multiple intersecting waves of different slopes. This allows us to apply SMD to complicated field data with a large number of arriving waves and still achieve 80% compression.

When applied to synthetic data and ocean gathers it was able to boost coherent signal and erase the majority of the noise. In synthetic data with signal-to-noise ratio of 1.9, it was able to increase the ratio to 4.7 in the case of 80% compression and to 12.3 in the case of 95% compression. However, the excellent result achieved with 95% compression in the synthetic data is an artifact of the lack of coherent noise and the small number of coherent waves. Due to noise coherence and more complicated overlap of coherent waves, we recommend applying 80% compression to examples of field data in order to accurately represent the signal of interest. In field data, SMD was able to reduce the noise in areas preceding the signal as well as in areas containing coherent waves. As a results, weak waves that were difficult to notice in the original data, can be seen more clearly in the data reconstructed from SMD results. The only drawbacks are that in some scenarios SMD may fail to boost weaker signals and it is not meant to be applied to seismic data obtained from a small number of receivers.

There is a good correlation between the physical properties such as elastic wave velocity and the results of the SMD. As an example, the average wave velocity in the medium through which the waves propagate (seawater) was roughly estimated by analyzing the shift vector curvature. In the future, we will use machine learning to build models that infer with high accuracy the properties of the source and the velocity model, directly from the SMD results. Results of SMD can also be used for other analysis, such as signal detection. By analyzing only the first several shift vectors, we can check for the presence of coherent waves. Because it requires such a small amount of data, this technique can be executed very quickly by taking only the initial results of SMD. We recommend using it during microseismic monitoring before fully processing the data with SMD.

References

[1] Wessels, Scott A., Alejandro De La Peña, Michael Kratz, Sherilyn Williams-Stroud, and Terry Jbeili. "Identifying faults and fractures in unconventional reservoirs through microseismic monitoring." *First break* 29, no. 7 (2011).

[2] Li, Xinyang, Jimmy Zhang, Marcel Grubert, Carson Laing, Andres Chavarria, Steve Cole, and Yassine Oukaci. "Distributed acoustic and temperature sensing applications for hydraulic fracture diagnostics." In *SPE Hydraulic Fracturing Technology Conference and Exhibition*. OnePetro, 2020.

[3] Spanias, Andreas S., Stefan B. Jonsson, and Samuel D. Stearns. "Transform methods for seismic data compression." *IEEE Transactions on Geoscience and Remote Sensing* 29, no. 3 (1991): 407-416.

[4] Villasenor, John D., R. A. Ergas, and P. L. Donoho. "Seismic data compression using high-dimensional wavelet transforms." In *Proceedings of Data Compression Conference-DCC'96*, pp. 396-405. IEEE, 1996.

[5] Khene, M. F., and S. H. Abdul-Jauwad. "Adaptive seismic compression by wavelet shrinkage." In *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No. 00TH8496)*, pp. 544-548. IEEE, 2000.

[6] Ma, Jianwei, Gerlind Plonka, and Hervé Chauris. "A new sparse representation of seismic data using adaptive easy-path wavelet transform." *IEEE Geoscience and Remote Sensing Letters* 7, no. 3 (2010): 540-544.

[7] Wang, Chao, and Yun Wang. "Robust singular value decomposition filtering for low signal-to-noise ratio seismic data." *Geophysics* 86, no. 3 (2021): V233-V244.

[8] Kreimer, Nadia, and Mauricio D. Sacchi. "A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation." *Geophysics* 77, no. 3 (2012): V113-V122.

[9] Cavalcante, Quézia, and Milton J. Porsani. "Prestack seismic data reconstruction and denoising by orientation-dependent tensor decompositionPrestack seismic data reconstruction." *Geophysics* 86, no. 2 (2021): V107-V117.

[10] Beckouche, Simon, and Jianwei Ma. "Simultaneous dictionary learning and denoising for seismic data." *Geophysics* 79, no. 3 (2014): A27-A31.

[11] Beckouche, Simon, and Jianwei Ma. "Simultaneous dictionary learning and denoising for seismic data." *Geophysics* 79, no. 3 (2014): A27-A31.

[12] Siahsar, Mohammad Amir Nazari, Saman Gholtashi, Vahid Abolghasemi, and Yangkang Chen. "Simultaneous denoising and interpolation of 2D seismic data using data-driven non-negative dictionary learning." *Signal Processing* 141 (2017): 309-321.

[13] Payani, Ali, Afshin Abdi, Xin Tian, Faramarz Fekri, and Mohamed Mohandes. "Advances in seismic data compression via learning from data: Compression for seismic data acquisition." *IEEE Signal Processing Magazine* 35, no. 2 (2018): 51-61.

[14] Tian, Xin. "Multiscale sparse dictionary learning with rate constraint for seismic data compression." *IEEE Access* 7 (2019): 86651-86663.

[15] Chen, Yangkang. "Fast dictionary learning for noise attenuation of multidimensional seismic data." *Geophysical Journal International* 209, no. 1 (2017): 21-31.

[16] Bekara, Maïza, and Mirko Van der Baan. "Local singular value decomposition for signal enhancement of seismic data." *Geophysics* 72, no. 2 (2007): V59-V65.

[17] Gan, Shuwei, Yangkang Chen, Shaohuan Zu, Shan Qu, and Wei Zhong. "Structure-oriented singular value decomposition for random noise attenuation of seismic data." *Journal of Geophysics and Engineering* 12, no. 2 (2015): 262-272.

[18] Fomel, Sergey. "Applications of plane-wave destruction filters." *Geophysics* 67, no. 6 (2002): 1946-1960.

[19] Krawczyk, Charlotte, Philippe Jousset, Gilda Currenti, Michael Weber, Rosalba Napoli, Thomas Reinsch, Giorgio Riccobene, Luciano Zuccarello, Athena Chalari, and Andy Clarke. "Monitoring volcanic and seismic activity with multiple fibre-optic Distributed Acoustic Sensing units at Etna volcano." In *EGU General Assembly Conference Abstracts*, p. 15252. 2020.

[20] Zhan, Zhongwen. "Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas." *Seismological Research Letters* 91, no. 1 (2020): 1-15.

[21] Zhan, Zhongwen. "Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas." *Seismological Research Letters* 91, no. 1 (2020): 1-15.

[22] Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. Chemom. Intell. Lab. Syst. 1987, 2, 37–52.

[23] Golub, Gene H., and Charles F. Van Loan. *Matrix computations*. JHU press, 2013.

[24] Yatsenko, Maxim, Milan Brankovic, Eduardo Gildin, and Richard L. Gibson. "A Novel Approach to Discovery of Hidden Structures in Microseismic Data Using Machine Learning Techniques." In *SPE Europec featured at 81st EAGE Conference and Exhibition*. OnePetro, 2019.

CHAPTER III

EVENT DETECTION IN DAS DATA BASED ON SHIFTED-MATRIX DECOMPOSITION

Introduction

Increasing global energy demand and the push towards decarbonization due to growing concerns over climate change have led to an intensification of renewable energy development as a means to reduce emissions in the energy sector [1,2,3]. Geothermal energy, while not as widely used as other renewables such as solar [4] and wind [5], may play an important role in the future of sustainable energy sources. Geothermal energy can provide a consistent power output that is independent of weather conditions [6]. Thus, adding a geothermal power source brings consistency to a future power grid that is heavily reliant on renewable energy.

In order to convert geothermal energy into electricity, the construction of enhanced geothermal system (EGS) infrastructure [7] is necessary. The development of EGS infrastructure starts by drilling an injection well into a geothermal source of energy. Then, pressurized water is injected into the hot rock to create a fractured volume. After a sufficient number of fractures have opened, production wells are installed. Once the injection well and production wells are established, EGS heats up water by pushing it into the hot subsurface through an injection well and then extracts it through a production well. The hot water can then be used for either heating or electricity production.

The main challenges with the construction of EGS concern the subsurface fracture network. Specifically, for EGS to be successful, the injection well must create a fracture network of sufficiently large volume to pump out a substantial amount of heat. Second, the fractures must be connected to the production wells. The latter should be placed such that the water flows from the injection well, through the fractures, to the production wells. Finally, the risk of induced seismicity during the development of the fracture network must be carefully managed.

One way to tackle these challenges is through microseismic event monitoring. By analyzing seismic data and tracking fracture propagation paths one can create a highly accurate and dynamic image of the subsurface. Such imagery provides important information for determining when the reservoir is adequately developed and where to place production wells. Knowledge of the fracture geometry also helps to estimate the risk of induced seismicity.

Acquisition of seismic data is generally performed with geophones. However, recent developments in reservoir monitoring utilize fiber optic cables to record both low-frequency strain and higher-frequency seismic waves in a technique called distributed acoustic sensing (DAS) [8]. DAS provides a unique view of the reservoir by recording strain rate data with high temporal and dense spatial sampling rates. Higher recording rates and abundant receiver positions increase the amount of microseismic activity that is recorded during reservoir monitoring.

Compared to downhole geophones, DAS fiber returns a smaller signal to noise ratio and can measure strain in only one direction, that of the borehole axis, unless deployed in a spiral around the well, which is a difficult operation. However, DAS fiber is more heat- and pressure-resistant than geophones and can be deployed closer to the geothermal energy source. Furthermore, DAS fiber can be deployed along-side any well, such as an injection well, eliminating the need for a dedicated monitoring well. Therefore, DAS provides a cost-effective alternative to downhole geophone deployment [9]. The problem of locating events while using DAS (because DAS usually records strain in only one direction) can be somewhat overcome by combining DAS data with surface geophone data [10], or by using multiple DAS cables.

It is important to note that the use of DAS data brings new challenges. Due to the extremely large amount of data captured by DAS, it can be difficult to process all the data in real time, a necessity for monitoring the propagation of fractures during hydraulic stimulation operations. Here, I introduce a highly efficient algorithm that uses shifted-matrix decomposition (SMD) to detect seismic events in DAS data. Shifted-matrix decomposition is a data decomposition method specifically developed for seismic data gathered by a large number of receivers [11]. The SMD-based detection algorithm is herein applied to DAS data gathered at the FORGE geothermal site in Utah [12].

Methodology

As stated above, I will use a data decomposition method (SMD) to detect seismic events in DAS data. In this section, I provide a description of the SMD output, and how it is used to differentiate between files containing seismic signal and those containing only noise. The following is a quick review of SMD [11].

The input to the shifted-matrix decomposition algorithm is a matrix containing seismic data. The row position indicates time, and the column position indicates receiver number. The SMD output is a series of sets, each set comprising three vectors. Two of the three vectors are referred to as basis vectors, (also previously refer to as singular vectors), and the third one is referred to as a shift vector. Similar to other data decomposition methods, the original data can be approximately reconstructed by summing the outer-products of the pairs of basis vectors. However, in SMD each outer-product has its columns shifted, by the amount specified in the shift vector, before being added to the sum. Equation 2.1 describes how the original seismic-data matrix $\boldsymbol{M}$ can be reconstructed from pairs of basis vectors ($\boldsymbol{a}^i$, $\boldsymbol{b}^i$) and corresponding shift vectors s^i:

$$\boldsymbol{M} \approx \sum_i \chi(\boldsymbol{a}^i {\boldsymbol{b}^i}^T, \boldsymbol{s}^i). \tag{2.1}$$

In the equation above, χ is an operator that takes a matrix ($\boldsymbol{a}^i\boldsymbol{b}^{iT}$) as its first argument and shifts the columns of said matrix up or down based on the values in the shift vector $\boldsymbol{s}^i$.

The result of including a shift vector in the data decomposition process is that the output more accurately captures coherent seismic waves. In an ideal scenario, each seismic wave arriving at the fiber is perfectly recorded with a single pair of basis vectors and a corresponding shift. The first basis vector captures the waveform, the second basis vector captures the amplitude of the wave at each receiver, and the shift vector captures the arrival time of the wave at each receiver. For this reason, I refer to the first basis vector as the "waveform vector" and the second basis vector as the "amplitude vector." An illustration of the ideal scenario is presented in Figure 2.1. Even though the ideal case is rarely achieved, there usually is a strong correlation between the SMD results and the aforementioned properties of the recorded waves.

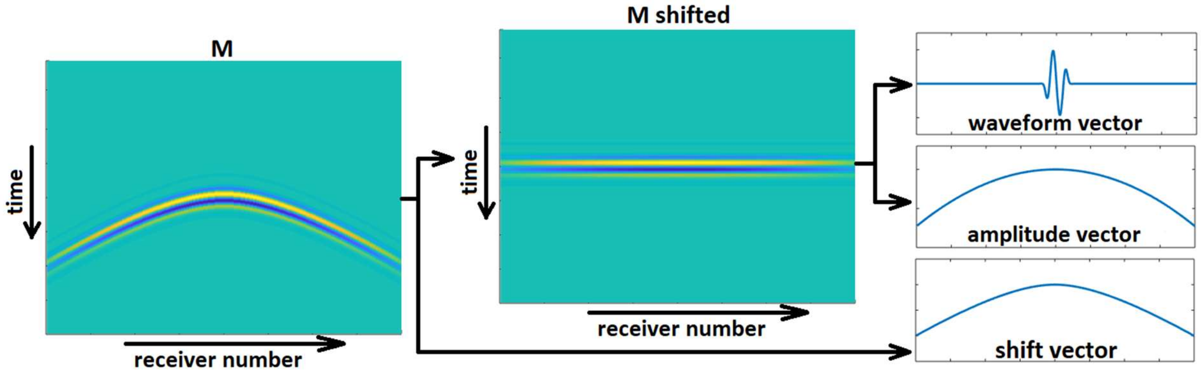

Figure 2.1. Shifted matrix decomposition applied to matrix ***M*** containing a recording of a single seismic wave to generate a waveform vector (first basis vector), an amplitude vector (second basis vector) and a shift vector.

A detailed description of SMD algorithm can be found in [11]. Here, I provide only a brief overview necessary for understanding the SMD output. Let ***M*** be a seismic data matrix to be processed by SMD. At the start, the algorithm uses an adaptive geometric mean filter to locate an element, or point, in matrix ***M*** that is part of a seismic wave, or rather, that has a high likelihood of being part of a seismic wave. The strain rate (or displacement) recorded by the elements in the same column and near the selected point is considered to be a temporary estimate of the waveform. We refer to this set of values as the ‘waveform sequence’. Then, the algorithm attempts to identify the waveform sequence in the surrounding columns using cross correlation. The search starts from the columns adjacent to the selected point and expands further outward as long as the correlation with the waveform sequence remains high. If the correlation drops below a predefined value of 0.25, the algorithm stops searching for the waveform sequence. As a result, the shift vector and the amplitude vector refer only to the subset of the columns in the matrix from which the waveform is identified. It is important to note that if the basis vectors and the shift vector are describing an actual seismic wave, the waveform sequence will often be noticeable in a large set of columns and the shift vector and the amplitude vector will contain many elements. Otherwise, if the basis vectors and the shift vector describe noise, the shift vector and the amplitude vector will contain fewer elements.

Once the row position of the waveform sequence has been identified in all the examined columns, the entries of the latter are shifted so that the wave becomes flattened, i.e. it appears in the same set of rows in every column. Finally, the wave is extracted, or rather, the basis vectors are recorded, and their outer product is subtracted from the matrix. Then, the columns are shifted back to their positions and the whole process is repeated in order to extract any remaining seismic waves. The algorithm terminates when a performance criterion is satisfied. The workflow is illustrated in Figure 2.2.

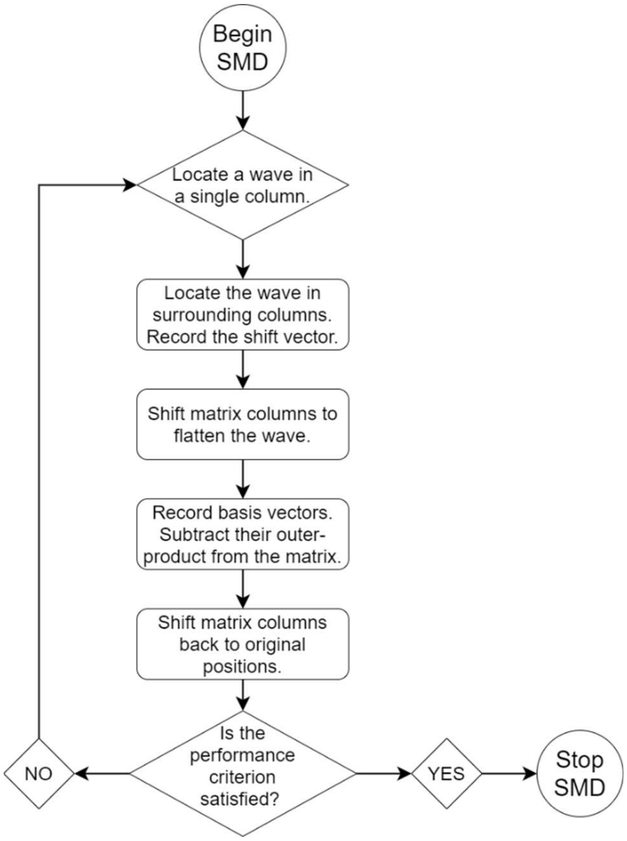
Begin SMD
Locate a wave in a single column.
Locate the wave in surrounding columns. Record the shift vector.
Shift matrix columns to flatten the wave.
Record basis vectors. Subtract their outer-product from the matrix.
Shift matrix columns back to original positions.
Is the performance criterion satisfied?
NO
YES
Stop SMD

Figure 2.2. A flowchart describing the SMD algorithm.

Once the SMD has processed a matrix containing seismic data, the next step is to analyze the SMD output to determine whether any seismic waves have been found. Specifically, I have created a signal detection algorithm that takes the SMD results as input and returns an estimate of the likelihood of seismic waves being present in the data.

It would be intuitive to regard the sum of the elements in the amplitude vector as a diagnostic of seismic waves. However, there are two types of noise that could affect the results of a signal detection algorithm based on this diagnostic. The first noise type is a sudden high value that appears in a small set of adjacent traces, usually fewer than ten. To avoid the effect of such noise bursts on the sum of elements in the amplitude vector, I subtract the sum of the ten highest values from the sum of all the values in the amplitude vector. Also note that the length of the amplitude vector is dependent on the number of columns in which SMD has identified the supposed wave. Therefore, if a waveform has been identified in a large number of columns, as usually is the case with coherent seismic waves, then the sum of the elements in the amplitude vector will be high.

The second type of noise, characteristic of the Forge DAS data set, is shown in Figure 2.3. The "ambient noise" (the pervasive signal curving downward to the right) in Figure 2.3 has some properties similar to seismic waves (the localized signals curving downward to the left). For example, the noise has high amplitudes and is present in a large number of columns; thus, the ambient noise is likely to be present in the SMD output. To prevent the ambient noise from affecting the results of the seismic-wave detection algorithm, I rely on the information stored in the shift vectors. In Figure 2.3, the ambient noise is traveling downward, from shallower depths to deeper depths. At the Forge site, seismic events occur near the injection well which is ~1000 m below the lowest point in the vertical monitoring well. This geometry implies that waves emanating from seismic events of interest will be traveling upwards as they reach the monitoring well equipped with the DAS fiber. Therefore, ambient noise in the detection algorithm can be ruled out by retaining only waves that are traveling upwards. These waves can easily be identified in SMD results because arrival times are recorded in the shift vector. Since I analyze data recorded by DAS at depths between 300-900 m, the detection algorithm is instructed to consider only waves traveling upward at velocities slower than 6000 m/s. The latter value is well above the expected wave velocities in the geological formation at this depth range.

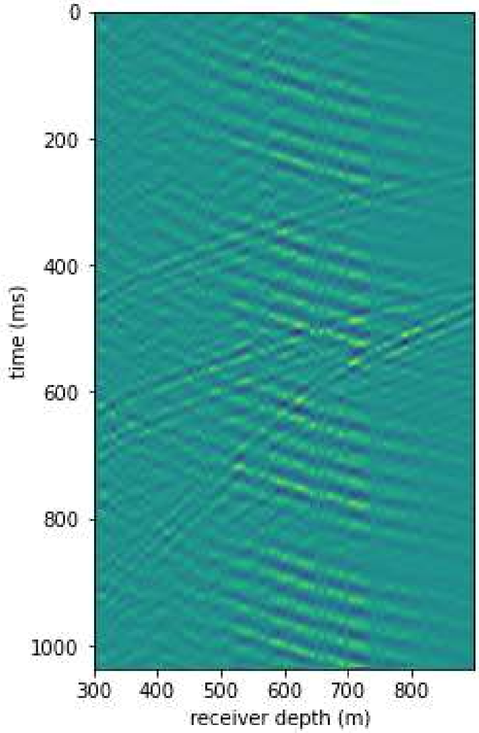

Figure 2.3. Ambient noise in DAS data.

Above, I have described how the detection algorithm processes a single pair consisting of an amplitude vector and a corresponding shift vector. I refer to this output as an "amplitude vector sum". However, the output of SMD is a series of pairs of waveform and amplitude vectors, coupled with corresponding shift vectors. Those results must be added together to create a complete picture of a file containing seismic data. Specifically, I consider two quantities. The first is the value of the greatest amplitude vector sum, and the second is the five greatest amplitude vector sums combined. However, rather than adding the five sums together, I consider their arrival time, or rather, the times at which they are recorded in the data. For example, noise bursts occur at random times and therefore their difference in row positions is random. On the other hand, seismic waves that that are propagating from a single seismic event will have similar arrival times (similar row positions). Therefore, considering the row position (or arrival time), which is recorded in the shift vector, enables a differentiation between signal and noise. An example of such process is provided in Figure 2.4.

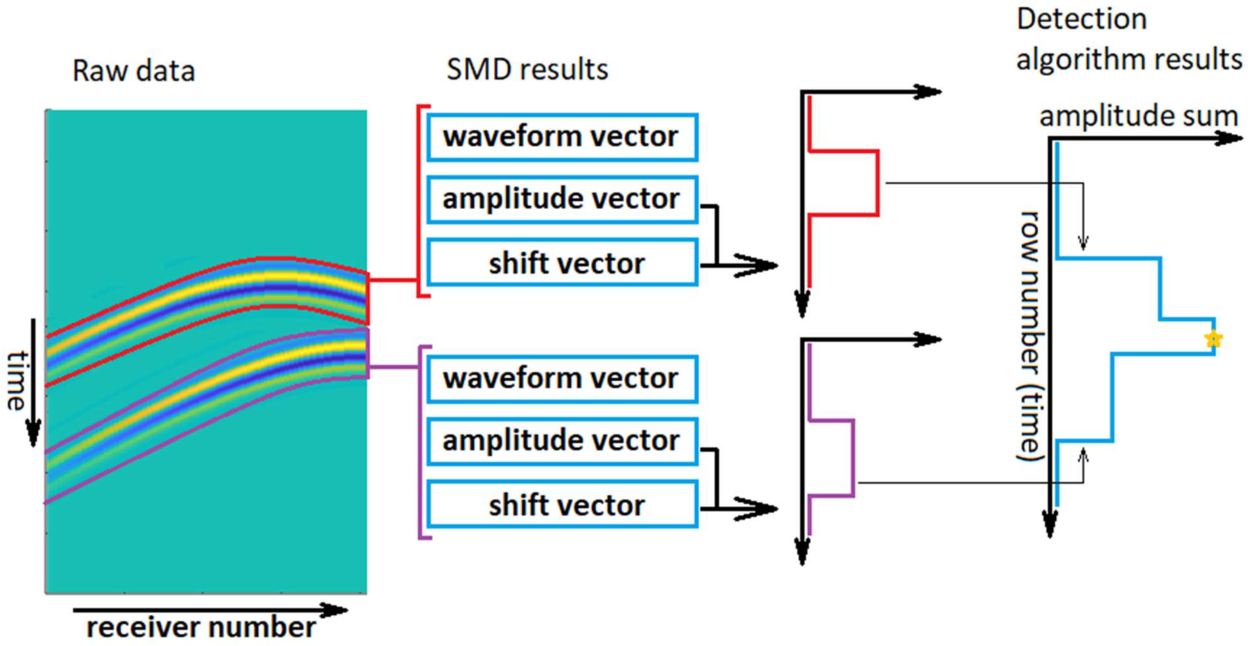

Figure 2.4. The overview of the process for detecting seismic waves in the data.

Specifically, to combine all the amplitude vector sums, the algorithm creates a new vector, termed the "result vector." The latter contains one value for each row in the seismic data matrix. For each pair of amplitude and shift vectors, the amplitude vector sum is added to the result vector, but only to elements corresponding to the matrix rows in which the waveform is present. Thus, if two distinct waves are closely spaced in time, they will be added to a similar set of rows in the result vector. If they are far apart in time, they will be added to different sets of rows in the result vector. The final output is the maximum value of the result vector, along with its position in the vector. A matrix containing seismic data is classified as containing only noise, or containing seismic waves, based on the maximum value of the result vector, and the value of the greatest amplitude vector sum. Optimal threshold values for these two classification criteria are determined by applying the detection algorithm to a small subset of all the seismic data files.

Results

As previously stated, I have applied the SMD seismic event-detection algorithm to DAS data recorded at the FORGE site in Utah [12]. At the FORGE site, the enhanced geothermal system comprises two wells, an injection well and an observation well. The injection well is ~2242 m deep with most microseismic events occurring near the bottom of the well. The observation well is 985 m deep and laterally separated from the injection well by ~360 m. The observation well is equipped with twelve geophones and DAS fiber, which records data every 1 m at a rate of 2000 samples/s along the entire length of the well. An illustration of the wells and surrounding geology is present in Figure 2.5.

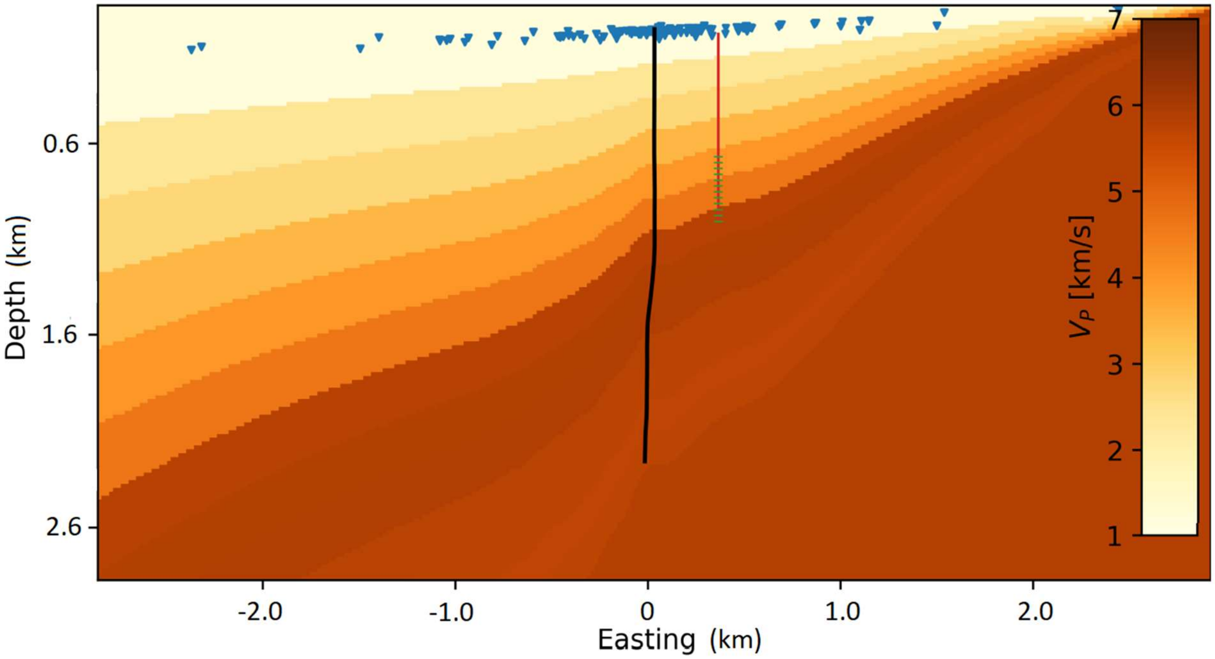

Figure 2.5. A sketch showing the Injection well (black), the monitoring well (red), downhole geophones (green) and the surface geophones (blue). The DAS fiber is deployed along the entire length of the monitoring well (red). This figure is taken from [10] and modified.

The SMD algorithm is not applied directly to the raw FORGE dataset. Instead, I first apply several basic pre-processing filters to improve the signal to noise ratio. An example of the filters applied is presented in Figure 2.6. First, a 2.5 ms averaging window is applied. Then all the traces are normalized to unit maximum amplitude in each trace. Next, I extract the systematic noise pattern (shown in Fig.2.6, second panel from left) that is present in all traces. The noise pattern is identified by summing all the columns into a single column, termed the “noise column”, which is then normalized (each entry is divided by the magnitude of the noise column). For each trace I subtract the noise column scaled by the dot product of the trace and the noise column. By doing this, the systematic noise is removed from the data. The final pre-processing step is frequency filtering. Spectral analysis of the continuous data revealed extraneous electrical noise at high frequencies, above the dominant frequency range of 20-50 Hz for microseismic events [12]. I use a filter that rejects frequencies below 20 Hz and above 60 Hz. It should also be noted that due to excessive noise recorded by DAS at the top and the bottom of the monitoring well, we only use data between the depths of 300 m and 900 m.

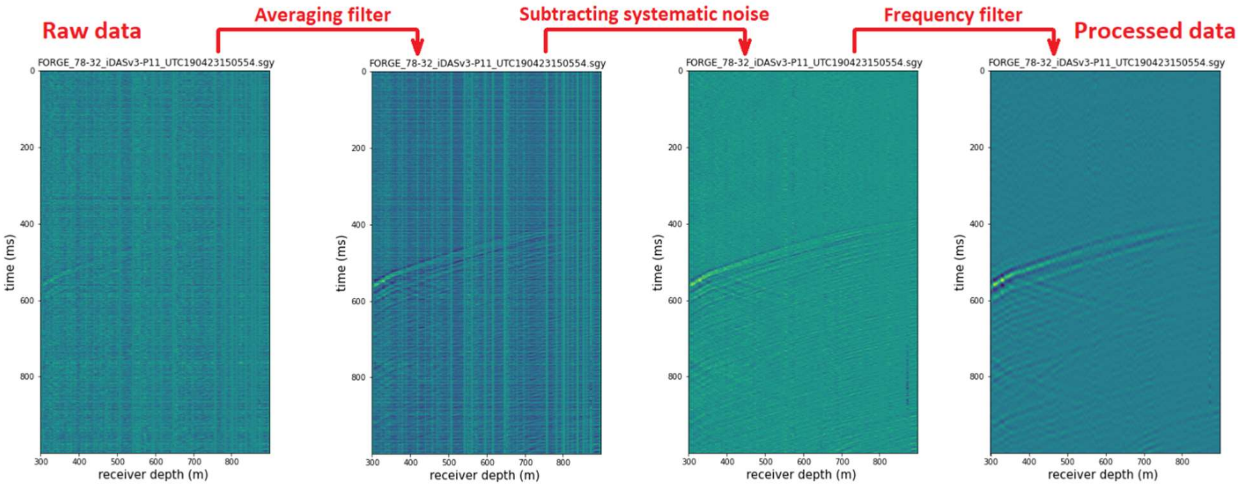

Figure 2.6. Example of the SMD pre-processing steps on a subset of data from a DAS data file.

To analyze the results of SMD detection, I compare my detected events to the set of detected seismic waves from two other FORGE seismic catalogs. The two catalogs recorded seismic events that emanated only from a ‘zone of interest’ which refers to the volume surrounding the bottom of the injection well.

The first seismic catalog [13] is generated mainly by data from geophones placed in the monitoring well. The geophones recorded data only during reservoir stimulation intervals. During other time periods, the catalog contains data from a three-component, short-period sensor installed at FORK, a seismic station located near the FORGE site.

The second catalog of events [9] is generated solely from an analysis of the DAS data and thus comprises a good performance benchmark for the SMD detection algorithm. The algorithm used to generate the second catalog is described in detail elsewhere [14], but here I provide a summary. The algorithm first uses a well perforation shot of known time and location to estimate the P-wave velocities in the geological formation adjacent to the monitoring well. To find the S-wave velocities, it uses a different event that generated strong S-waves. Based on the resulting velocity model, the algorithm calculates the moveout of a plane wave along the monitoring well as a function of the incidence angle at the bottom of the well. Then, operating on a given DAS data matrix, the columns are shifted once for each angle in a set of trial incidence angles. The columns are shifted such that the curved waves arriving at the said incidence angle are transformed into flat waves in the matrix. Once the columns are so shifted for a given incidence angle, a ‘semblance’ function is computed for each row in the matrix. If there exists an incidence angle and a row position for which the semblance value exceeds a predefined threshold, the seismic matrix is marked as containing coherent seismic waves. I refer to this event detection algorithm as semblance-based detection.

Compared to SMD-based detection, downsides of the semblance-based method are that it is more computationally expensive, and it requires a known velocity model around the monitoring well, which can be difficult to obtain if there aren’t any strong events or perforation shots from known locations. The SMD method requires only upper and lower bounds for seismic wave moveouts. To process 15 s of DAS data, SMD-based detection takes ~2.4 s, several times faster than semblance-based detection.

To estimate the computation time of semblance-based detection, we created an algorithm with equal number of operations and complexity. For example, we might not know by how much the columns needed to be shifted, but it takes the equal amount of time to shift them correctly and randomly. The subsampling information and number of incidence angles tested wasn’t provided in [9], so we used information from an earlier application of semblance-based detection [14]. We estimated the computation time of semblance-based detection to be about 7.25 second. However, the author reported longer computation times, so it is likely that our estimates of subsampling and number of incidence angles tested were incorrect. It should also be noted that the computation time of SMD can also be further decreased with subsampling.

The reason for the speed advantage of the SMD method is that, while SMD shifts only a subset of a column that is proposed to contain a coherent wave, semblance-based detection shifts the entire column. Furthermore, for every incidence angle, to calculate semblance all columns must be shifted, squared and added together. However, the semblance-based algorithm does provide information on the incidence angle whereas SMD detects events only, without providing information on the incidence angle. Compared to SMD, semblance-based detection is a standard and well-tested method for processing seismic data.

The semblance-based detection was tested on DAS data acquired over the 24-hour period between April 27th 2022 at 5:00 pm and April 28th 2022 at 5:10 pm. Henceforth I will be using this time interval to compare the performance of the SMD and semblance-based detection schemes.

During this time interval, the geophones were active and 299 events were recorded in the first catalog. The second catalog [9] contains 110 DAS sensor-recorded events, all of which are also present in the first catalog. The first catalog of events will be used to determine if an event detected by SMD originated from the zone of interest or if it is presumably due to a nearby earthquake unrelated to the well-site hydraulic stimulation operations.

During the time interval described above, the SMD-based algorithm detected 86 events. Of these, 43 were already cataloged by the semblance-based detection algorithm. The remaining 43 events were not matched to any events in the first catalog [13]. Therefore, the remaining 43 events probably do not describe seismic waves originating from the zone of interest. Moreover, the SMD-based detection found fewer events than semblance-based detection.

First, I consider the set of events that were detected by both semblance-based and SMD-based detection. Figure 2.7 shows 4 such events in which distinct P-wave and the S-wave signatures are the most noticeable, and an additional event in which the P-wave and S-wave signatures are relatively weak compared to other events.

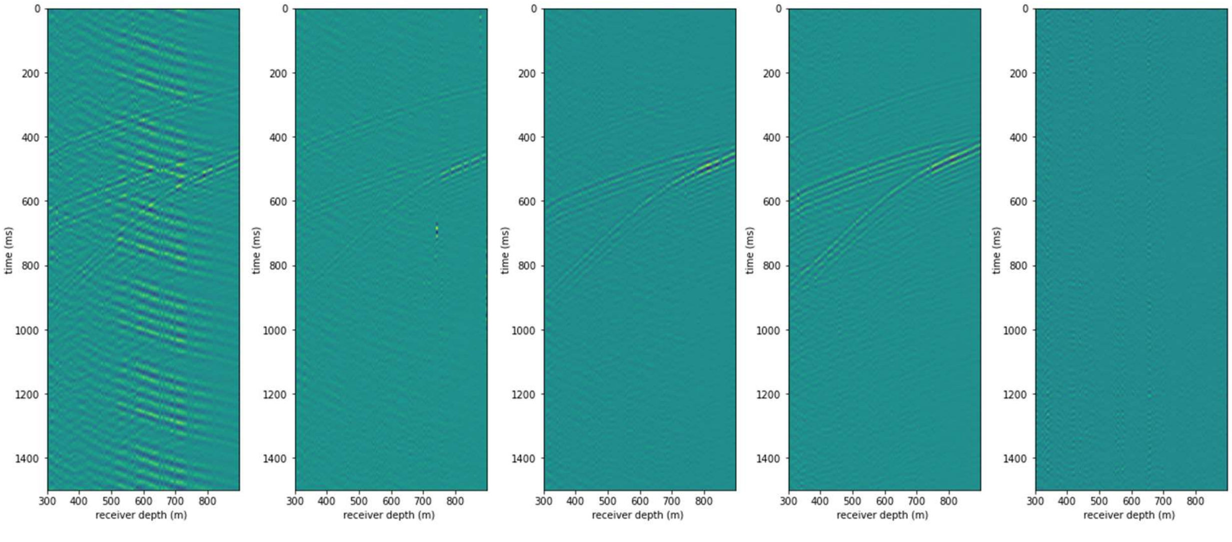

Figure 2.7. Four of the clearest events and, at right, one weak event from the set of events detected by both semblance detection and SMD detection algorithms. On the third plot from the right, P-wave, S-wave and S-P conversion are marked.

In the four clear events (leftmost four panels of Figure 2.7), we can see the P-wave, the S-wave and the S-P conversion (also marked on the third figure from the left with red, yellow and orange, respectively). The difference in arrival times between the P-wave and the S-wave at the receiver at 900 m depth ranges from 200 to 222 ms. We also see the S-P conversion being created at the depth of about 900 m which is consistent with the layer boundary seen in the velocity model from Figure 2.5. Based on the velocity model, we estimate the velocity in the granite layer below the boundary to be 6 km/s and uniform. Then the P-S time difference indicates that the distance from the source to the receiver at 900 m depth ranges from 1200 to 1332m, consistent with the position of the observation well relative to the bottom of the injection well (Figure 2.5).

Events that were detected by the SMD algorithm but are not present in the two published catalogs are presented in Figure 2.8. While we cannot show plots of all 43 such events, the plots in the figure provide a good representation of them.

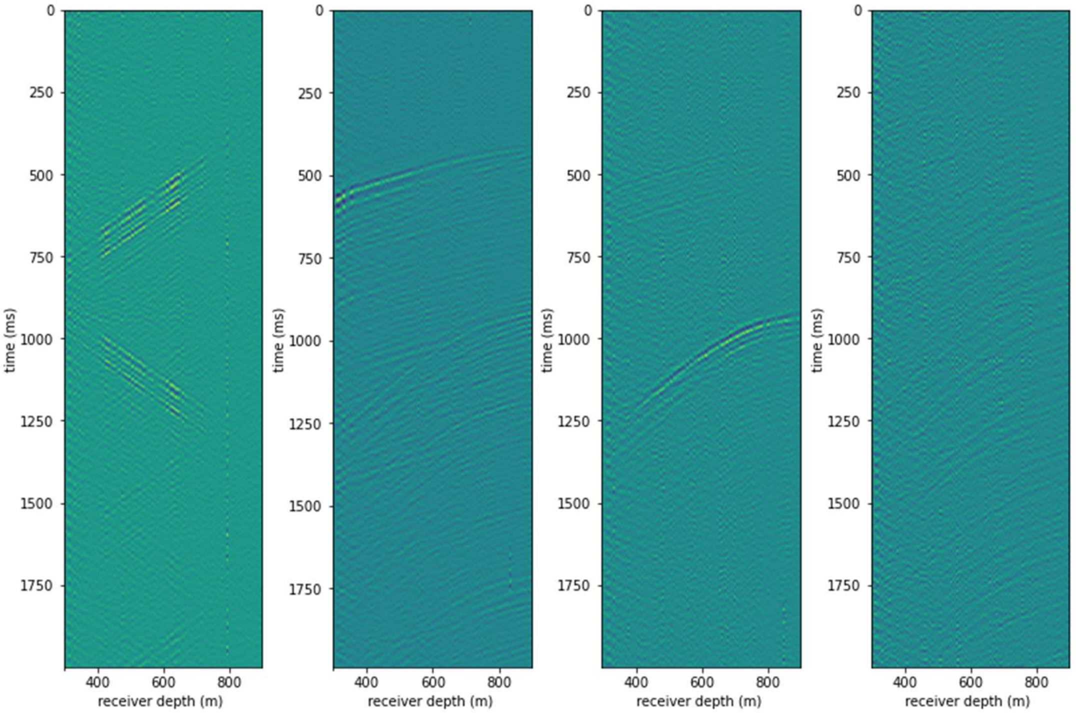

Figure 2.8. The representation of detected seismic waves that were not present in the published catalogs.

The leftmost plot in Figure 2.8 is interpreted as tube waves recorded by DAS, likely caused by site operators moving the geophones in the observation well. The SMD detection picked up the tube wave since it doesn't use specific information regarding velocity around the observation well. However, I could in future add a criterion to the SMD detection algorithm to exclude seismic tube waves when the moveout (estimated by the shift vector) fits a linear trend.

For the two events shown in the middle panels of Figure 2.8, I estimated the difference in arrival times between the P-wave and the S-wave to be 488 and 525 ms, respectively. These differences in arrival times indicate that the distance from the source is too great for the source to be located in the zone of interest, i.e. at the bottom of the injection well. Finally, the rightmost plot in Figure 2.8 shows a repetitive series of S-waves. Here the difference in arrival times between the P-wave and the S-wave (5.5 s) is too great to display them both in the same plot. The large number of arriving S-waves most likely indicate multiples of a head S-wave. In the middle two plots one can also observe multiples of the head S-wave but there are fewer of them. Since the S-wave on the rightmost plot traveled a greater distance than the S-waves in the middle two plots, it has passed through more formation heterogeneities and thus more multiples were generated. In conclusion, among this set of events, the seismic waves were either tube waves or waves emanating from sources that are not in the zone of interest.

Finally, I have examined events that were detected by the semblance-based algorithm but not by the SMD algorithm. Figure 2.9 shows the first three events from this set and an additional, later event for which the recorded seismic wave signatures are the clearest.

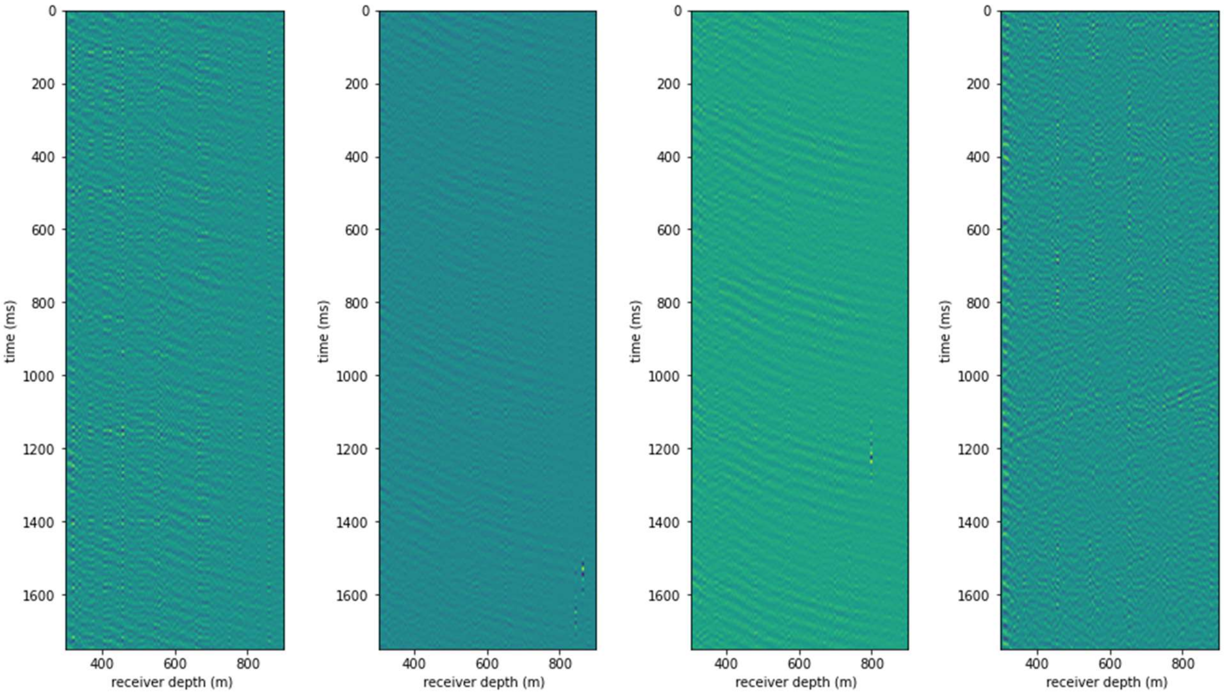

Figure 2.9. The first three events recorded by the catalog from [9] that weren't picked by SMD detection, and an event from the catalog with the most noticeable seismic waves that also wasn't picked by SMD detection. In all four plots, the events were detected in [9] at 750 ms time.

In the three leftmost plots of Figure 2.9, seismic waves are not evident. The non-appearance of seismic waves is most likely because my data pre-processing sequence is less extensive than that of [9]. Specifically, pre-processing steps in the semblance algorithm included a sample-by-sample median filter, frequency filtering and an f-k velocity filter. This pre-processing sequence is superior to my workflow, described at the beginning of this section. The main difference is that for the semblance-based algorithm the ambient noise was removed in the pre-processing stage, while SMD was applied to data still containing ambient noise and relied on the shift vectors to filter it out.

To see if I can generate results comparable to those of [9] and achieve a reduction in ambient noise, I added a step to the pre-processing workflow. I tried shifting the columns to maximize their correlation and then I subtracted from them the outer-product of the first pair of basis vectors. Figure 2.10 shows two examples of how the added ambient noise reduction changes the pre-processing results. The upper example shows the noise reduction done to the leftmost plot from Figure 2.9 and the lower example shows the nose reduction applied to the rightmost plot. While this did not make a big difference in the pre-processing, it did enhance the visibility of several events enough for SMD-based algorithm to detect them. For example, the added pre-processing was enough to make the bottom example (from Figure 2.10) visible to the SMD-based detection algorithm, but it was unable to make visible the event from the upper example.

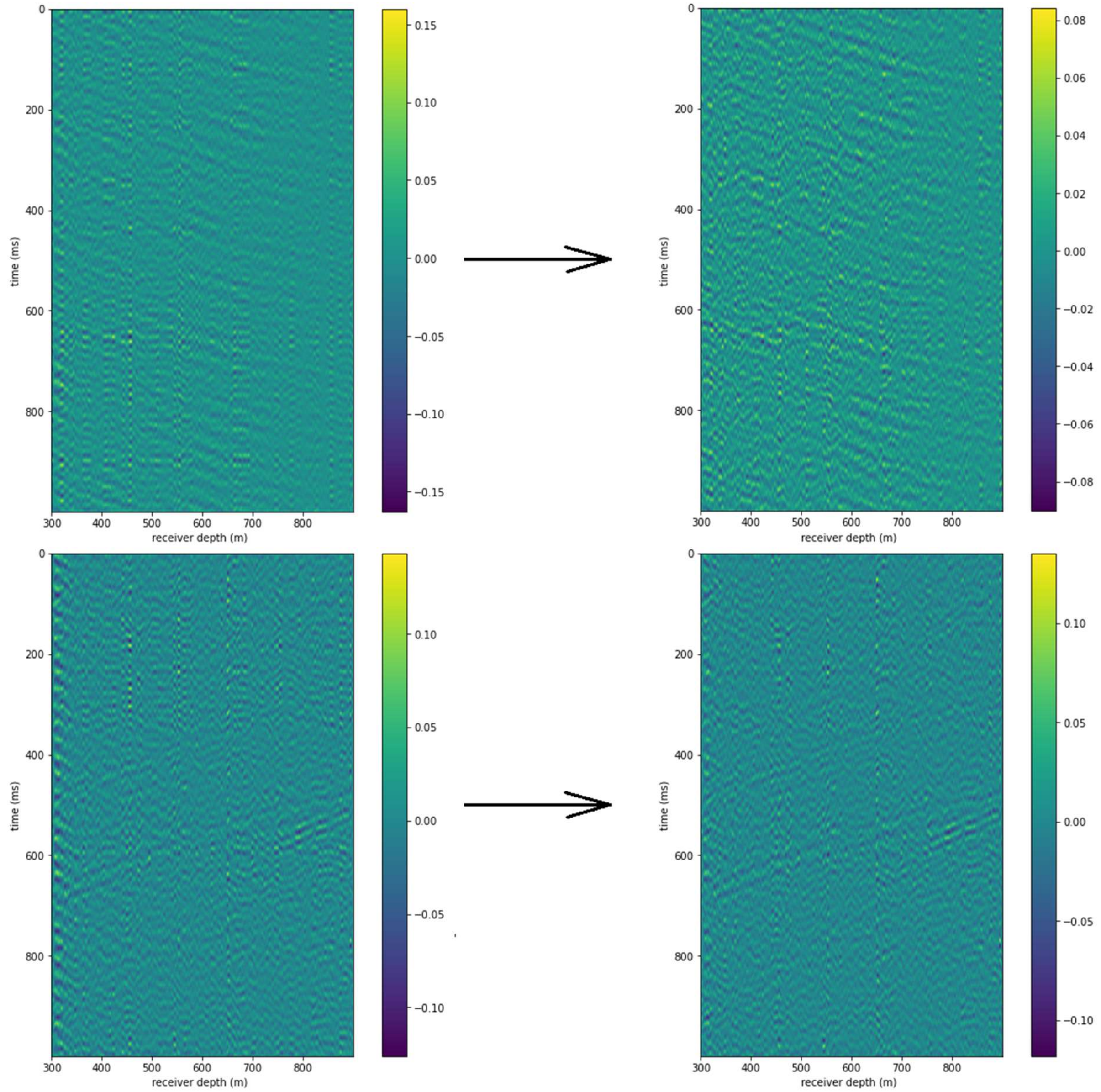

Figure 2.10. Rightmost and leftmost plot from Figure 2.9, before and after the added step in pre-processing.

Conclusion

I have introduced a new DAS event detection method based on shifted-matrix decomposition (SMD) that can be applied in real time. Advantages of the method are rapid computation time, and the fact that it does not require specific knowledge of the velocity model surrounding the receivers. Compared to the more expensive semblance-based detection method, my scheme detected fewer seismic events. The majority of the events that SMD-based detection failed to pick up could not be visually identified when looking at the processed data. This indicates that the differences in the sets of detected events are largely be due to different pre-processing sequence that was done to prior to applying the detection methods.

References

[1] al Irsyad, Muhammad Indra, Anthony Halog, and Rabindra Nepal. "Renewable energy projections for climate change mitigation: An analysis of uncertainty and errors." *Renewable energy* 130 (2019): 536-546.
[2] Dulal, Hari Bansha, Kalim U. Shah, Chandan Sapkota, Gengaiah Uma, and Bibek R. Kandel. "Renewable energy diffusion in Asia: Can it happen without government support?" *Energy Policy* 59 (2013): 301-311.

[3] Luderer, Gunnar, Volker Krey, Katherine Calvin, James Merrick, Silvana Mima, Robert Pietzcker, Jasper Van Vliet, and Kenichi Wada. "The role of renewable energy in climate stabilization: results from the EMF27 scenarios." *Climatic change* 123, no. 3 (2014): 427-441.

[4] Hayat, Muhammad Badar, Danish Ali, Keitumetse Cathrine Monyake, Lana Alagha, and Niaz Ahmed. "Solar energy—A look into power generation, challenges, and a solar-powered future." *International Journal of Energy Research* 43, no. 3 (2019): 1049-1067.

[5] DeCastro, M., S. Salvador, M. Gómez-Gesteira, X. Costoya, D. Carvalho, F. J. Sanz-Larruga, and L. Gimeno. "Europe, China and the United States: Three different approaches to the development of offshore wind energy." *Renewable and Sustainable Energy Reviews* 109 (2019): 55-70.

[6] Kagel, Alyssa, Diana Bates, and Karl Gawell. "A guide to geothermal energy and the environment." (2005).

[7] Olasolo, P., M. C. Juárez, M. P. Morales, and I. A. Liarte. "Enhanced geothermal systems (EGS): A review." *Renewable and Sustainable Energy Reviews* 56 (2016): 133-144.

[8] Li, Xinyang, Jimmy Zhang, Marcel Grubert, Carson Laing, Andres Chavarria, Steve Cole, and Yassine Oukaci. "Distributed acoustic and temperature sensing applications for hydraulic fracture diagnostics." In *SPE Hydraulic Fracturing Technology Conference and Exhibition*. OnePetro, 2020.

[9] Lellouch, Ariel, Nathaniel J. Lindsey, William L. Ellsworth, and Biondo L. Biondi. "Comparison between distributed acoustic sensing and geophones: Downhole microseismic monitoring of the FORGE geothermal experiment." *Seismological Society of America* 91, no. 6 (2020): 3256-3268.

[10] Binder, Gary, and Zagid Abatchev. "Joint microseismic event location with surface geophones and downhole DAS at the FORGE geothermal site." In *First International Meeting for Applied Geoscience & Energy*, pp. 2001-2005. Society of Exploration Geophysicists, 2021.

[11] Brankovic, Milan, Eduardo Gildin, Richard L. Gibson, and Mark E. Everett. "A Machine Learning-Based Seismic Data Compression and Interpretation Using a Novel Shifted-Matrix Decomposition Algorithm." *Applied Sciences* 11, no. 11 (2021): 4874.

[12] Moore, Joseph, John McLennan, Kristine Pankow, Stuart Simmons, Robert Podgorney, Philip Wannamaker, Clay Jones, William Rickard, and Pengju Xing. "The Utah Frontier Observatory for Research in Geothermal Energy (FORGE): a laboratory for characterizing, creating, and sustaining enhanced Geothermal systems." In *Proceedings of the 45th Workshop on Geothermal Reservoir Engineering*. Stanford University, 2020.

[13] Dzubay, Alex, Maria Mesimeri, Katherine M. Whidden, Daniel Wells, and Kris Pankow. "Developing a comprehensive seismic catalog using a matched-filter detector during a 2019 stimulation at Utah FORGE."

[14] Lellouch, Ariel, Siyuan Yuan, William L. Ellsworth, and Biondo Biondi. "Velocity-Based Earthquake Detection Using Downhole Distributed Acoustic Sensing—Examples from the San Andreas Fault Observatory at DepthVelocity-Based Earthquake Detection Using Downhole Distributed Acoustic Sensing." *Bulletin of the Seismological Society of America* 109, no. 6 (2019): 2491-2500.

CHAPTER IV

A METHOD FOR MODELING ACOUSTIC WAVES IN MOVING SUBDOMAINS

Introduction

The new simulation method presented in this manuscript is largely motivated by the recent developments in the data acquisition technology. Specifically, it is motivated by the development of distributed acoustic sensing (DAS), which uses fiber optic cables to record both low frequency strain and high frequency seismic waves [1]. DAS provides a new and unique view of the reservoir by sampling cable strain at rapid cadence and at densely spaced locations. High recording rates at numerous receiver positions increase the amount of registered microseismic activity. While it was originally developed for geophysical exploration, DAS has recently seen increasing use in other fields of geophysics. Distributed acoustic sensing has been coupled to existing submarine cables to monitor ground motion signals from seismic events and identify fault zones [2]. Because of its unprecedented spatial and temporal resolutions, DAS is expected to see further use in earthquake monitoring, imaging of faults and other geologic structures, and natural hazard assessments [3]. In conclusion, DAS records data at high frequency and over a long range of densely spaced locations. Furthermore, it can turn fiber-optic cables, which were initially intended for other purposes, into large collectors of seismic data. While DAS is excellent for gathering seismic data, it also has the potential to drastically increase the amount of seismic data recorded in the future.

To prepare for the future increase in the volumes of seismic data, we developed an algorithm to decrease the computational cost of forward wave modeling, which will speed up the processing and analysis of these data. The initial application, presented in this manuscript, is to acoustic waves, modeled by the acoustic wave equation in two dimensional domains, but the algorithm can be extended to three-dimensional models. The appropriate method for modeling waves depends on the purpose of modeling, the size and properties of the modeling domain, and the available computer resources. There does not exist an ideal finite difference method that can be used in every situation. For example, Zhou et al. [4] show improvements in accuracy via optimization that allows a reduction in the length of the FDM operator. Here, we take an alternate approach for optimizing scalar wave equation simulations. We develop an algorithm that can be used with any finite difference method that utilizes pre-defined finite difference operators and any model discretization regardless of the grid-point distribution. Specifically, the algorithm allows the user to calculate the pressure in only a subset of grid points in the modeling domain through which waves are propagating. Therefore, the numbers of grid points and physical degrees of freedom are reduced, while the grid-point spacing remains the same. Therefore, the numbers of grid points and physical degrees of freedom are reduced, while the gridpoint spacing remains the same. However, if the physical nature of the problem is such that active waves are propagating over the entire domain, with no quiet areas, such as in [5], then the RDM method loses its principal advantage.

This is not the first study that aims to speed up a finite difference scheme by modeling waves in only a subset of all the grid points, i.e., in a moving subdomain. Initially, Boore [6] noted that the displacement does not need to be computed in the areas which the first arrival has not yet reached. This idea was further developed when Vidale [7] used an eikonal equation to calculate the arrival times of waves at each grid point and then modeled the evolution of the wave at each grid point for a predetermined amount of time after the arrival. The drawback of this method is that it is focused on modeling only the head waves. There have also been studies published that model the propagation of seismic waves in moving zones (or boxes [8–10]. The path of the box shaped moving zone is pre-defined, and the box represents a subset of the entire modeling domain that is focused on the waves of interest (which are often the head waves). By restricting the modeling to the box enclosing the wave of interest, reflections outside the zone of interest are neglected. In both methods discussed above, the constraints to the modeling subdomain that provide the computational speed-up also restrict the applicability of the method.

In this work we introduce a new flexible approach for selecting the subset of grid points on which the wave is modeled. This method allows for the modeling of reflected waves even if they are far from the first-arriving wavefront. At the neglected, or irrelevant, grid points, disturbances caused by the waves should be small, or even non-existent, depending on the application and user-defined parameters. Because the purpose of our method is to reduce the number of grid points in the domain at which the pressure is calculated, we refer to it as the “reduced domain method” or RDM. By defining certain parameters in RDM, the user may adjust the criterion which differentiates between relevant and irrelevant grid points. Because of this, while the performance may vary, the algorithm can be useful in a large variety of scenarios of wave propagation.

It should also be noted that the most recent application of a moving subdomain, or a moving frame to be more accurate, was for modeling acoustic waves propagating through the earth’s atmosphere using the Navier–Stokes equations. The numerical simulations have been performed in two dimensions on Cartesian grids [11], in a two dimensional cylindrical coordinate system with assumed axial symmetry [12,13], and in full three dimensions [14]. While the algorithm developed in our research is implemented for the acoustic wave equation, with additional programming effort it can also be applied to FDMs used for modeling the elastic wave equation as well as the Navier–Stokes equations. This is because the reduced domain method is designed to be applied to any

FDM and velocity model, regardless of the grid-point distribution as long as the FDM has pre-defined operators. For simplicity, we will refer to such methods as standard FDMs.

With additional programming effort, the RDM may also be adapted to some methods even if the finite difference operators are not pre-determined. A good example of this is [15], where acoustic waves are modeled while the coefficients in the finite difference operators adaptively change. However, there are some methods such as [16], in which the acoustic waves are modeled by finite difference operators which change length adaptively during the simulation. For such methods, the implementation of RDM becomes more difficult. To summarize, RDM provides a reduction in computational cost by modeling the waves in only a subset of the entire domain and it can be applied to any velocity model and a variety of FDM. However, thanks to its flexible and adaptive selections of subdomains during the simulation time, it also allows the user to accurately observe the majority of the wavefield. This is an improvement compared to previous methods that modeled waves in moving subdomains.

Methodology

RDM is a method that first determines the “active” portion of the modeling domain, i.e., the zone within which waves are propagating. The method then uses the selected finite difference method to simulate wave propagation within the active area. Further details are given below.

In finite difference schemes, the pressure field is described by a vector p whose number of elements equals the number of grid points in the modeling domain. We seek to reduce the length of this vector and refer to the new, smaller vector as the ”reduced” pressure vector, or simply the reduced vector p_r. The elements in the reduced vector at a given time step comprise only the pressure at those grid points through which a wave is actively propagating. We refer to these grid points as ”relevant” grid points. As waves propagate through the modeling domain, the set of relevant grid points changes. To find the reduced vector at each time step, RDM determines the set of relevant grid points without actually evaluating the pressure at all the grid points. The following paragraphs and Figure 3.1 below explain how RDM achieves this goal.

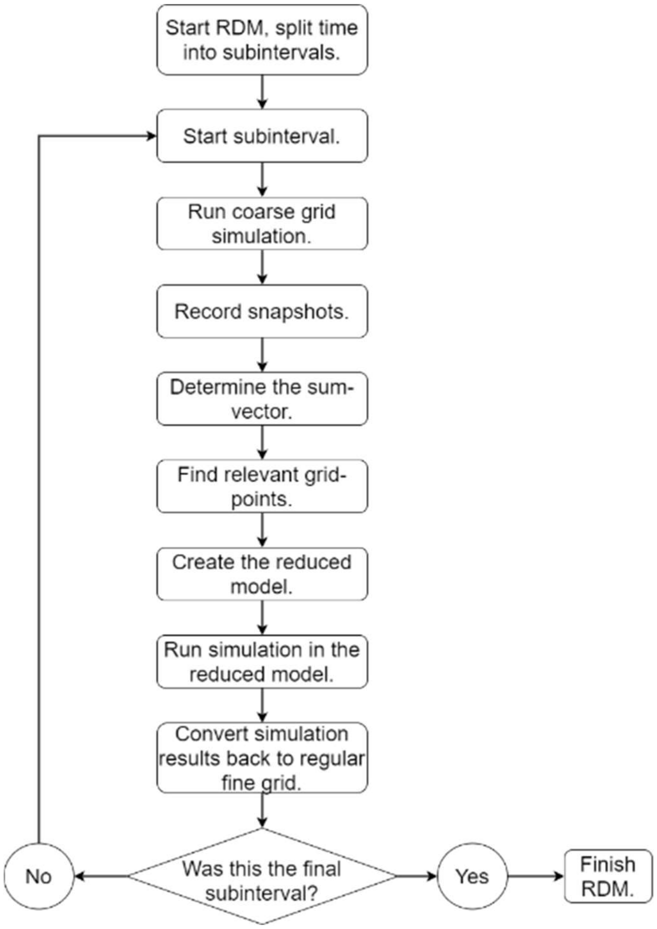
Start RDM, split time into subintervals.
Start subinterval.
Run coarse grid simulation.
Record snapshots.
Determine the sum-vector.
Find relevant grid-points.
Create the reduced model.
Run simulation in the reduced model.
Convert simulation results back to regular fine grid.
Was this the final subinterval?
No
Yes
Finish RDM.

Figure 3.1. A flowchart providing a high-level description of RDM.

The RDM algorithm starts by dividing the time during which the wave propagation is simulated into subintervals of length T s . The subinterval length T s needs to be small in order to keep the set of relevant grid points within subintervals small. On the other hand, decreasing the subinterval length T s will increase the number of subintervals in the simulation and the time spent finding the set of relevant grid points for all subintervals could start having a large effect on computation time. The best value for T s depends on the velocity model and the source, and there does not exist an ideal T s value that gives the best results in every scenario. However, you can still run consistently fast and accurate simulations while always using the same value of T s . We set T s to be equal to the period of the dominant frequency of the source, and as can be seen in the following section, RDM drastically reduced computation time and maintained accuracy in all of the tested models.

At the start of each subinterval, prior to advancing the simulation using the reduced vector, RDM runs a fast simulation on a coarse grid that spans the entire modeling domain. The grid-point spacing, and the time step of the coarse-grid simulation are set to be twice as large as those in the standard grid, or rather, fine-grid simulation. The FDM that is used to run the simulation in the coarse grid is the same as the FDM used in the fine-grid simulation. For future improvement of RDM one could consider using a different FDM for the coarse-grid simulation, which allows the use of fewer grid points and reduces the computation time. However, on average only 25–30% of RDM computation time is spent in the coarse-grid simulation, so the potential for the computation reduction is limited. The simulation on the coarse grid is not intended to yield a highly accurate displacement field, but it is detailed enough to avoid excessive numerical dispersion and allow a sufficiently accurate determination of the relevant grid points.

The relevant grid points are determined by first defining a “sum vector” ***v*** sum. Each component of the sum vector is a time summation, over the current subinterval, of the squared time derivatives of the corresponding component of the pressure field on the coarse grid:

$$v_i^{sum} = \sum_{j=1}^{N} \left(\frac{\partial p_i^c\left(t^0+(j-1)t^{snp}\right)}{\partial t}\right)^2 \tag{3.1}$$

where $\boldsymbol{p}^c$ is the pressure vector computed in the coarse-grid simulation, N is the number of snapshots in a subinterval, and $\boldsymbol{t}^{snp}$ is the time between two subsequent snapshots. The magnitude of the sum vector for a given subinterval, at a given grid point, is large if the corresponding pressure on that grid point was large during the subinterval.

The values of N and t^{snp} depend on the length of the subinterval T^s and the time step of the finite difference scheme. Although N and t^{snp} need not have specific values, typically we set t^{snp} such that $40 \geq N \geq 20$. The goal is to use enough snapshots to accurately describe the wavefield during the interval, while also not using so many snapshots as to affect the computation time. It should be noted that changing N and t^{snp} has very little effect on the performance of the simulation. Thus, we do not think that optimizing those parameters can lead to noticeable improvements.

Before the values recorded in $\boldsymbol{v}^{sum}$ are used to estimate the map of relevant grid points, an equal weight averaging filter is applied to $\boldsymbol{v}^{sum}$. Filtering is performed to smooth the results stored in the sum vector, which reduces the length of the bounding curve between the relevant and irrelevant grid points, as can be seen in Figure 3.2. A big part of the error caused by using RDM is produced at the boundary between relevant and irrelevant grid points. Having wavefield drop from near-zero values to zero can create a source of error in the wavefield. By reducing the length of the boundary between relevant and irrelevant grid points, we reduce the error. This allows us to reduce the computation time more aggressively, while still maintaining a small error. The averaging filter is two dimensional and 32 grid points wide and long. This is because its length is defined as four times the shortest wavelength in the model, i.e., four times the period of the dominant frequency multiplied by the velocity from the slowest area in the model. The processing time of the averaging filter is proportional to the length of the filter and the number of grid points used in the simulation. As RDM was tested on multiple models, the computation time of the averaging filter varied between 3% and 7% of the entire simulation time when using RDM. To further reduce the computation time of the averaging filter, we could use better picks for the window size that are based on the velocity average rather than minimum velocity, and we could also apply the filter to a different vector that has fewer elements than v sum, for example, a vector containing elements of $\boldsymbol{v}^{sum}$ for grid points with spacing four times as large as that of the fine grid. Once the sum vector $\boldsymbol{v}^{sum}$ is determined, the set of relevant grid points is constructed.

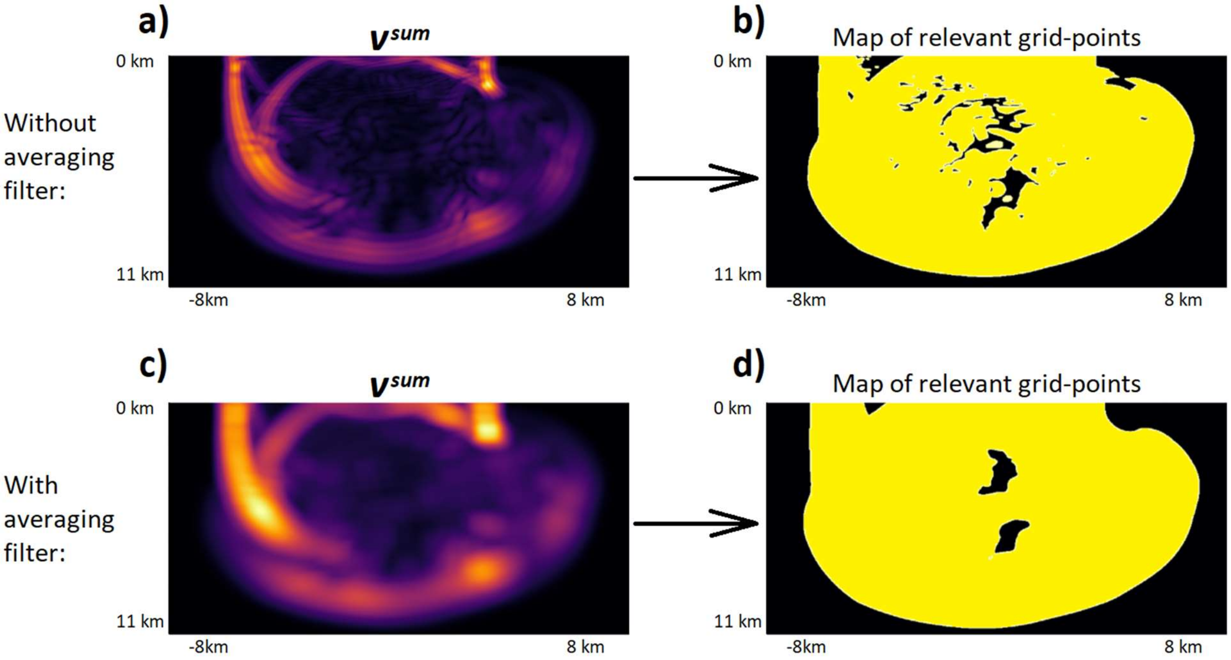

Figure 3.2. (**a**) The sum vector v sum without the averaging filter and (**b**) the resulting map of relevant grid points. (**c**) The sum vector v sum with the averaging filter and (**d**) the resulting map of relevant grid points.

The set of relevant grid points is defined as the smallest subset $\boldsymbol{U}$ of all the grid points such that the sum of elements in the sum vector $\boldsymbol{v}^{sum}$ representing those grid points is greater than or equal to some pre-determined threshold fraction $(1-e^{-\delta})$ of the sum of all elements in the sum vector:

$$\min\big(n(\boldsymbol{U})\big):\ \sum_{j=1}^{n(\boldsymbol{U})} v_{(U_j)}^{sum} \geq (1-e^{-\delta}) \sum_{j=1}^{n(\boldsymbol{v}^{sum})} v_j^{sum} \qquad (3.2)$$

where $n(\boldsymbol{U})$ is the number of elements in $\boldsymbol{U}$, $n(\boldsymbol{v}^{sum})$ is the number of elements in $\boldsymbol{v}^{sum}$, and the threshold $(1-e^{-\delta})$ is defined by the parameter δ. The threshold is defined in this way so that an increase in δ causes the threshold to increase, converging closer to the value of 1. An increase in the threshold results in more grid points being included in the set of relevant grid points. Therefore, increasing the parameter δ increases the accuracy of RDM but also increases the computation time.

While the parameters T^s, t^{snp} and N should remain the same in all simulations, the parameter δ can be adjusted to best support the FDM we are applying our method to, which is the purpose of the simulation. For example, if we are doing reverse time migration (RTM) for the purpose of locating a seismic event, we can ignore a lot of weak waves that we know will not contribute to the convergence at the source location. In this case we could set up δ to a small value that will result in fewer relevant grid points and faster simulation. Alternatively, if the user is interested in simulating weak reflection, the parameter δ would be set up to a larger value to make sure the weak reflections are represented.

When using Equation (3.2), RDM determines the set of relevant grid points based on the amplitudes of the propagating waves. This method allows the user to observe the large majority of the wavefield while reducing the computation time. In a more specific example, the user might have a special interest in reflections in a given area of the model, even if the reflections are weak. In such a case, Equation (3.2) can be modified so that the summations include weights for each grid point. This way, we can add extra importance to grid points from a specific area. Therefore, depending on the specific purpose of the wave simulations in the future, the parameter δ and Equation (3.2) may be adjusted and modified.

Once the set of relevant grid points has been determined, the next step is to create the reduced model. The set of relevant grid points ($\boldsymbol{U}$) is applied to the vectors describing the pressure from the two latest subsequent time steps ($\boldsymbol{p}, \boldsymbol{p}^o$) and velocity ($\boldsymbol{c}$) on the fine grid. Specifically, the set of relevant grid points tells us which elements in pressure (or velocity) vectors represent the pressure (or velocity) on the relevant grid points. To generate reduced pressure and velocity vectors, an algorithm goes through all the elements of the pressure vectors $\boldsymbol{p}$, $\boldsymbol{p}^o$, and velocity vector $\boldsymbol{c}$ and the values describing the pressure or velocity at relevant grid points are recorded in reduced pressure and velocity vectors $\boldsymbol{p}^r$, $\boldsymbol{p}^{or}$, and $\boldsymbol{c}^r$. The set of vectors converted to the reduced model may vary between different models and different simulation methods. For example, in cases with heterogeneous density, we also must apply the set of relevant grid points to the density vector ($\boldsymbol{\rho}$) in order to generate the reduced density vector $\boldsymbol{\rho}^r$.

At the next stage of the algorithm, the fine-grid simulation is executed over the subinterval on the set of relevant grid points. The computation is executed on $\boldsymbol{p}^r$ and $\boldsymbol{p}^{or}$ rather than on $\boldsymbol{p}$ and $\boldsymbol{p}^o$, creating a significant reduction in computation time. Once the fine-grid simulation reaches the end of the subinterval, the values of the pressure on the standard fine grid $\boldsymbol{p}$ and $\boldsymbol{p}^o$ are updated using the reduced vectors $\boldsymbol{p}^r$ and $\boldsymbol{p}^{or}$ and the set of relevant grid points $\boldsymbol{U}$. The process is repeated until the simulation reaches the end of the final subinterval.

In the introduction we stated that RDM is used to reduce the cost of modeling the acoustic wave equation. The specific FDM that RDM is applied to is the one by Alford et al. [17], also described in Zakaria et al. [18], which was chosen for its efficiency and simple implementation. Here, the second time derivative of pressure, $\frac{\partial^2 \boldsymbol{p}}{\partial t^2}$, is estimated with a fourth-order accurate nine-point stencil:

$$\frac{\partial^2 p_{i,j}}{\partial t^2} = \left(\left(p_{i-1,j} + p_{i+1.j} + p_{i,j-1} + p_{i,j+1} \right) \frac{4}{3} - \left(p_{i-2,j} + p_{i+2,j} + p_{i,j-2} + p_{i,j+2} \right) \frac{1}{12} - 5p_{i,,j} \right) \frac{c_{i,j}^2}{\Delta x^2} \tag{3.3}$$

where Δx is the spacing between grid points. Furthermore, the finite difference scheme can be adapted to heterogeneous density models by altering the finite difference coefficients:

$$\frac{\partial^2 p_{i,j}}{\partial t^2} = \Big(\Big(\big(2-\rho^x_{i,j}\big)p_{i-1,j} + \big(2+\rho^x_{i,j}\big)p_{i+1,j} + \big(2-\rho^y_{i,j}\big)p_{i,j-1} + \big(2+\rho^y_{i,j}\big)p_{i,j+1}\Big)\frac{2}{3} - \Big(\big(1-\rho^x_{i,j}\big)p_{i-2,j} + \big(1+\rho^x_{i,j}\big)p_{i+2,j} + \big(1-\rho^y_{i,j}\big)p_{i,j-2} + \big(1+\rho^y_{i,j}\big)p_{i,j+2}\Big)\frac{1}{12} - 5p_{i,j}\Big)\frac{c^2_{i,j}}{\Delta x^2} \qquad (3.4)$$

where:

$$\rho^x_{i,j} = \rho_{i,j}\left(\frac{1}{\rho_{i-2,j}} - \frac{8}{\rho_{i-1,j}} + \frac{8}{\rho_{i+1,j}} - \frac{1}{\rho_{i+2,j}}\right)\frac{1}{12}$$

$$\rho^y_{i,j} = \rho_{i,j}\left(\frac{1}{\rho_{i,j-2}} - \frac{8}{\rho_{i,j-1}} + \frac{8}{\rho_{i,j+1}} - \frac{1}{\rho_{i,j+2}}\right)\frac{1}{12}. \qquad (3.5)$$

Equations (3.4) and (3.5) above, which are applied to inhomogeneous density models, can be derived from the first equation from [19]. It is important to point out that Equations (3.3-3.5) do not provide a perfectly accurate representation of the processes in RDM. Specifically, the pressure, velocity, and density are all recorded as vectors in RDM whereas in Equations (3.3–3.5) they are presented as matrices. We made this decision because we wanted to provide a more clear and easy-to-read representation of the finite difference operators being used.

In the fine-grid simulations we set the grid-point spacing to be sixteen times smaller than the period of the dominant frequency of the source multiplied by the velocity in the slowest area in the model. This way, there is never fewer than sixteen grid points per wavelength, or eight grid points per wavelength in the coarse-grid simulation. Once the second derivative of the pressure field $\boldsymbol{p}$ is calculated, a second-order accurate scheme uses the current pressure $\boldsymbol{p}$ and the pressure from the previous time step $\boldsymbol{p}^o$ to calculate the pressure field at next time step $\boldsymbol{p}^n$:

$$\boldsymbol{p}^n = 2\boldsymbol{p} - \boldsymbol{p}^o + \Delta t^2 \frac{\partial^2 \boldsymbol{p}}{\partial t^2} \qquad (3.6)$$

where Δt is the time-step size of the simulation. To maintain stability of the simulation, the size of the time steps is set to be one half of the grid-point spacing divided by the maximum wave velocity in the model. Therefore, the number of time steps per wave period is dependent on the ratio between the velocities in the slowest and the fastest regions in the model. However, if the model were homogeneous, there would be 32 time-steps in the period of the dominant frequency.

Results

The RDM algorithm was tested on four synthetic models. The first two models comprise scenarios in which a steel object is buried partially or completely beneath the seafloor. They are used to demonstrate the performance of the algorithm and illustrate how the map of relevant grid points progresses along with the waves during a simulation. All calculations in this manuscript are fully 2-D, i.e., an infinite line source excites an infinite structure invariant along strike, the direction parallel to the source. The run time of RDM is compared to the run time of the standard FDM without RDM. The relative error is defined as the change to the final pressure vector resulting from RDM application, specifically:

$$E_{rel} = \frac{||\boldsymbol{p} - \boldsymbol{p}^{RDM}||}{||\boldsymbol{p}||} \tag{3.7}$$

where $\boldsymbol{p}$ is the pressure vector obtained from standard FDM and $\boldsymbol{p}^{RDM}$ is the pressure vector obtained by applying RDM to the same FDM.

The code was written in Julia programming language, which is designed for rapid execution of numerical simulations. We chose Julia because it can conveniently optimize functions and implement vectorization, thereby reducing computation time. Furthermore, the simulations are executed on a single core of Intel i7-6820HQ which has a base frequency of 2.70 GHz and 16 GB of RAM. Adjusting the code to run on multiple cores would require more programming, but the algorithm is inherently parallelizable.

The first two models are energized by a Ricker source wavelet with peak frequency 2.3 kHz placed 8.4 m above the ocean floor at a midpoint between the two lateral boundaries. In the first model the steel object is not completely buried in the surrounding limestone ocean floor. The velocity and density values used for limestone in this model were obtained from Table 1 of Bayer [20]. In the second model the entire steel object is buried in sediment. The velocity and density values used for sediment were obtained from Hamilton [21]. If a grid point is positioned on the interface of the two layers, an arithmetic average is used to determine the velocity and density at said grid point. These two models are shown in Figure 3.3, along with the evolution of the map of the relevant grid points throughout the simulation and the final wavefield in each of the two models.

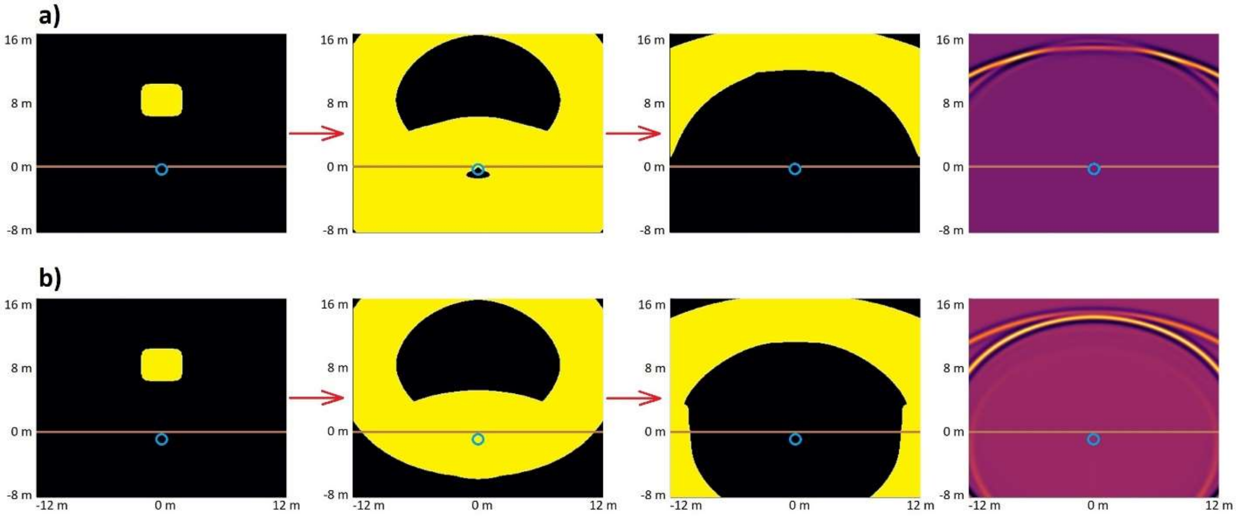

Figure 3.3. (**a**) Results for the first limestone model. (**b**) Results for the second sediment model. From left to right we see: the map of relevant grid points in the first subinterval, nineteenth subinterval, and thirty-seventh subinterval, and finally, the wavefield corresponding to the end of the thirty-seventh subinterval. The figures were acquired during the simulation, with δ set to 12. The relevant grid points are in the yellow region. The brown line represents the surface of the ocean floor, and the blue line represents the steel object.

In both models, the wave propagation is simulated over a period of 0.016 s, and it took about 255 s with standard FDM to run the simulation, which contained 4515 time-steps on a 501 by 501 grid. During the simulations, the waves propagate to the steel object and are reflected past their point of origin. We stated above that the parameter δ affects the size of the set of relevant grid points, such that an increase in δ causes an increase in both accuracy and run time. Table 3.1

displays the computation time reduction and relative error of RDM for various value choices of δ for the two models.

Table 3.1. Performance indicators for the first two models.

	First Model		Second Model	
Parameter $\boldsymbol{\delta}$	**Comp. Time Reduction (%)**	**Relative Error**	**Comp. Time Reduction (%)**	**Relative Error**
10	56.5	0.017	63.0	0.062
20	54.8	0.016	61.8	0.043
30	48.0	0.014	56.9	0.026
40	40.5	0.015	49.1	0.033
50	40.1	0.005	50.2	0.020
60	40.1	0.001	47.8	0.005

In the third test we use the velocity model from [22] but assuming a spatially uniform density. The purpose of the modeling is to test RDM on a more heterogeneous velocity model in which many reflections and dispersions are generated. A Ricker source with peak frequency 7.1 Hz is located at the center of the upper boundary of the model domain. The wave is propagated in the simulation for 10.1 s, until it reaches the lateral boundaries of the model domain. The computation time of the standard FDM simulation, which contained 7764 time-steps on a 5795 by 1155 grid, was about 9420 s. The velocity model (top), the map of relevant grid points (middle), and the wavefield (bottom) at the end of the simulation are shown in Figure 3.4. The performance of RDM for different choices of δ is presented in Table 3.2.

In the fourth test we combine the velocity and density models from [22] with the same source and simulation time as in the third test. Here, the computation time of the standard FDM simulation, which contained 7764 time-steps on a 5795 by 1155 grid was about 10,210 s. The density is strongly heterogeneous which produces a great number of reflections such that every grid point in the model domain is populated with strong coda after the passage of the first-arriving wave. For as long as the strong coda remains, all the grid points through which a head wave has passed would be considered relevant grid points. Thus, we add a new criterion to the selection of grid points that forces RDM to neglect low-amplitude waves. The objective is to assess the accuracy by which RDM models the high-amplitude waves while neglecting weaker ones.

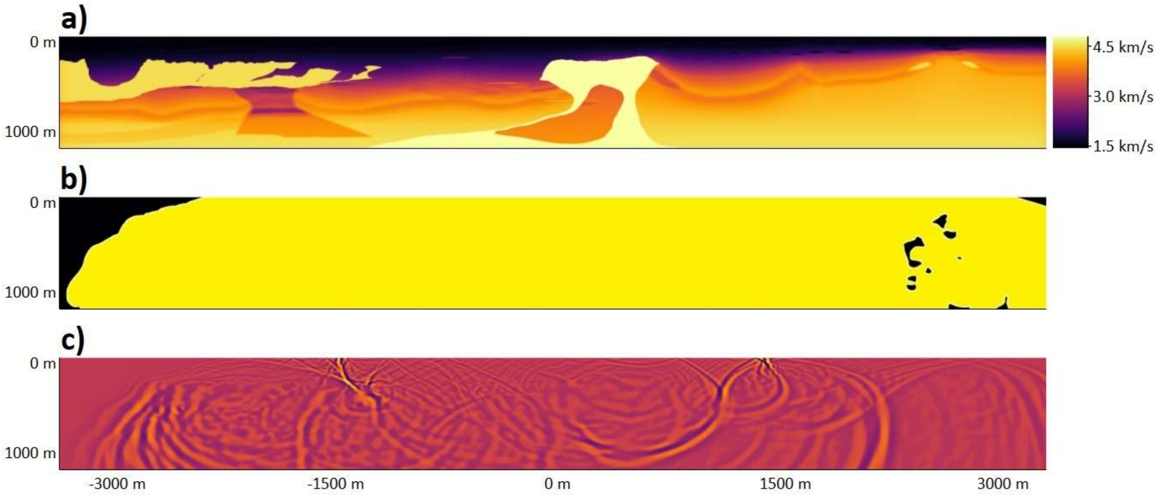

Figure 3.4. (**a**) The velocity model from [22]. (**b**) The map of relevant grid points in the final subinterval. (**c**) The wavefield at the final subinterval of the simulation. The parameter δ was set to 12.

Table 3.2. Performance indicators for the third model.

Parameter $\boldsymbol{\delta}$	Comp. Time Reduction (%)	Relative Error
10	71.3	0.026
20	67.9	0.011
30	66.9	0.003
40	68.8	6.0×10^{-4}
50	65.9	2.7×10^{-4}
60	65.8	6.5×10^{-5}

The new criterion is set by fixing the parameter δ to 20 and adding an extra condition that the number of relevant grid points may not be larger than the total number of grid points multiplied by some fraction θ. This is enforced by adding a new criterion to Equation (3.2) that states $n(\boldsymbol{U}) \leq \theta n(\boldsymbol{p})$. This criterion is designed to maintain or ensure a low computation time, wherein the weaker waves are presumably negligible, e.g., below the sensitivity of the recording instruments. The relative error will also be calculated in the zones enclosing the high-amplitude waves, which are presented in Figure 3.5, along with the density model (top), the map of relevant grid points (middle), and the wavefield (bottom). The performance of RDM on the fourth model is presented in Table 3.3 below.

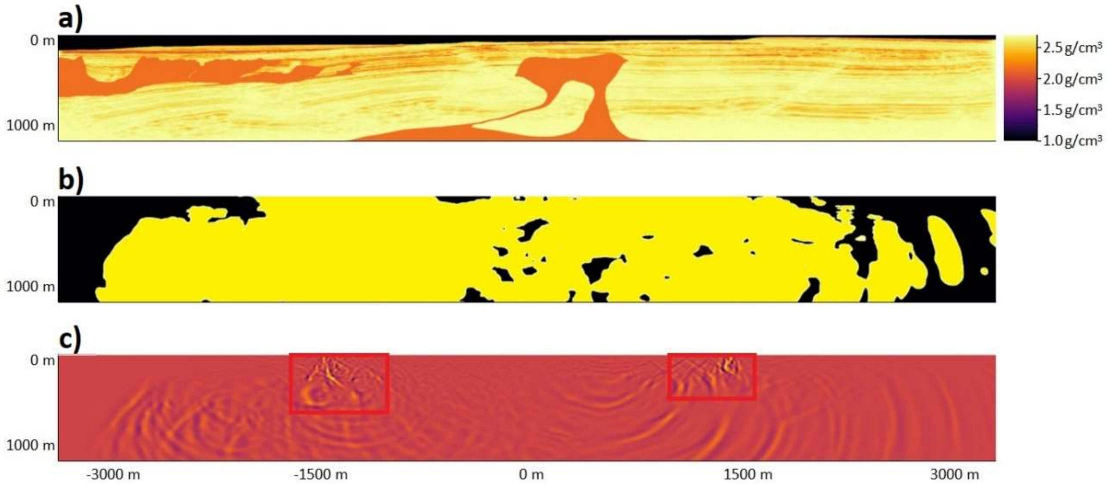

Figure 3.5. (**a**) The density model from [22]. (**b**) The map of relevant grid points at the final subinterval of the simulation with parameter θ set to 0.7. (**c**) The wavefield at the end of the standard FDM simulation with the strong waves marked with red squares.

Table 3.3. Performance indicators for the fourth model.

Parameter θ	**Comp. Time Reduction (%)**	**Relative Error**	**Relative Error in the Area of Interest**
0.5	72.5	0.310	0.112
0.6	71.7	0.229	0.073
0.7	71.5	0.136	0.025
0.8	71.0	0.066	0.002
0.9	66.7	0.005	5.7×10^{-7}
1.0	66.7	5.4×10^{-5}	5.7×10^{-7}

Discussion

The data in Tables 3.1 and 3.2 show that the reduction in computation time from RDM ranges between 40 and 70%, depending on the modeling scenario and the value of the δ parameter. Even though the third model is more heterogeneous than the first two, the reduction in computation time is greater in the third test. This is because the first two modeling domains are much smaller, so the waves reach the boundaries earlier, and therefore the map of relevant grid points expands across the domain more rapidly than in the third model.

While relative error maintains low values in the first three tests, from several percent to 10^{-4}, in the fourth test it reaches 0.3. The large relative error in the fourth test occurs because an aggressive criterion is used to select relevant grid points, so that many of the weaker waves are not modeled. The goal in the fourth test is to efficiently yet accurately model the high-amplitude waves. This goal is achieved as the relative error in areas enclosing the strong waves (presented in Figure 3.5) is much smaller, as shown in Table 3.3.

As stated in the introduction, RDM can be applied to any FDM with pre-defined spatial operators. This means that RDM can also be applied to other, higher-order FDM. Higher-order FDM are generally used to reduce the number of grid points needed in the model, which reduces the memory requirements and can also lead to lower computation times. We expect the percent reduction in computation to remain the same as RDM is applied to various FDM. This is because we do not expect that changing the FDM to which RDM is applied would have a noticeable effect on the RDM's estimation of what portion of the domain contains irrelevant grid points. As long as there are parts of the domain with weak or non-existent waves, RDM can identify areas with irrelevant grid points in the simulation, and the computation time can be reduced. However, if the total number of grid points in the domain decreases as a result of the use of higher-order FDM, the perimeter of the areas of relevant and irrelevant grid points would become more coarse. This could cause the error due to RDM application to increase slightly. It should also be noted that the application of RDM does not create any new limitations on the frequency range of waves that can be modeled in the simulation. The frequency range is entirely dependent on the grid-point spacing, time-step size, and the FDM to which RDM is being applied.

The calculations in this manuscript are performed in 2-D models. The simulations can therefore be used to describe a line source and the response from a structure invariant along strike, which is the direction parallel to the source. In such a scenario, the line of receivers, possibly provided by DAS, can be oriented in any direction relative to the source.

There are several possibilities for future work. The RDM method can be adapted to more complex finite difference schemes or to 3-D applications. Currently, we are more focused on developing a 3-D version of RDM as it will have a greater impact on the range of applications of RDM. This will take time and additional programming effort, but we expect RDM to be able to maintain its significant reduction in computational cost in the three-dimensional simulations.

Conclusion

Wave modeling methods that allow the calculation of pressure in only a subset of grid points can provide an excellent reduction in computation time and complement a variety of finite difference schemes. The RDM algorithm developed herein can be applied to any FDM that uses pre-defined finite difference operators. The developed method uses an adaptively changing subset of all the grid points to accurately model both first-arriving and reflected waves. As a result, a high level of accuracy is obtained while reducing computation time by more than 50%. The reduced domain method was tested on simple models describing the ocean floor with a buried steel object, which were used to demonstrate how the map of relevant grid points changes throughout the simulation, and also on more complex and realistic models which contained many layers and substantial heterogeneity.

While RDM is valuable in most forward modeling scenarios, we expect RDM to be the most useful in inverse problems wherein many forward modeling runs are required. Running repeated forward models can take a lot of time. Here, the computation cost reduction to forward modeling obtained with RDM could be of great value. Furthermore, RDM may also be very useful when studying a seismic source with reverse time migration (RTM), where we are only concerned with the waves converging at the source. Because a big portion of the waves in RTM does not converge at the source, we could apply RTM aggressively to drastically reduce computation time, while having little effect on the accuracy of the results.

References

[1] Li, Xinyang, Jimmy Zhang, Marcel Grubert, Carson Laing, Andres Chavarria, Steve Cole, and Yassine Oukaci. "Distributed acoustic and temperature sensing applications for hydraulic fracture diagnostics." In *SPE Hydraulic Fracturing Technology Conference and Exhibition*. OnePetro, 2020.

[2] Lindsey, Nathaniel J., T. Craig Dawe, and Jonathan B. Ajo-Franklin. "Illuminating seafloor faults and ocean dynamics with dark fiber distributed acoustic sensing." *Science* 366, no. 6469 (2019): 1103-1107.

[3] Zhan, Zhongwen. "Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas." *Seismological Research Letters* 91, no. 1 (2020): 1-15.

[4] Zhou, Hongyu, Yang Liu, and Jing Wang. "Optimizing orthogonal-octahedron finite-difference scheme for 3D acoustic wave modeling by combination of Taylor-series expansion and Remez exchange method." *Exploration Geophysics* 52, no. 3 (2021): 335-355.

[5] Thongchom, Chanachai, Pouyan Roodgar Saffari, Nima Refahati, Peyman Roudgar Saffari, Hossein Pourbashash, Sayan Sirimontree, and Suraparb Keawsawasvong. "An analytical study of sound transmission loss of functionally graded sandwich cylindrical nanoshell integrated with piezoelectric layers." *Scientific Reports* 12, no. 1 (2022): 1-16.

[6] Boore, David M. "Finite difference methods for seismic wave propagation in heterogeneous materials." *Methods in computational physics* 11 (1972): 1-37.

[7] Vidale, John. "Finite-difference calculation of travel times." *Bulletin of the seismological society of America* 78, no. 6 (1988): 2062-2076.

[8] Hansen, Thomas Mejer, and Bo Holm Jacobsen. "Efficient finite difference waveform modeling of selected phases using a moving zone." *Computers & geosciences* 28, no. 7 (2002): 819-826.

[9] JafarGandomi, Arash, and Hiroshi Takenaka. "Non-standard FDTD for elastic wave simulation: two-dimensional P-SV case." *Geophysical Journal International* 178, no. 1 (2009): 282-302.

[10] Serdyukov, Alexandr S., and Anton A. Duchkov. "Hybrid kinematic-dynamic approach to seismic wave-equation modeling, imaging, and tomography." *Mathematical Problems in Engineering* 2015 (2015).

[11] Sabatini, R., O. Marsden, C. Bailly, and O. Gainville. "Numerical simulation of infrasound propagation in the Earth's atmosphere: study of a stratospherical arrival pair." In *AIP Conference Proceedings*, vol. 1685, no. 1, p. 090002. AIP Publishing LLC, 2015.

[12] Sabatini R, Snively JB, Hickey MP, Garrison JL. "An analysis of the atmospheric propagation of underground-explosion-generated infrasonic waves based on the equations of fluid dynamics: Ground recordings." In J Acoust Soc Am. 2019 Dec;146(6):4576. doi: 10.1121/1.5140449. PMID: 31893690.

[13] de Groot-Hedlin, C. D. "Long-range propagation of nonlinear infrasound waves through an absorbing atmosphere." *The Journal of the Acoustical Society of America* 139, no. 4 (2016): 1565-1577.

[14] Sabatini, Roberto, Olivier Marsden, Christophe Bailly, and Olaf Gainville. "Three-dimensional direct numerical simulation of infrasound propagation in the Earth's atmosphere." *Journal of Fluid Mechanics* 859 (2019): 754-789.

[15] Yao, Gang, Di Wu, and Henry Alexander Debens. "Adaptive finite difference for seismic wavefield modelling in acoustic media." *Scientific Reports* 6, no. 1 (2016): 1-10.

[16] Zhou, Hongyu, Yang Liu, and Jing Wang. "Finite-difference modeling with adaptive variable-length temporal and spatial operators." In *2018 SEG International Exposition and Annual Meeting*. OnePetro, 2018.

[17] Alford, R. M., K. R. Kelly, and D. Mt Boore. "Accuracy of finite-difference modeling of the acoustic wave equation." *Geophysics* 39, no. 6 (1974): 834-842.

[18] Zakaria, Ahmad, John Penrose, Frank Thomas, and Xiuming Wang. "The Two Dimensional Numerical Modeling Of Acoustic Wave Propagation in Shallow Water." In *Proceedings of the Australian Acoustical Society Conference, Joondalup, Australia*, pp. 15-17. 2000.

[19] Hicks, Graham J. "Arbitrary source and receiver positioning in finite-difference schemes using Kaiser windowed sinc functions." *Geophysics* 67, no. 1 (2002): 156-165.

[20] Bayer, Jacob Bartscht. "Structural Analysis of Bonaire, Netherlands Leeward Antilles-A Seismic Investigation." PhD diss., 2016.

[21] Hamilton, Edwin L. "Geoacoustic modeling of the sea floor." *The Journal of the Acoustical Society of America* 68, no. 5 (1980): 1313-1340.

[22] Cha, Young Ho, and Changsoo Shin. "Two-dimensional Laplace-domain waveform inversion using adaptive meshes: An experience of the 2004 BP velocity-analysis benchmark data set." *Geophysical Journal International* 182, no. 2 (2010): 865-879.

CHAPTER V

CONCLUSION

I have developed a seismic data decomposition algorithm that is specifically optimized for large datasets containing many traces. The algorithm was used for denoising, compressing and detecting events in seismic data. In a short illustrative example, its capability for estimating the velocities of P-waves was tested. I call this method shifted-matrix decomposition (SMD). The majority of data decomposition methods, including SMD, take a matrix containing seismic data as input. What makes SMD unique is its output, which was specifically designed to optimally describe seismic data. In fact, SMD was developed by first determining what kind of results I wanted from a data decomposition method applied to a seismic data matrix, and then designing an algorithm that can return such results.

To be more specific, the result of applying SMD to seismic data matrix is a series of sets of three vectors. The three vectors consist of a pair of basis vectors, also referred to as a pair of singular vectors, and a third vector I term the shift vector. In an ideal scenario, for a single seismic wave, the first basis vector describes the waveform, the second describes the amplitude at each receiver, and the shift vector describes the arrival time at each receiver. For this reason, the two basis vectors are termed the waveform vector and the amplitude vector. While the ideal case is rarely achieved, the set of three vectors usually captures a significant part of a recorded seismic wave. There are many advantages in storing seismic data this way, as outlined in Chapter II.

The first advantage is that it requires less memory to store seismic data as SMD output than as a matrix. When tested on marine seismic gathers, SMD results reduced the memory requirements of the data by 80% while capturing most of the coherent waves. This is because SMD is designed for, and very efficient at, capturing seismic signals. For the same reason, SMD is effective at denoising seismic data. Because SMD is much more efficient at capturing seismic signals than noise, reconstructing the original data from SMD results creates a new version with much less noise and better SNR. Examples of SMD enhancing weak reflections in marine seismic gathers can be found in Chapter II.

To further stress that the results of SMD provide a unique and valuable view of seismic data, in Chapter III I created an event detection algorithm that directly operates on the results of SMD, rather than the seismic data matrix. The algorithm was applied to DAS data from FORGE testing site [10] and it was able to accurately pick seismic signals. However, it detected only about 40% of events that were previously detected by a semblance-based detection algorithm. The difference is largely due to the elementary pre-processing steps performed on the data to which SMD was applied. For the semblance-based algorithm, the ambient noise was removed in a more sophisticated pre-processing stage, while SMD was applied to data still containing ambient noise but relied on the shift vectors to filter it out. It should also be noted however that the SMD-based detection algorithm is much faster than the semblance-based algorithm. Furthermore, the semblance-based algorithm requires knowledge of the seismic velocities in the area surrounding the receivers, while SMD does not. Regarding the application to DAS data, SMD can detect events in real time, reduce the memory requirements (which are extensive for DAS) and improve the signal to noise ratio which could prove useful for source location and mechanism determinations.

With the goal of efficiently inferring source location and mechanism, I've also created an algorithm for reducing the computation time of finite difference methods used for acoustic wave simulations. I refer to it as reduced domain method (RDM), and it can easily be generalized to elastic wave simulations. In Chapter IV the algorithm is developed for two-dimensional acoustic-wave modeling and is found to reduce computation time by over 50% while maintaining low errors.

In summary, in this dissertation I have developed practical tools for reducing the memory requirements of seismic data storage, improving SNR, efficiently running event detection on large quantities of data in real time, and reducing the computational cost of acoustic wave simulations. These developments together reduce the cost and computational requirements for seismic monitoring which will prove valuable in geothermal resource development as society progresses along the clean energy transition.

REFERENCES

[1] Warpinski, Norman Raymond, Michael J. Mayerhofer, Michael C. Vincent, Craig L. Cipolla, and E. P. Lolon. "Stimulating unconventional reservoirs: maximizing network growth while optimizing fracture conductivity." *Journal of Canadian Petroleum Technology* 48, no. 10 (2009): 39-51.

[2] Hayat, Muhammad Badar, Danish Ali, Keitumetse Cathrine Monyake, Lana Alagha, and Niaz Ahmed. "Solar energy—A look into power generation, challenges, and a solar-powered future." *International Journal of Energy Research* 43, no. 3 (2019): 1049-1067.

[3] DeCastro, M., S. Salvador, M. Gómez-Gesteira, X. Costoya, D. Carvalho, F. J. Sanz-Larruga, and L. Gimeno. "Europe, China and the United States: Three different approaches to the development of offshore wind energy." *Renewable and Sustainable Energy Reviews* 109 (2019): 55-70.

[4] Kagel, Alyssa, Diana Bates, and Karl Gawell. "A guide to geothermal energy and the environment." (2005).

[5] Wessels, Scott A., Alejandro De La Peña, Michael Kratz, Sherilyn Williams-Stroud, and Terry Jbeili. "Identifying faults and fractures in unconventional reservoirs through microseismic monitoring." *First break* 29, no. 7 (2011).

[6] Li, Xinyang, Jimmy Zhang, Marcel Grubert, Carson Laing, Andres Chavarria, Steve Cole, and Yassine Oukaci. "Distributed acoustic and temperature sensing applications for hydraulic fracture diagnostics." In *SPE Hydraulic Fracturing Technology Conference and Exhibition*. OnePetro, 2020.

[7] Lindsey, Nathaniel J., T. Craig Dawe, and Jonathan B. Ajo-Franklin. "Illuminating seafloor faults and ocean dynamics with dark fiber distributed acoustic sensing." *Science* 366, no. 6469 (2019): 1103-1107.

[8] Zhan, Zhongwen. "Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas." *Seismological Research Letters* 91, no. 1 (2020): 1-15.

[9] Olasolo, P., M. C. Juárez, M. P. Morales, and I. A. Liarte. "Enhanced geothermal systems (EGS): A review." *Renewable and Sustainable Energy Reviews* 56 (2016): 133-144.

[10] Moore, Joseph, John McLennan, Kristine Pankow, Stuart Simmons, Robert Podgorney, Philip Wannamaker, Clay Jones, William Rickard, and Pengju Xing. "The Utah Frontier Observatory for Research in Geothermal Energy (FORGE): a laboratory for characterizing, creating, and sustaining enhanced Geothermal systems." In *Proceedings of the 45th Workshop on Geothermal Reservoir Engineering*. Stanford University, 2020.

[11] Brankovic, Milan, Eduardo Gildin, Richard L. Gibson, and Mark E. Everett. "A Machine Learning-Based Seismic Data Compression and Interpretation Using a Novel Shifted-Matrix Decomposition Algorithm." *Applied Sciences* 11, no. 11 (2021): 4874.

[12] Brankovic, Milan, and Mark E. Everett. "A Method for Modeling Acoustic Waves in Moving Subdomains." In *Acoustics*, vol. 4, no. 2, pp. 394-405. MDPI, 2022.

APPENDIX A
THE QUANTITATIVE EXAMPLE

Consider a small matrix $\boldsymbol{M}^{s} \in \mathbb{R}^{8,8}$ representing a simple wave arrival recorded by 8 receivers.

$$\mathbf{M}^{\mathbf{s}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & -3 & -2 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the matrix $\boldsymbol{M}^{s}$, each receiver is represented by one of the columns, and the time increases from top row to the bottom row. The source is closest to the third receiver, and the perceived waveform is simply:

$$\mathbf{w}^{\mathbf{s}} = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

Figure A1 loosely describes a toy model that can generate the matrix $\boldsymbol{M}^{s}$.

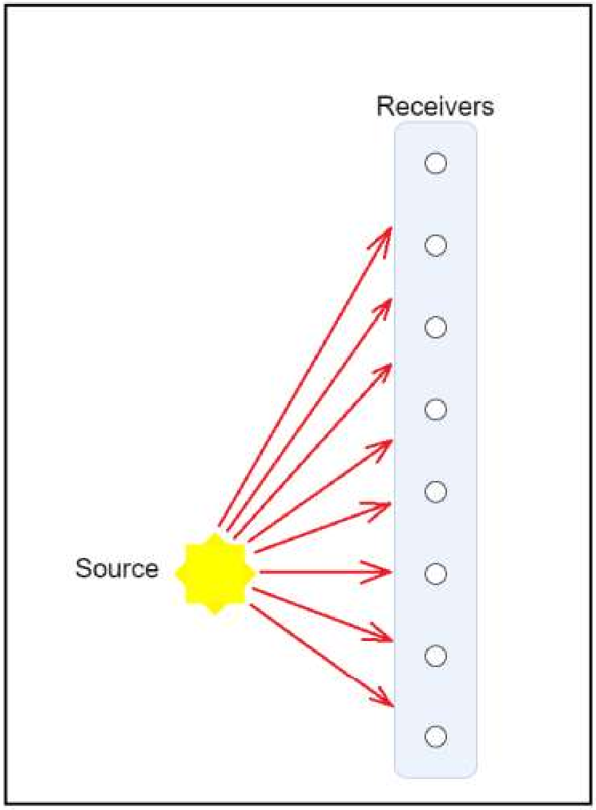

Figure A1. A toy model corresponding to the data recorded in the matrix $\boldsymbol{M}^s$.

By applying SMD to matrix $\boldsymbol{M}^s$ we extract a pair of the singular vectors ($\boldsymbol{u}^{SMD}$, $\boldsymbol{v}^{SMD}$), and a shift vector ($\boldsymbol{s}$). Keep in mind that when using SMD, the singular value λ^{SMD} is stored in right singular vector $\boldsymbol{v}^{SMD}$ by multiplying the vector with it:

$$\mathbf{u}^{\mathbf{SMD}} = \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}^{\mathbf{SMD}} = \begin{bmatrix} \sqrt{2} \\ 2\sqrt{2} \\ 3\sqrt{2} \\ 2\sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Using the singular vectors and the shift vector we can reconstruct a matrix $\boldsymbol{M}^{SMD}$ to be identical to $\boldsymbol{M}^s$:

$$\mathbf{M}^{\mathbf{SMD}} = \chi(\mathbf{u}^{\mathbf{SMD}}\mathbf{v}^{\mathbf{SMD}^T}, \mathbf{s})$$

$$\mathbf{M}^{\mathbf{SMD}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & -3 & -2 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can also apply regular SVD to the matrix $\boldsymbol{M}^s$ to extract a singular value λ and singular vectors $\boldsymbol{u}$ and $\boldsymbol{v}$:

$$\lambda \approx 5.92, \quad \mathbf{u} \approx \begin{bmatrix} 0 \\ 0.68 \\ -0.73 \\ 0.05 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v} \approx \begin{bmatrix} -0.13 \\ 0.48 \\ 0.72 \\ 0.48 \\ -0.13 \\ 0.01 \\ 0 \\ 0 \end{bmatrix},$$

We can reconstruct a matrix $\boldsymbol{M}^{SVD}$ from the extracted eigenpair with the following equation:

$$\mathbf{M}^{\mathbf{SVD}} = \lambda(\mathbf{u}\mathbf{v}^T)$$

$$\mathbf{M}^{\mathbf{SVD}} \approx \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.53 & 1.93 & 2.90 & 1.93 & -0.53 & 0.03 & 0 & 0 \\ 0.56 & -2.06 & -3.08 & -2.06 & 0.56 & -0.03 & 0 & 0 \\ -0.04 & 0.13 & 0.19 & 0.13 & -0.04 & 0 & 0 & 0 \\ 0 & 0 & -0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

There is a significant difference between $\boldsymbol{M}^s$ and $\boldsymbol{M}^{SVD}$. The data from matrix $\boldsymbol{M}^s$ cannot be described by a single pair of singular vectors and a singular value obtained by applying regular SVD. However, it can be described by a pair of singular vectors coupled with a shift vector obtained by applying SMD.