SIGNAL
ANALYSIS &
IMAGING GROUP

Annual Report

SAIG 18

November 2017

Department of Physics
University of Alberta
Edmonton, Alberta

Contact

M D Sacchi
Department of Physics
University of Alberta
Edmonton
Canada T6G 2E1

Phone: (780)-492-1060
E-mail: msacchi@ualberta.ca
URL: http://saig.physics.ualberta.ca/

Contributors / Members

Amsalu Y. Anagaw received a B.Sc. in Applied Physics from Addis Ababa University (Ethiopia) in 2003, and a Ph.D. in Geophysics from University of Alberta 2014. Research interests are in the area of linear and non-linear inversion, regularization of wave-equation migration and seismic imaging.

`EMAIL: aanagaw@ualberta.ca`

Breno Bahia graduated in geophysics at the Federal University of Bahia (Brazil) in 2016 with a sixteen months visiting period at the University of Alberta through the program Science without Borders. Currently he is enrolled in the MSc program at the University of Alberta. His main research interest are in signal processing and inverse theory.

`EMAIL: figueire@ualberta.ca`

Luca Bianchin graduated in Civil and Environmental Engineering at University of Trieste (Italy) with a master degree thesis on seismic signal enhancement methods for single channel shallow surveys (2014). After an experience as a seismic data processor at CGG he went back to Trieste where he is currently a PhD student within the Geosciences department. He is focusing his PhD work on bandwidth extension and trace reconstruction applied to absolute impedance inversion from seismic data.

`EMAIL: bianchin@ualberta.ca`

Fernanda Carozzi received her Diploma in Geophysics in 2010 from Universidad Nacional de La Plata, Argentina and M.Eng. in Oil and Gas Production in 2012 from Instituto Tecnologico de Buenos Aires, Argentina. She joined the Signal Analysis and Imaging Group at the University of Alberta in September 2015 as a Ph.D. Her research is focused on seismic reconstruction.

EMAIL: `carozzi@ualberta.ca`

Ke Chen received his B.Sc. in Geophysics from the University of Science and Technology of China in 2011 and M.Sc. in Geophysics from University of Alberta in 2013. He is currently enrolled in the Ph.D. program in Geophysics in the University of Alberta. His research is mainly focused on robust methods for erratic noise attenuation, seismic data reconstruction and time-domain elastic full waveform inversion.

EMAIL: `ke7@ualberta.ca`

Jinkun Cheng received his B.Sc. in Geophysics from Jilin University in 2011 and recenlty (2017) completed a PhD in Geophysics at the University of Alberta. His research is focused simultaneous source separation and tensor algebra for seismic data processing. Jinkun has recently joined CGG.

EMAIL: `jinkun@ualberta.ca`

Wenlei Gao received his B.Sc in 2010 and M.Sc in 2013 in Geophysics from China University of Petroleum, Beijing, China. From 2013 to 2014 he worked for the Research Institute of China National Offshore Oil Company as seismic data interpreter. He is currently enrolled in the Ph.D. program in Geophysics in the University of Alberta. His research is mainly focused on multicomponent seismic data registration and joint deconvolution.

EMAIL: `wgao1@ualberta.ca`

Bernal Manzanilla is a Geophysical engineer from UNAM (National Autonomous University of Mexico), currently doing his PhD at the University of Alberta. From 2014 and until 2016, he worked at the Engineering Faculty of UNAM as a part time lecturer in Geophysics and Engineering. From February 2010 and until September 2014, he worked in quality control and seismic data processing both onshore and offshore.

EMAIL: `bmanzani@ualberta.ca`

Gian Matharu received his BSc in Honours Physics and MSc in Geophysics from the University of British Columbia. His MSc work involved studies of crustal anisotropy in subduction zones and tomographic inversion. As a member of SAIG (since 2014) his research has been focused on elastic full waveform inversion in global and exploration seismology.

EMAIL: gian@ualberta.ca

Mauricio D. Sacchi received a diploma in geophysics from the National University of La Plata (Argentina) in 1988 and a Ph.D. in geophysics from UBC in 1996. He joined the Department of Physics at the University of Alberta in 1997 where he currently is a professor of geophysics. His research interest are in applied and global seismology.

EMAIL: msacchi@ualberta.ca

Industrial sponsors

ARCIS

CGG

CHEVRON

Claritas

CONOCO PHILLIPS

ENI

EXXONMOBIL

KEY SEISMIC

PGS

SHELL

Contents

:

Random noise attenuation via randomized CP decomposition

Wenlei Gao[1] and Mauricio D Sacchi

Abstract

Tensor algebra provides a powerful framework for multidimensional seismic data processing. A noise-free seismic volume can be represented by a low-rank tensor. Noise will increase the rank of the tensor. Hence, random noise attenuation can be attained via low-rank tensor filtering. Our filtering method adopts the CANDECOMP / PARAFAC (CP) decomposition. It decomposes N-dimensional seismic data in rank-one N-dimensional volumes. Alternating Least Squares (ALS) is adopted to compute the CP decomposition. In addition, we introduce a randomized CP decomposition to speed up the ALS algorithm. Computational time is saved by avoiding unfolding and folding large tensors. We examine the performance of the fast CP decomposition on synthetic data and two 3D field data sets.

1.1 Introduction

Random noise attenuation is a critical step in seismic data processing. Traditional methods for random noise attenuation adopt prediction filters in $f-x$ domain (Canales, 1984; Gulunay, 2005) or in $t-x$ domain (Abma and Claerbout, 1995). Incoherent noise removal can also be formulated

[1] Email: wgao1@ualberta.ca

as a matrix rank-reduction problem (Trickett, 2003) in the $f-x-y$ domain. Similarly, random noise reduction and seismic data interpolation can be implemented via rank-reduction methods that operate on block Hankel matrices in frequency-space domain (Trickett et al., 2010; Oropeza and Sacchi, 2011).

There is no ambiguity in the definition of the rank of a matrix. There is also no ambiguity in a reduced rank decomposition of matrices (Strang et al., 1993). However, the decomposition of tensors in terms of low-rank tensors requires some attention. First of all, there is no unique form in which a tensor can be decomposed in terms of a tensors of lower rank because there is no unique definition of tensor rank (Kolda and Bader, 2009). For instance, one can decompose a tensor in terms of the superposition of matrices and a small tensor by adopting the high-order SVD (HO-SVD). This approach was used for interpolation of 5D seismic volumes Kreimer and Sacchi (2011, 2012). Similarly, one can adopt Parallel Matrix Factorization (PMF) to represent a tensor for one with low-rank unfondings (Xu et al., 2013; Sacchi et al., 2015). Similarly, a tensor can be decomposed via the CP decomposition (Carroll and Chang, 1970; Harshman, 1970; Battaglino et al., 2017). In this particular decomposition, a rank-one tensor is a volume computed by multi-dimensional outer product of vectors. This article investigated the performance of the CP decomposition on random noise attenuation in the time-space domain.

1.2 Theory

We denote N-dimensional seismic data by $\boldsymbol{\mathcal{D}}$ with element $D_{i_1,\dots,i_N}$. The first dimension represents time and the remaining dimensions indicate spatial coordinates. For instance, in a $5D$ seismic dataset, spatial dimensions could represent x, y midpoint coordinates and x, y offsets. The size of each dimension is indicated by $\{I_1, \cdots, I_N\}$. Using the tensor CP decomposition (Kolda and Bader, 2009), noise-free seismic data can be approximated via the sum of R rank-one tensors

$$\boldsymbol{\mathcal{D}} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}\,, \tag{1.1}$$

where, R is called the rank of the tensor. The factor vectors are given by $\mathbf{a}_r^{(n)}$ of length I_n, the symbol $\circ$ represents *outer product*. The collection of all factor vectors for a given mode organized as factor matrix

$$\mathbf{A}^{(n)} = \begin{bmatrix}\mathbf{a}_1^{(n)} & \mathbf{a}_2^{(n)} & \cdots & \mathbf{a}_R^{(n)}\end{bmatrix} \in \mathbb{R}^{I_n \times R}.$$

The mode-n unfolding of the tensor $\boldsymbol{\mathcal{D}}$ can be represented by means of factor matrices as follows

$$\mathbf{D}_{(n)} = \mathbf{A}^{(n)}\mathbf{Z}^{(n)} \tag{1.2}$$

where, $\mathbf{D}_{(n)}$ is the mode-n unfolding or matricization of the tensor $\boldsymbol{\mathcal{D}}$, $\mathbf{Z}^{(n)}$ is computed as

$$\mathbf{Z}^{(n)} = \underset{k\neq n}{\odot} \mathbf{A}^{(k)} = \mathbf{A}^{(N)} \odot \cdots \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \cdots \odot \mathbf{A}^{(1)}.$$

Where, $\odot$ is used to indicate $Khatri-Rao\,product$, which is also known as the matching *columnwise* $Kronnecker$ *product* defined as

$$\mathbf{A}^{(i)} \odot \mathbf{A}^{(j)} = \begin{bmatrix}\mathbf{a}_1^{(i)} \otimes \mathbf{a}_1^{(j)} & \cdots & \mathbf{a}_R^{(i)} \otimes \mathbf{a}_R^{(j)}\end{bmatrix},$$

where $\otimes$ represents *kronnecker product*. The standard algorithm for finding each factor matrix adopts the alternating least squares (ALS) method (Harshman, 1970; Kolda and Bader, 2009; Battaglino et al., 2017). This method loop over all the modes, fixing every factor matrix but $\mathbf{A}^{(n)}$. According to equation 2.2, the factor matrix $\mathbf{A}^{(n)}$ is obtained by minimizing the following cost function

$$\underset{\mathbf{A}^{(n)}}{arg\,min}\, ||\mathbf{D}_{(n)} - \mathbf{A}^{(n)}\mathbf{Z}^{(n)T}||_F^2\,. \tag{1.3}$$

The normal equations for equation 2.3 is given by

$$\mathbf{A^{(n)}}(\mathbf{Z^{(n)T}Z^{(n)}}) = \mathbf{D}_{(n)}\mathbf{Z}^{(n)}, \tag{1.4}$$

by taking advantage of the properties of $Khatri-Rao\,product$

$$\left(\mathbf{A}^{(i)} \odot \mathbf{A}^{(j)}\right)^T \left(\mathbf{A}^{(i)} \odot \mathbf{A}^{(j)}\right) = \mathbf{A}^{(i)T}\mathbf{A}^{(i)} \circledast \mathbf{A}^{(j)T}\mathbf{A}^{(j)}, \tag{1.5}$$

where, $\circledast$ represents element-wise product of the matrices. We have

$$\mathbf{Z}^{(n)T}\mathbf{Z}^{(n)} = \underset{k\neq n}{\circledast} \mathbf{A}^{(k)T}\mathbf{A}^{(k)} \tag{1.6}$$

Algorithm 1 shows the application of ALS method to compute the CP decomposition. Phan et al. (2013) propose a method to reorganize the computation of $line7$ and $line9$ in Algorithm 1. The idea is to reduce the times of permuting N-dimensional array and the overall memory cost of the ALS algorithm. The over-determined nature of equation 2.3 becomes clear when it is expressed in

Algorithm 1.1 ALS

```
function $\{\mathbf{A}^{(n)}\} = ALS(\boldsymbol{\mathcal{D}}, R)$
    Initialize factor matrices $\mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}$
    while termination criteria not met do
        for n=1 : N do
            $\mathbf{X} = \underset{k \neq n}{\circledast} \mathbf{A}^{(k)T}\mathbf{A}^{(k)}$

            $\mathbf{Z}^{(n)} = \underset{k \neq n}{\odot} \mathbf{A}^{(k)}$

            $\mathbf{Y} = \mathbf{D}_{(n)}\mathbf{Z}^{(n)}$

            $\mathbf{A}^{(n)} = \mathbf{Y}/\mathbf{X}$
        end for
    end while
    return factor matrices $\{\mathbf{A}^{(n)}\}$
end function
```

its transposed form

$$\underset{\mathbf{A}^{(n)}}{arg\,min} ||\mathbf{Z}^{(n)}\mathbf{A}^{(n)T} - \mathbf{D}_{(n)}^{T}||_F^2. \tag{1.7}$$

The system is illustrated by figure 2.1. In general, the number of rows $\prod_{i \neq n} I_i \gg R$. To reduce the workload of the *ALS* algorithm drastically without sacrificing quality, we introduce a *randomized ALS* algorithm. This algorithm uniformly samples rows from $\mathbf{Z}^{(n)}$ and the corresponding rows from $\mathbf{D}_{(n)}^{T}$. The sampled rows can be formed without explicitly forming $\mathbf{Z}^{(n)}$ and unfolding the tensor $\boldsymbol{\mathcal{D}}$. The latter are the most time consuming step in the *ALS* algorithm. In fact, the *jth* row of $\mathbf{Z}^{(n)}$ is the element-wise product of the corresponding rows of the factor matrix

$$\mathbf{Z}^{n}[j,:] = \mathbf{A}^{(N)}[i_N,:] \circledast \cdots \circledast \mathbf{A}^{(n+1)}[i_{n+1},:] \circledast \mathbf{A}^{(n-1)}[i_{n-1},:] \circledast \cdots \circledast \mathbf{A}^{(1)}[i_1,:], \tag{1.8}$$

the mapping between j and $[i_1, \cdots, i_{n-1}, i_{n+1}, \cdots, i_N]$ is via the relation

$$j = 1 + \sum_{\substack{m=1 \\ m \neq n}}^{N} (i_m - 1)J_m, \quad where \quad J_m = \sum_{\substack{k=1 \\ k \neq n}}^{m-1} I_k.$$

Similarly, the fiber $\boldsymbol{\mathcal{D}}[i_1, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N]$ corresponding to the *jth* row of $\mathbf{D}_{(n)}^{T}$. Assuming a desired number of sampled rows n_s, which greater than $max\{I_1, \cdots, I_N, R\}$. $\mathbb{S}$ is used to denote the samples from the set $\{1, 2, \cdots, \prod_{i \neq n} I_i\}$. Battaglino et al. (2017) point out that the sample size $n_s = 10Rlog(R)$ is sufficient for most applications. The randomized version of *ALS* algorithm in

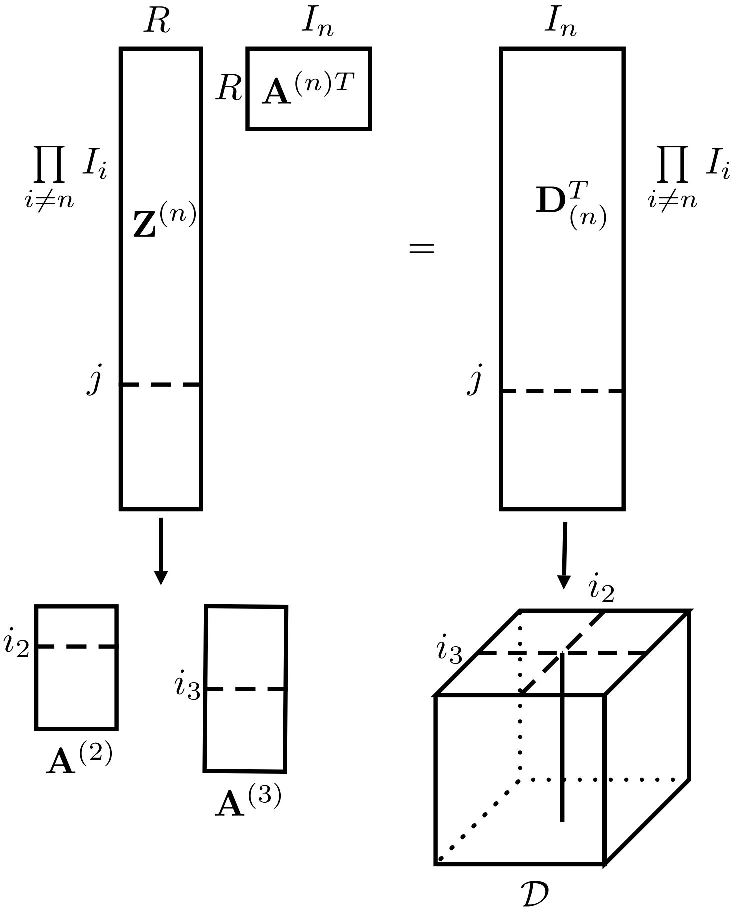


Figure 1.1: The resulted over-determined linear equations of Alternating Least Squares method. Take $3rd$ order tensor as an example, sampled rows of $\mathbf{Z}^{(n)}$ corresponding to rows of factor matrices (dashed lines on factor matrices), Similarly sampled rows of $\mathbf{D}_{(n)}^T$corresponding to fiber in $\boldsymbol{\mathcal{D}}$ (solid line inside of cube), this sampling strategy can void explicitly forming $\mathbf{Z}^{(n)}$ and tensor unfolding.

provided in algorithm 2. *line* 6 and *line* 7 represents extract sampled rows, which is specified by the set $\mathbb{S}$, from factor matrices $\{\mathbf{A}^{(n)}\}$ and tensor $\boldsymbol{\mathcal{D}}$.

Algorithm 1.2 RandALS

```
 1: function $\{\mathbf{A}^{(n)}\} = RandALS(\boldsymbol{\mathcal{D}}, R, ns)$
 2:     Initialize factor matrices $\mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}$
 3:     while termination criteria not met do
 4:         for n=1 : N do
 5:             Define sample set $\mathbb{S} \in \{1, \ldots, \prod_{i \neq n} I_i\}$
 6:             $\mathbf{Z}_s \leftarrow (\mathbb{S}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, \cdots, \mathbf{A}^{(N)})$
 7:             $\mathbf{D_s^T} \leftarrow (\mathbb{S}, \boldsymbol{\mathcal{D}})$
 8:             $\mathbf{A}^{(n)} = \underset{\mathbf{A}^{(n)}}{arg\,min}\, ||\mathbf{Z}_s \mathbf{A}^{(n)T} - \mathbf{D}_s^T||_F^2$
 9:         end for
10:     end while
11:     return factor matrices $\{\mathbf{A}^{(n)}\}$
12: end function
```

We compare the efficiency of ALS with that of $RandALS$ by applying them to two synthetic tensors designed to have rank 20 with 10% random noise, one is $3rd$ order size of $500 \times 300 \times 300$, the other is $200 \times 80 \times 80 \times 80$. This experiment is run on $JULIA$ 0.6 on an Intel i5 3.2 GHz machine with 8GB memory. The target rank is also chosen to be 20. The convergence curve is plotted in figure 2.2. The curves represent relative fitting error versus running time, the relative fitting error is defined as

$$relativeerror = \frac{||\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{D}}'||_F}{||\boldsymbol{\mathcal{D}}||_F},$$

where $\boldsymbol{\mathcal{D}}$ represents original synthetic tensor and $\boldsymbol{\mathcal{D}}'$ is the fitted low-rank tensor. In figure 2.2, stars mark 5 iteration of ALS algorithm and triangles indicate the result of $RandALS$. Both of the proposed algorithms can reach to 10% relative fitting error, which is consistent with the level of added on random noise. we see that the relative speed up of randomized algorithm can increase as much as 5× for $3rd$ order tensor to 15× for $4th$-order tensor.

1.3 Examples

To demonstrate the performance of the proposed algorithms, we generated three synthetic $3D$ data sets $(t - CMP_x - CMP_y)$ with linear, hyperbolic and irregular seismic events, respectively. The

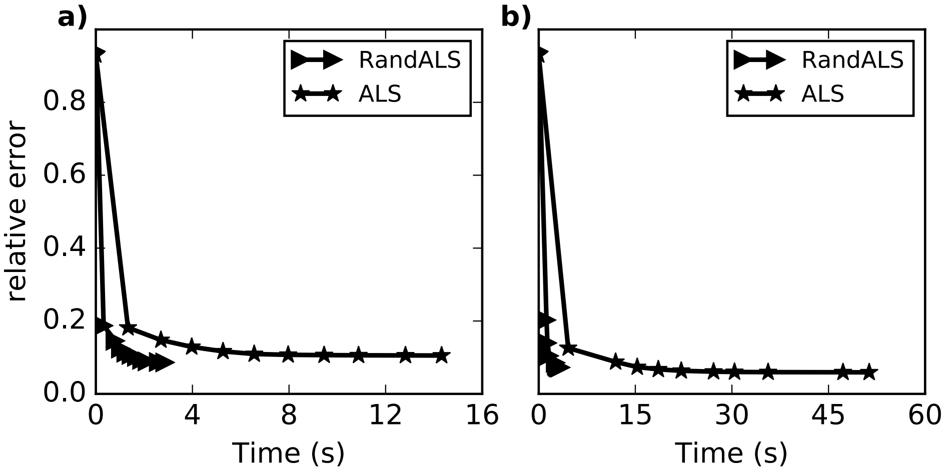


Figure 1.2: The computation time comparison between *ALS* and *RandALS*, one marker represent 5 iterations, stars is the result of *ALS* and triangles indicate *RandALS*. In this test, we run the two algorithms with 50 iterations, both of them can reach to 10% relative error, but the randomized one takes much less time. a) shows the convergence curve for a $3rd$ order $500 \times 300 \times 300$ tensors, the *ALS* algorithm takes $14.31s$ and *RandALS* only taks $2.87s$. b) shows the running time for a $4th$ order $200 \times 80 \times 80 \times 80$ tensor. The *ALS* algorithm takes $51.31s$ and *RandALS* only taks $3.39s$.

size of each data set is of $500 \times 200 \times 200$. Band-limited random noise was added to make the SNR of each data set equal to 0.5. In this report, the SNR is defined as

$$SNR = \frac{||\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{S}}||_F}{||\boldsymbol{\mathcal{S}}||_F}, \tag{1.9}$$

where $\boldsymbol{\mathcal{D}}$ indicates noise-contaminated data, $\boldsymbol{\mathcal{S}}$ is signal, $\mathbf{F}$ denotes the Frobenius norm of tensor. The CP decomposition is performed within sliding windows size of $60times60times60$ and the overlap is 20 samples in each direction. We set the target rank for the data sets with linear or hyperbolic events to be 30 and the data set with irregular events to be 40. The maximum number of iterations is 50, We also stop iterating as the changes of relative error is less than 0.0001.

Figure 2.3 shows one frontal slice ($CMP_y = 130$) of the processed data cube with linear events. From left to right end, each panel shows original noisy data, de-noised one and the difference between them, respectively. The data and the difference are plotted in the same scale and each panel is separated by two empty traces. The data is plotted every 5 traces for visualization. Most of the random noise is removed successfully by the proposed algorithm and the weak event, which is hardly seen in the input data, is correctly recovered by CP decomposition. Similarly, Figure 2.4 shows the result for hyperbolic events, which demonstrates the effectiveness of the proposed algorithm on curved seismic events.

Compared to traditional seismic data de-noising methods, like prediction error filter (PEF), Singular spectrum analysis (SSA), one advantage of tensor decomposition is that the assumption of linear events in local windows is avoided. To demonstrate this superiority, a synthetic data consisted of five irregular events is generated for testing. We first made a data set with linear, parabolic and hyperbolic events, on top of that, we introduced band-limited random time shifts to the trajectory of each events to mimic the irregularity of field seismic data. The result is shown in Figure 2.5.

We also tested the introduced methods on a land 3D post-stack data which is acquired in central Alberta, this data set shows abundant features of paleo-channel. Consider the complexity of real field data, we perform low-rank CP decomposition on 3D overlapping sliding window with size of $60 \times 60 \times 60$ samples, the overlapping of neighbouring windows is 20 samples. The rank R is set to be 30 for all windows, same termination criteria as that for synthetic data is implemented. Figure 2.6a,b,c show the original data cube, the processed one and the difference between them. The time slice and the lateral section indicated by the yellow lines are plotted in Figure 2.7 and 2.8. The features of paleo-channel become more clear and continuous after processing.

The proposed algorithms are further tested on a marine data set with relatively complex structures. The size of this data is $400 \times 600 \times 400$ samples. Like the previous examples, the CP decomposition

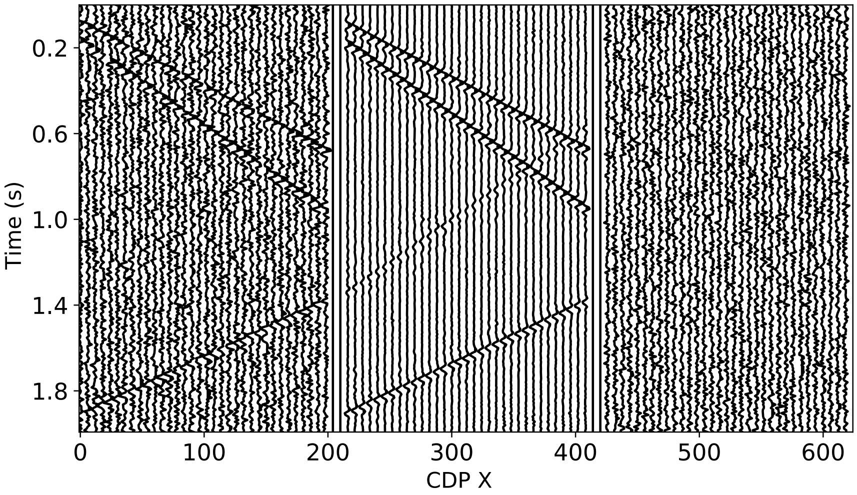


Figure 1.3: One slice of synthetic $3D$ data with linear events size of $500 \times 200 \times 200$, $SNR = 0.5$. From left to right end, each panel shows original noisy data, de-noised one and the removed noise. The data is plotted every 5 traces for visualization.

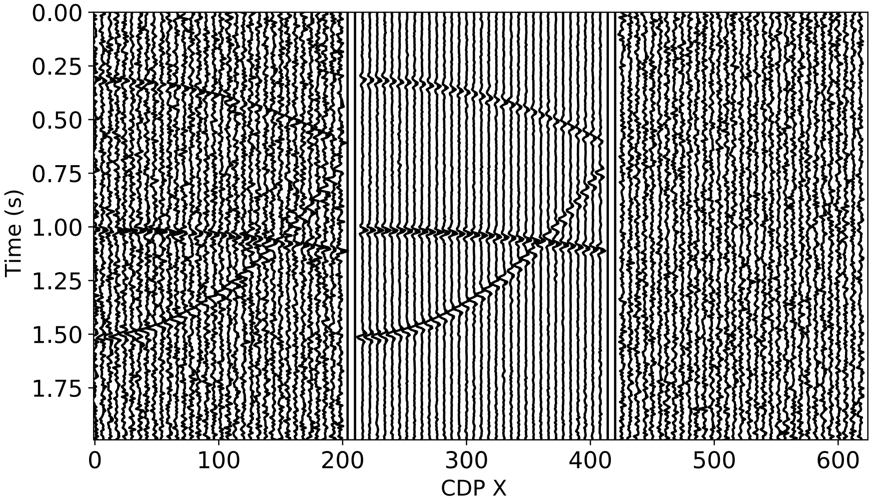


Figure 1.4: One frontal slice of synthetic $3D$ data with hyperbolic events size of $500 \times 200 \times 200$, $SNR = 0.5$. From left to right end, each panel shows original noisy data, de-noised one and the removed noise. The data is plotted every 5 traces for visualization.

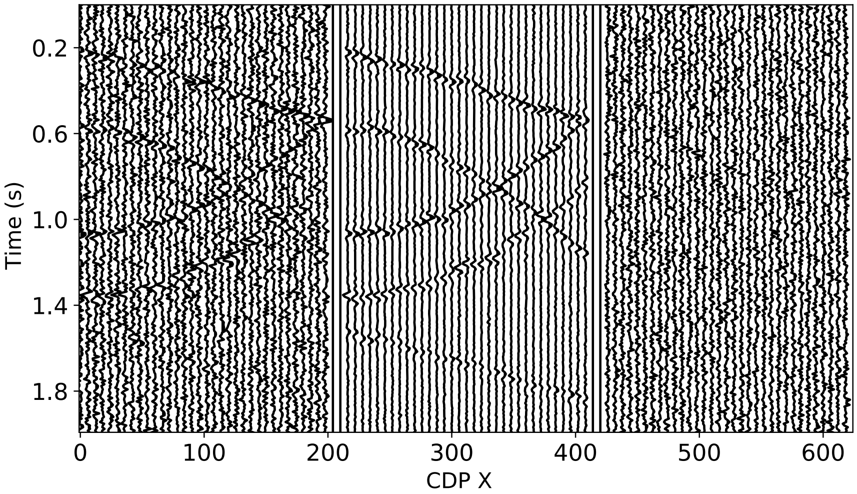


Figure 1.5: One slice of synthetic $3D$ data with irregular events with size of $500 \times 200 \times 200$, $SNR = 0.5$. From left to right end, each panel shows original noisy data, de-noised one and the removed noise. The data is plotted every 5 traces for visualization.

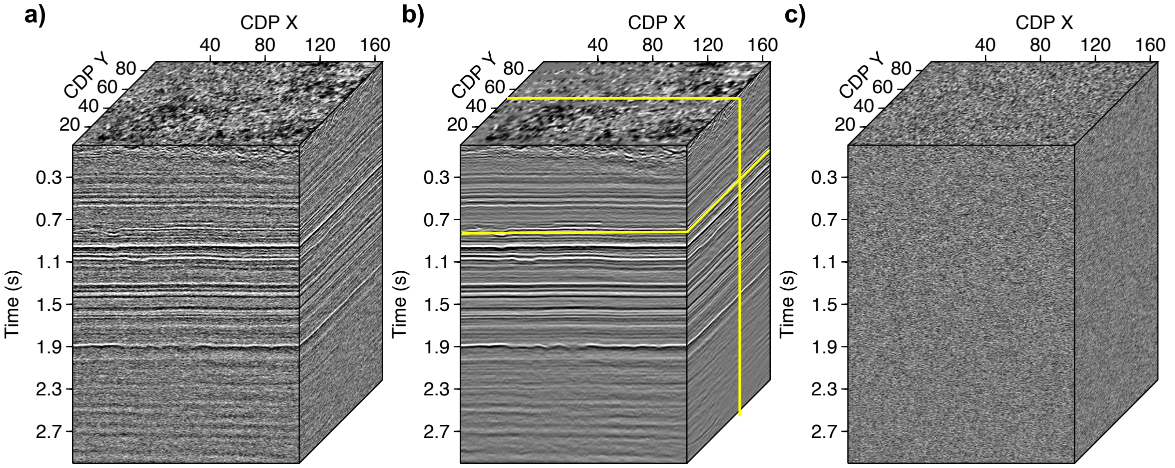


Figure 1.6: 3D land data set. a) shows the original data cube. b) is the processed result. c) displays the difference between them.

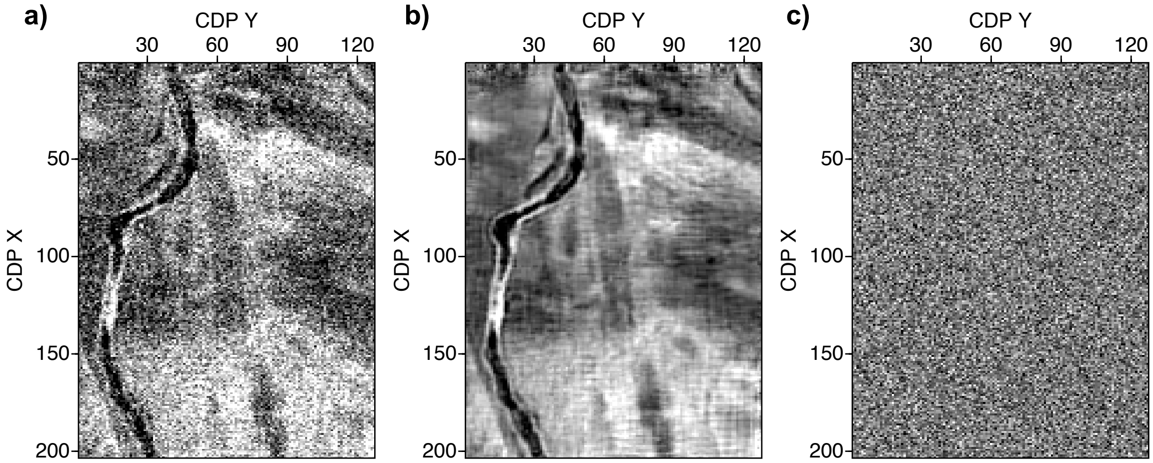


Figure 1.7: Time slice of the $3D$ land data set at time about 0.85 s, a) shows the original time slice. b) is the processed result. c) displays the difference between them.

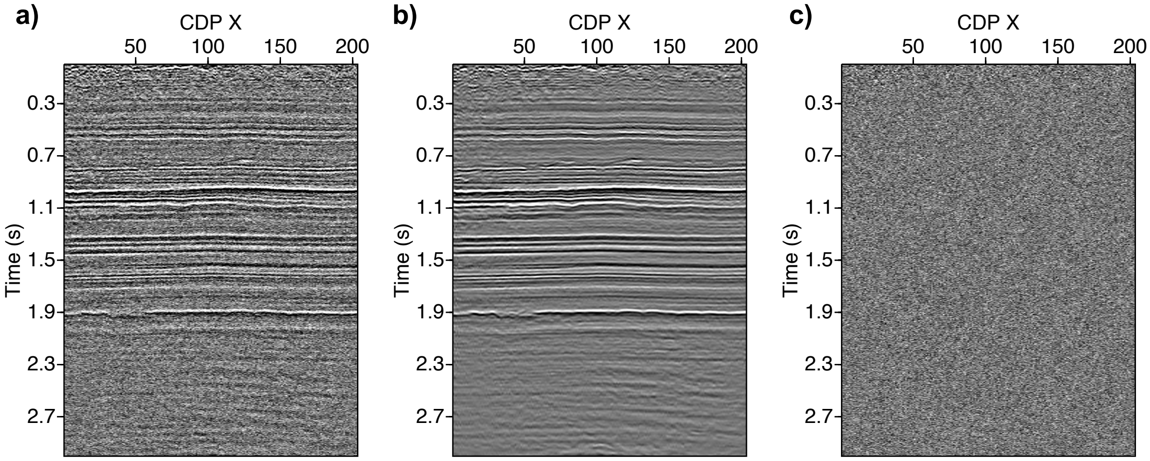


Figure 1.8: Frontal section of the $3D$ data with CMP y number $= 50$, a) shows the original frontal section. b) is the processed result. c) displays the removed noise.

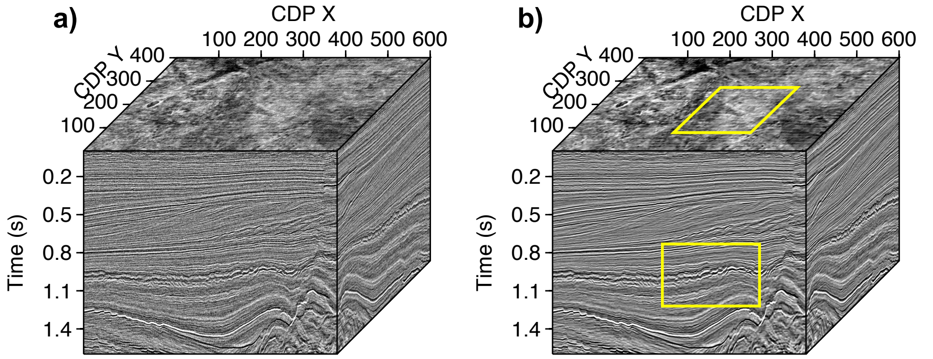


Figure 1.9: 3D migrated marine data set with size of $400times600time400$ samples. a) Original seismic data, b) processed result.

algorithm is performed in overlapping sliding windows. The size of the window is reduced to $30 \times 30 \times 30$ to accommodate the structural complexities of the data and target rank is set to be 25. Figure 2.9a,b show the original, the processed data cube, respectively. To further examine the results of de-noising, we amplify part of the horizontal and frontal slices which are indicated by the yellow boxes. The results are shown in Figure 2.10, 2.11.

1.4 Conclusion

In this article, we have presented a random noise attenuation method for multidimensional seismic data via low rank CP decomposition, a randomized version of Alternating Least squares algorithm is also introduced to speed up the performance without a sacrifice in quality. The effectiveness of the proposed algorithms are demonstrated on synthetic $3D$ data with linear, curve events and real field data. Even thought our examples is given with 3D cases, the extension of proposed method for $5D$ seismic data noise attenuation is straightforward.

References

Abma, R., and J. Claerbout, 1995, Lateral prediction for noise attenuation by tx and fx techniques: Geophysics, **60**, 1887–1896.

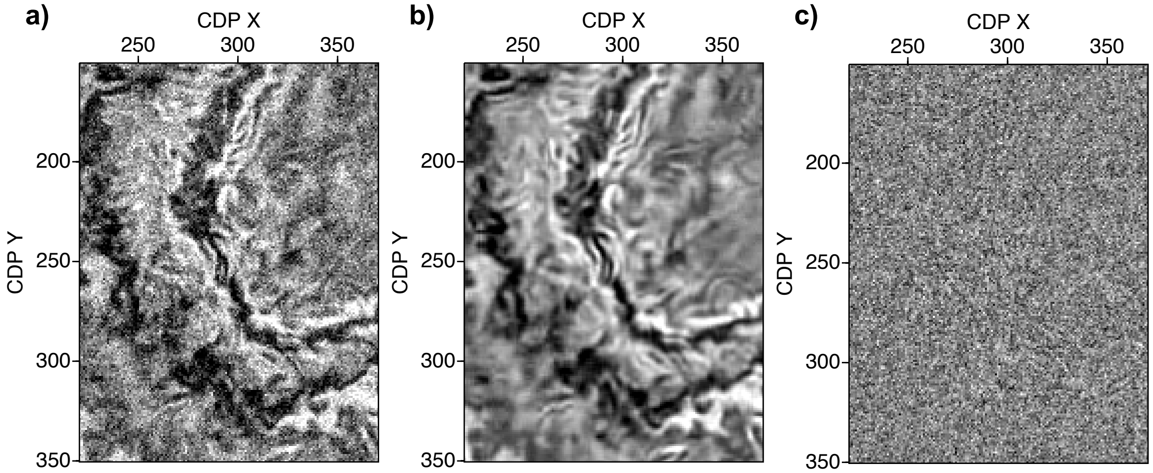


Figure 1.10: Zoomed view of time slice of $3D$ marine data set at time about 0.9 s, the location of this slice is indicated by the yellow square in Figure 2.9b. b) is the processed result. c) displays the difference between them.

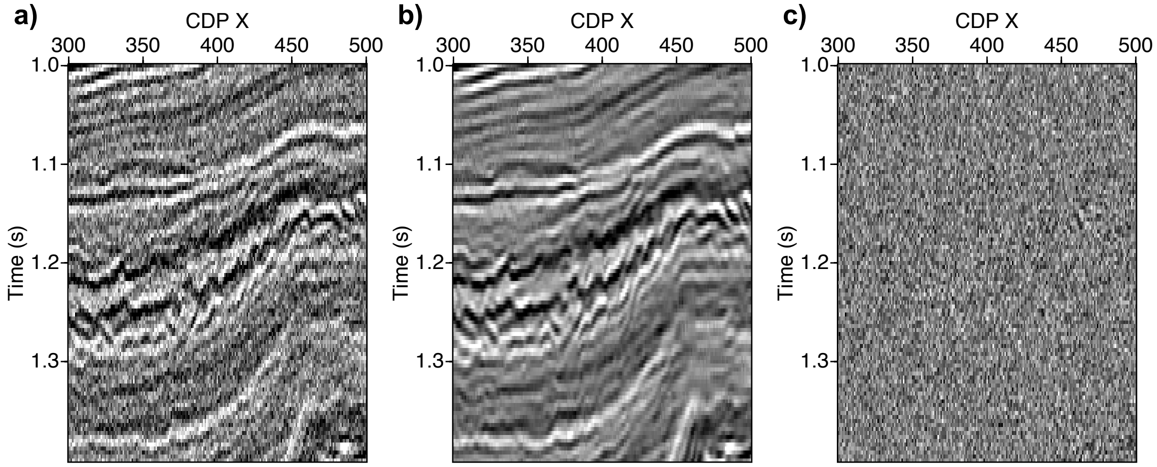


Figure 1.11: Zoomed view of frontal slice of $3D$ marine data set at CDP y number $= 200$, the location of this slice is indicated by the square in Figure 2.9b. b) is the processed result. c) displays the difference between them.

Battaglino, C., G. Ballard, and T. G. Kolda, 2017, A practical randomized cp tensor decomposition: arXiv preprint arXiv:1701.06600.

Canales, L. L., 1984, Random noise reduction, *in* SEG Technical Program Expanded Abstracts 1984: Society of Exploration Geophysicists, 525–527.

Carroll, J. D., and J.-J. Chang, 1970, Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition: Psychometrika, **35**, 283–319.

Gulunay, N., 2005, *in* FXDECON and complex wiener prediction filter: 279–281.

Harshman, R. A., 1970, Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis.

Kolda, T. G., and B. W. Bader, 2009, Tensor decompositions and applications: SIAM review, **51**, 455–500.

Kreimer, N., and M. Sacchi, 2011, Evaluation of a new 5d seismic volume reconstruction method: Tensor completion versus fourier reconstruction: Presented at the CSPG CSEG CWLS Convention.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, V113–V122.

Oropeza, V., and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Phan, A.-H., P. Tichavskỳ, and A. Cichocki, 2013, Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations: IEEE Transactions on Signal Processing, **61**, 4834–4846.

Sacchi, M. D., J. Gao, A. Stanton, and J. Cheng, 2015, Tensor factorization and its application to multidimensional seismic data recovery, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 4827–4831.

Strang, G., G. Strang, G. Strang, and G. Strang, 1993, Introduction to linear algebra: **3**.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3829–3833.

Trickett, S. R., 2003, F-xy eigenimage noise suppression: GEOPHYSICS, **68**, 751–759.

Xu, Y., R. Hao, W. Yin, and Z. Su, 2013, Parallel matrix factorization for low-rank tensor completion: arXiv preprint arXiv:1312.1254.

Pre-stack multicomponent seismic data registration

Wenlei Gao[1]

Abstract

Mapping post-stack or time migrated PS-wave data to the PP-wave time domain is a critical step before joint PP-wave and PS-wave data interpretation. Registration techniques are often constrained by having access to a known V_p/V_s ratio. When an accurate V_p/V_s ratio is not provided, one can solve the problem of seismic data registration by minimizing the difference between the PP-wave and the warped PS-wave data with a smoothing constraint applied on the warping function. For joint pre-stack elastic inversion, it is required to extend the current registration algorithms to the pre-stack domain. However, two main challenges are limiting the application of pre-stack multi-component seismic data. At first, the signal-to-noise ratio (SNR) of pre-stack seismic data is much worse than the one of post-stack data. Secondly, seismic data registration is usually formulated as a highly non-linear optimization problem; an acceptable solution can hardly be obtained by any gradient-based solver if the initial guess is too far away from the global minima. To deal with these challenges, we propose attenuating the noise of the pre-stack seismic data via 5D reconstruction, then applying our registration method on the post-stack seismic data to obtain an estimation of the warping function. Finally, using the warping function for post-stack data as an initial guess, the registration algorithm is implemented on the regularized or raw pre-stack multi-component seismic data.

[1] Email: `wgao1@ualberta.ca`

2.1 Introduction

Multi-component seismic data registration is typically performed by cross-correlation methods. For instance, Gaiser (1996) introduced a correlation-based method to determine the long wave-length components of V_p/V_s ratios. Geis et al. (1990) used both P-wave and S-wave VSP information, combined with a suite of well logs, for the proper correlation of seismic markers in both time and depth. Van Dok et al. (2003) discussed different tools for obtaining and refining V_p/V_s ratios. These authors also studied methods to include the seismic data registration in the processing work-flow to improve the quality of seismic images. Fomel (2007) defined local correlation and applied this local attribute to multi-component seismic image registration to a nine-component land survey. The continuity and smoothness of the local correlation are controlled by changing a regularization term. Hale (2009) described a method to calculate the local correlation between two time-lapse seismic data sets and to estimate three components of displacement have been estimated.

Fomel et al. (2003) showed how one can warp PS-wave events to match PP-wave events by minimizing the differences between PP-wave data and warped PS-wave data. The latter can improve the correlation of seismic volumes obtained by manual interpretation and registration. Seismic data registration by least-squares techniques is a highly non-linear problem. Nickel et al. (2004) also reported applying similar method to multi-component seismic data to estimate high resolution V_p/Vs ratios. All the gradient-based algorithms may easily get trapped in a local minimum. In order to alleviate this problem, Fomel et al. (2005) proposed a multi-step registration method which consists of initial interpretation, amplitude and frequency balancing, registration scan, and least-squares optimization. In some cases, only accounting for time shifts caused by velocity difference is not enough in multi-component seismic data registration, lateral displacements are also need to be handled, Hall (2006) described a 7D warping method for time-lapse seismic data, his method can be extended to multi-component seismic data straight-forwardly.

As we mentioned above, local minima pose big challenges to optimization-based registration methods. In order to avoid this problem, Liner and Clapp (2004) proposed a modified Needleman-Wunsch algorithm, a global optimization method developed for aligning amino acid sequences in proteins, to match PS-wave traces to their corresponding PP-wave traces. Yuan et al. (2008) used Simulated Annealing to minimize the normalized cross-correlation between PP-wave data and PS-wave data. They also account for frequency-domain differences by time-variant spectrum whitening. Compton and Hale (2014) proposed to use dynamic warping to align PS-wave traces to its corresponding PP-wave traces. The sequence of time shifts estimated by this method are a globally optimal solution to the non-linear optimization problem.

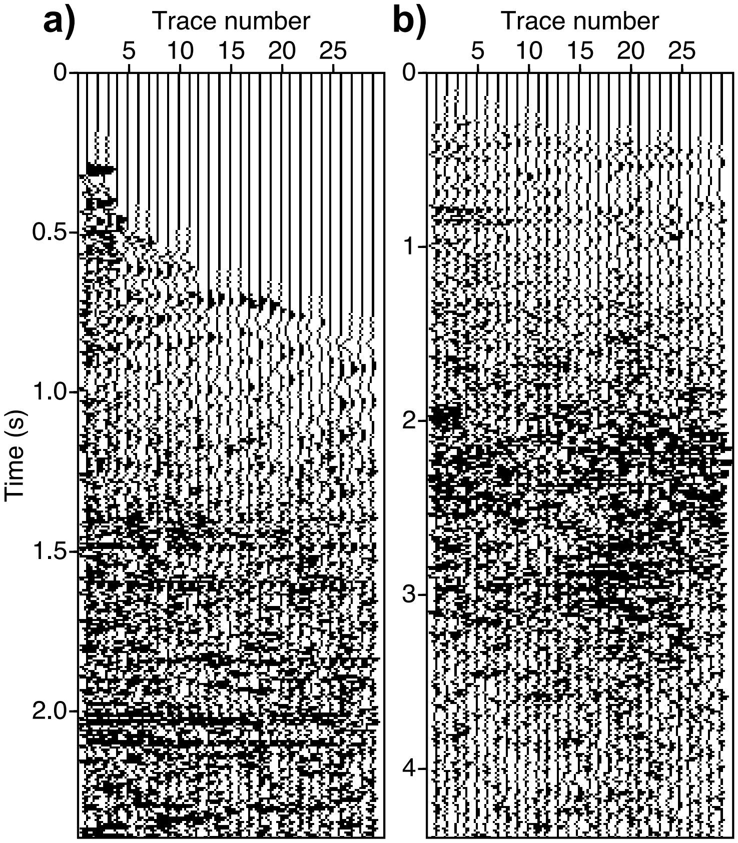


Figure 2.1: Normal move-out time corrected pre-stack seismic data. (a) Pre-stack PP-wave data, (b) Pre-stack PS-wave data.

All the previous cited methods try to estimate a high resolution V_p/V_s ratios from time shifts, which are evaluated by minimizing the difference between PP-wave and warped PS-wave data. However, as V_p/V_s ratios are related to the derivative of time shifts, any non-smoothness in time shifts may cause rapid changes in the estimated V_p/V_s ratios. we proposed a new method which inverts the V_p/V_s ratios directly instead of estimating them from time shifts. To minimize the rapid changes in the inverted V_p/V_s ratios, the cost function is constrained by a smoothing operator (Gao and Sacchi, 2017).

The registration algorithm may fail if we simply extended it to pre-stack domain as the SNR of the pre-stack multi-component seismic is far worse than the post-stack one. Figure 2.1a,b shows the common-depth-gather (CDP) of the normal move-out (NMO) corrected PP, PS-wave data, respectively. Notice the difference of the time axis used for plotting the data. The data are

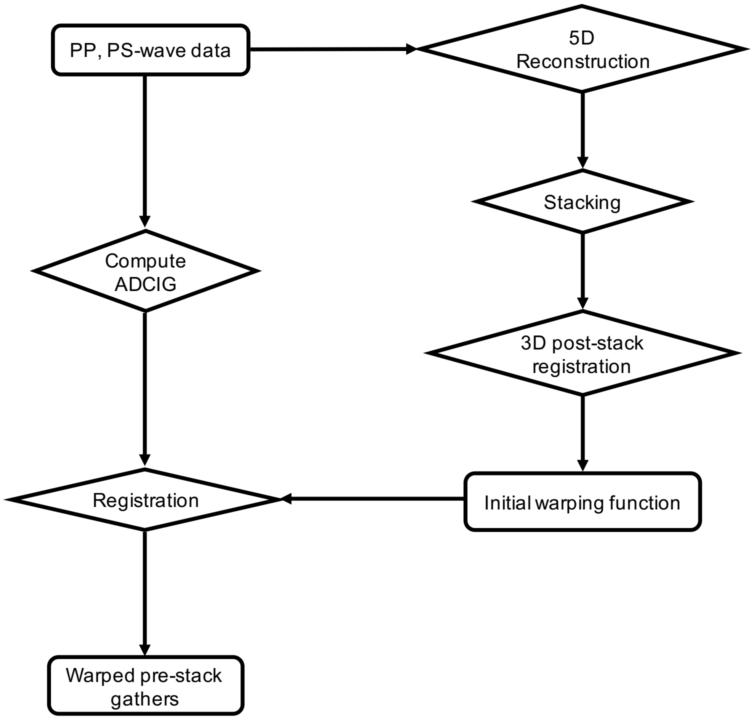


Figure 2.2: Work-flow for pre-stack seismic data registration.

significantly contaminated by noise, which pose great challenges to any registration algorithm. To deal with these challenges properly, we propose a workflow for pre-stack seismic data registration and it is shown in Figure 2.2. The angle domain common image gathers ($ADCIG$) are computed from the input pre-stack NMO-corrected multi-component seismic data. Instead of implementing registration algorithm directly on those generated gathers, we detour by regularizing and attenuating the noise of the pre-stack seismic data via 5D reconstruction. Then high-quality 3D post-stack multi-component data can be prepared for our registration method. The estimated warping function from the post-stack registration is later used as the initial guess for pre-stack seismic data registration. We tested the proposed workflow on an field $3D-3C$ seismic data acquired in central Alberta to demonstrate the effectiveness of the algorithm.

2.2 Theory

In this section, we will briefly illustrate 5D seismic data reconstruction via low-rank tensor completion and multi-component seismic data registration methods, for more details, people can refer to the

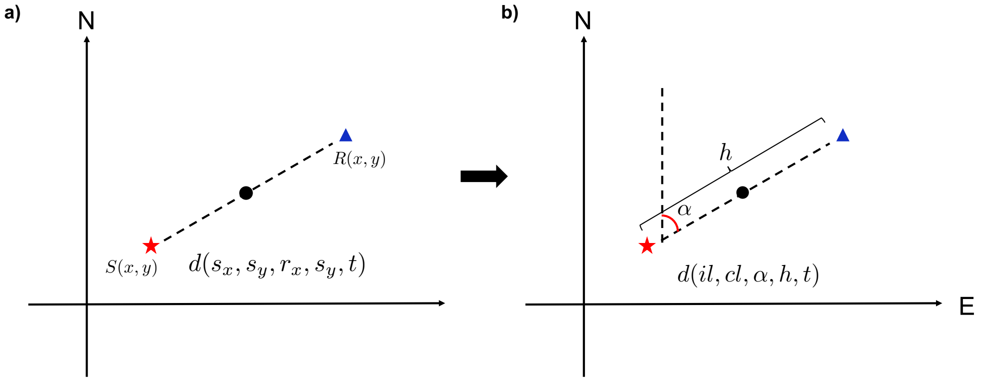


Figure 2.3: Schematic diagram of 5D seismic data acquisition. (a) The geometry of field recorded seismic traces, which depend on 5 variables includes source x coordinate (s_x), source y coordinate (s_y), receiver x coordinate (r_x), receiver y coordinate (s_y) and time t. (b) The domain where $5D$ reconstruction was performed, the recorded data is sorted and binned to CDP gathers, the seismic traces are depending on inline number (il), crossline number (cl), azimuth (α, offset h) and time (t).

works (Kreimer and Sacchi, 2012; Kreimer et al., 2013; Gao et al., 2015) and our Geophysics paper (Gao and Sacchi, 2017).

2.2.1 5D reconstruction

Figure 2.3a shows an schematic diagram of the acquisition geometry of the field $3D$ seismic data. The red star denotes source location $S(x, y)$ and blue triangle is receiver location $R(x, y)$, each of them depends on two spatial variables (x, y), plus the dependency on time t, that is the definition of $5D$ seismic data. Seismic data is usually sorted and binned into CMP/CDP gathers for NMO-corrections. The processed seismic data now depend on the inline (il) and crossline number (cl) of the CDP gather, azimuth (α), offset (h) and time (t), which are shown in Figure 2.3b. Moreover, the $5D$ data is transformed to frequency domain and the seismic data is reformulated as a fourth-order tensor for a particular frequency. In reality, $5D$ seismic data reconstruction is realized by fourth-order low-rank tensor completion for each frequency. The cost function for tensor completion is given as

$$\Phi = ||\boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{D}}||_F^2 + \mu||\boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \times_4 \mathbf{A}^{(4)} - \boldsymbol{\mathcal{D}}||_F^2, \tag{2.1}$$

where $\boldsymbol{\mathcal{S}}$ denotes sampling operator which depends on the acquisition geometry, $\boldsymbol{\mathcal{Z}}$ is the estimated low-rank tensor, $\boldsymbol{\mathcal{D}}$ is one frequency of the observed data. The first term in the cost function is the data fitting term and the second term is the low-rank constraint, the low-rank tensor $\boldsymbol{\mathcal{Z}}$ is represented by Tucker model (Kolda and Bader, 2009), so $\boldsymbol{\mathcal{G}}$ is called core tensor and $\mathbf{A}^{(i)}$ are factor matrix along each dimension. $|| \; ||_F$ represents the Frobenious norm for tensors. It is worth noting that the low-rank tensor completion algorithm is performed in overlap sliding windows.

2.2.2 Multi-component post-stack seismic data Registration

After the $5D$ reconstruction, the high-quality post-stack $3D$ cube can be obtained by stacking the regularized pre-stack data. To estimate a proper initial guess for the pre-stack seismic data registration, we first perform the registration method on the obtained post-stack data. The cost function for the post-stack seismic data registration is

$$\begin{aligned} J = & \iiint \left(d_{PP}(t,x,y) - d_{PS}(w(t,x,y),x,y)\right)^2 \, dt\, dx\, dy \\ & + \iiint \mu_t \left(\frac{\partial^2 w(t,x,y)}{\partial t^2}\right)^2 + \mu_x \left(\frac{\partial^2 w(t,x,y)}{\partial x^2}\right)^2 + \mu_y \left(\frac{\partial^2 w(t,x,y)}{\partial y^2}\right)^2 \, dt\, dx\, dy. \end{aligned} \tag{2.2}$$

Where $d_{PP}(t,x,y)$ is the scaled $3D$ post-stack PP-wave data, which depends PP-wave time t, inline number x and crossline number y, d_{PS} is the PS-wave data, $w(t,x,y)$ is the warping function. The first term of the cost function indicates that the registration process is performed via matching the warped PS-wave to PP-wave data. The smoothness of the warping function is guaranteed by damping the second order derivative of the warping function along each direction. To make the warping function monotonic increasing and reduce the number of unknowns, we further proposed to take advantage of the relationship between the warping function and the V_p/V_s ratio and represent V_p/V_s ratio by cubic splines. For more details about this re-parameterization, please refer to our paper (Gao and Sacchi, 2017).

2.2.3 Multi-component pre-stack seismic data registration

The main purpose of the post-stack registration in this workflow is to obtain an proper initial guess for the pre-stack registration algorithm. The last step is to perform pre-stack registration and the cost function is given as

$$\begin{aligned} J = & \iiint \left(d'_{PP}(t,x',y') - d'_{PS}(w(t,x',y'),x',y')\right)^2 \, dt\, dx'\, dy' \\ & + \mu \iiint \left(w(t,x',y') - w_0(t,x',y')\right)^2 \, dt\, dx'\, dy'. \end{aligned} \tag{2.3}$$

Where d'_{PP} and d'_{PS} are the pre-stack PP-wave and PS-wave data, x', y' denote the azimuth and the incident angle. The constraint is a simple $l2$ norm which guarantees the solutions will not move too far away from the warping function we estimated from post-stack registration.

Note that the rigorous way for pre-stack multi-component seismic data registration is to apply the last step on common azimuth, common incident angle PP, PS-wave traces. However, at the time of preparing this report, we have no access to the velocity model, which is required for generating common incident angle gathers. Therefore, the last step of the workflow were implemented on common azimuth, common-offset gathers, we deemed it is enough to demonstrate the concepts.

2.3 Examples

We used an $3D3C$ data to test the proposed workflow. The testing data is acquired in central Alberta, 2014. The acquisition geometry is shown in Figure 2.4a, where orange lines denote receiver lines and blue lines are source lines. The fold-map of binned PP-wave and PS-wave seismic data are shown in Figure 2.4c,d. The average number of the folds for PP-wave is about 65 and the folds for PS-wave is increased to 85. Limited by the computation power, we only take a swath of the data for testing the algorithm and the location of the data are indicated by the black boxes in the fold maps.

Figure 2.5a,b shows one inline section of the stacked raw PP-wave and PS-wave data, respectively. The recording length is 6.1 s with 1 ms sampling interval. We can see that the raw stacked data is contaminated with strong amplitude noise, especially for PP-wave data from 0.5 s to 1.4 s. It is hardly to see any coherent seismic events. As our main research is about PP-PS wave registration and there is almost no signal after 4.5 s in PS-wave data, so we cut part of the data for registration, the time range of the selected PP and PS-wave data is indicated by the red box in Figure 2.5a,b.

The $5D$ reconstruction method is performed in overlap sliding windows, The size of the window is 128 time samples, azimuth interval is set as 45 degrees and the offset interval is 180 m when organizing the seismic data into tensors and all the traces in one CDP gather are included. The size of the window is 16 in both inline and crossline directions. For each frequency, the target rank is set as $(4, 8, 8, 8)$ for the PP-wave data. For the PS-wave data, same window size is implemented but the target rank is set as $(3, 5, 5, 5)$ considering the lower SNR of PS-wave data. Figure 2.6 shows the results of the PP-wave reconstruction, the raw PP-wave data with missing traces is shown in Figure

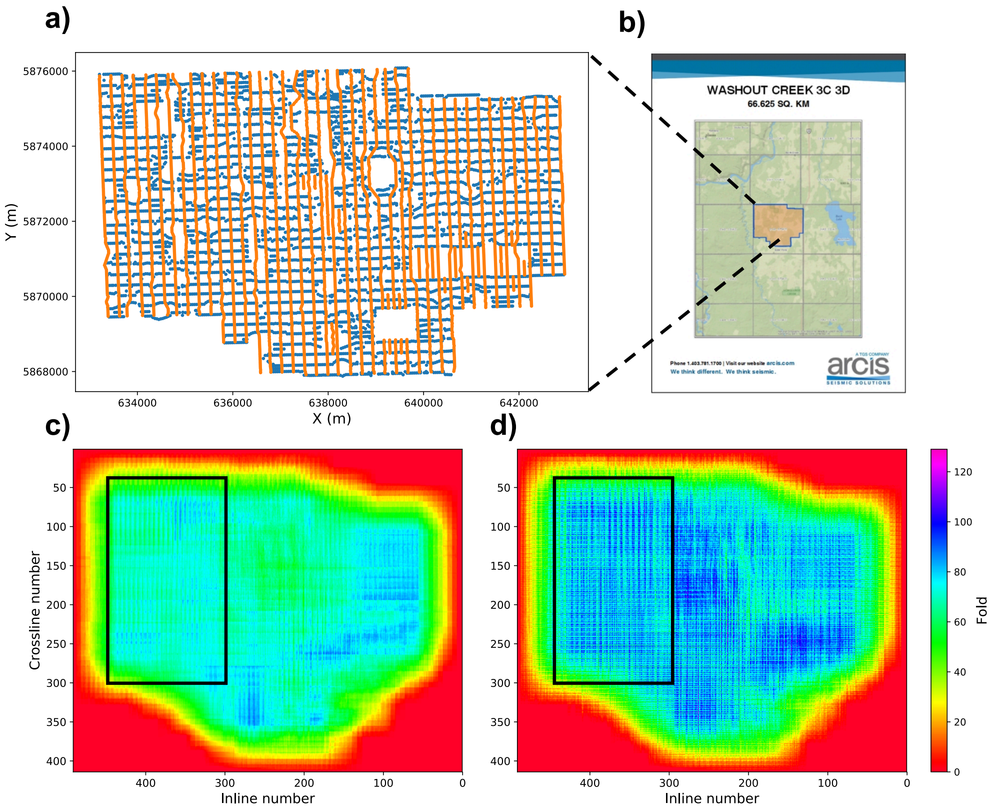


Figure 2.4: Washout-creek seismic data. (a) Acquisition geometry of the 3D seismic data set, orange lines denote receiver lines and blue lines indicate source lines. (b) Location of the seismic survey. (c) The fold-map for PP-wave data, (4) the fold-map for PS-wave data. The black box indicates the area of the data used for testing the proposed algorithm.

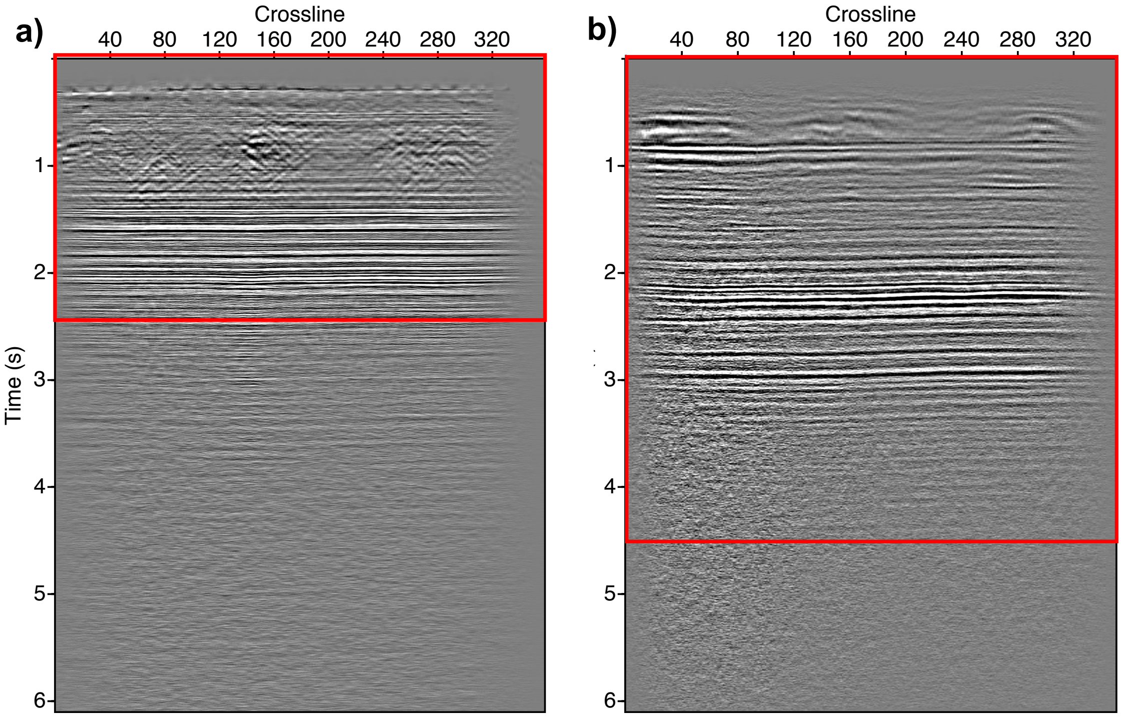


Figure 2.5: Post-stack inline sections and the recording length is 6.1 s. (a) Post-stack PP-wave section. (b) Post-stack PS-wave section. The red boxes denote the time range of the selected data used for testing.

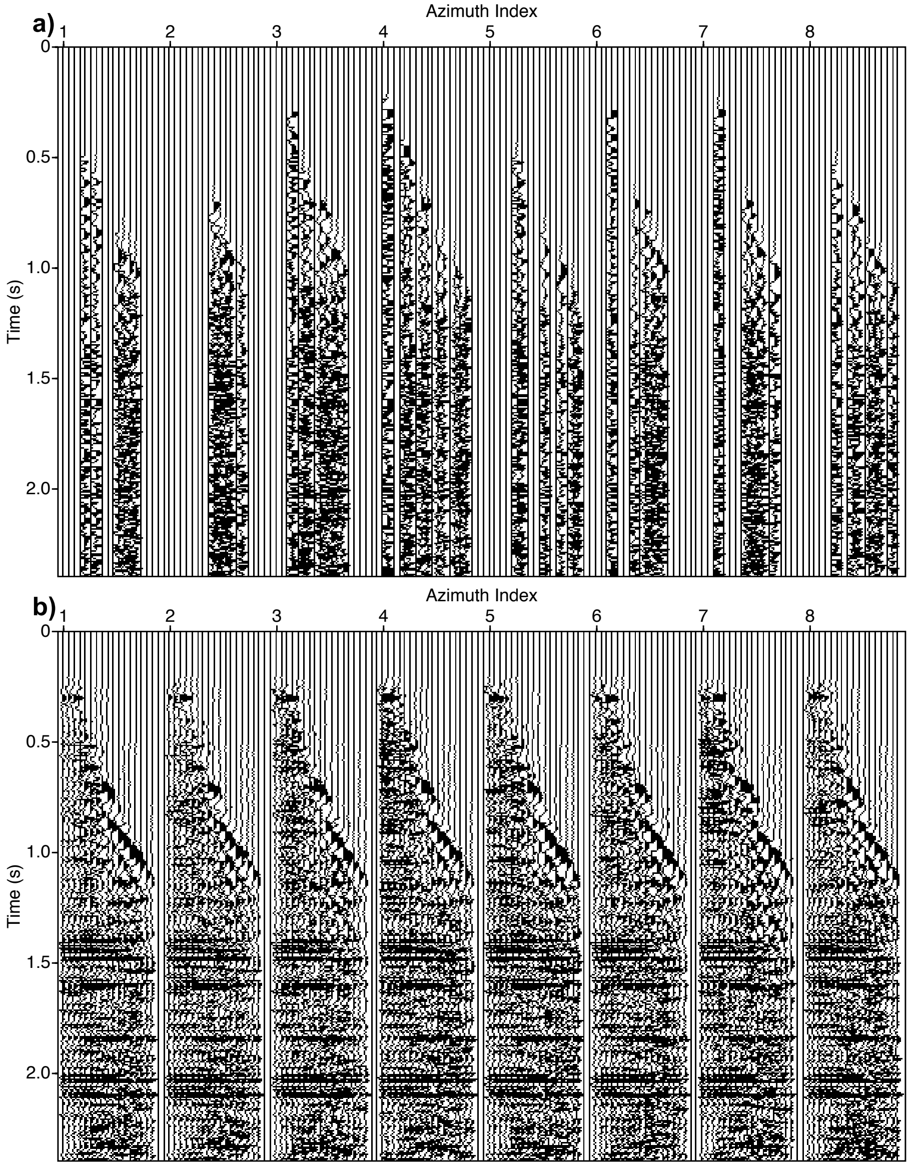


Figure 2.6: The result of 5D reconstruction for the PP-wave data. (a) Raw pre-stack PP-wave data with missing traces. (b) The corresponding reconstructed PP-wave data.

2.6a and the corresponding reconstruction result is shown in Figure 2.6b. After reconstruction, we can see the coherency of PP-wave data are improved significantly.

Similarly, the reconstruction result for the PS-wave is shown in Figure 2.7a,b. Please pay attention to reconstructed near-offset of PS-wave data. The weak reflection features are preserved by our reconstruction algorithm.

Figure 2.8 shows the comparison between the stack sections of the raw PP-wave data and the counterpart obtained after $5D$ reconstruction. The difference is apparent, not only the coherency are improved, The signal in the time range from 0.4 s to 1.4s are shown up in the processed post-stack cube.

Figure 2.9 shows the results for PS-wave data, the amplitude of the post-stack sections after processing are more balanced than the raw stack one. After $5D$ reconstruction, the SNR of the post-stack data are improved greatly and it is crucial for the convergence of the post-stack registration algorithm. To examine the result of the post-stack registration, we will compare the time slice, inline sections and crossline sections of the post-stack PP-wave and warped PS-wave cube. Figure 2.10 shows the time slices of PP-wave data and warped PS-wave data at time about 1.9 s. The main pattern of the amplitude of PP-wave time slice is very similar to the one of the warped PS-wave data.

Figure 2.11a,b shows the inline sections of post-stack PP-wave and warped PS-wave data, respectively. All the main events in the PP-wave and the warped PS-wave data are aligned, we can that even the minor events between 0.6 s and 1.0 s are matched as well if we compared closely.

FIgure 2.12a,b shows the crossline sections of the post-stack PP-wave and the warped PS-wave data, respectively. After post-stack registration and carefully examining the registration result, the obtained warping function is qualified as the initial guess for the pre-stack registration. This step is firstly performed on the processed pre-stack seismic as it is easier to quality-control the registration result, which is shown in Figure 2.13 and 2.14. In Figure 2.13, The first four azimuth sections is displayed and remaining azimuth is shown in Figure 2.14. In each panel, the left half shows the reconstructed pre-stack PP-wave data and right half shows the warped PS-wave data. The main events are aligned by registration.

Finally, the estimated warping function is also applied to raw pre-stack seismic data and the result is shown in Figure 2.15. We did not run the registration algorithm on the raw seismic data, the registration process is realized by using the warping function obtained from the reconstructed data. This is because we have not figured out a good way to evaluate the registration result of the noisy raw seismic data.

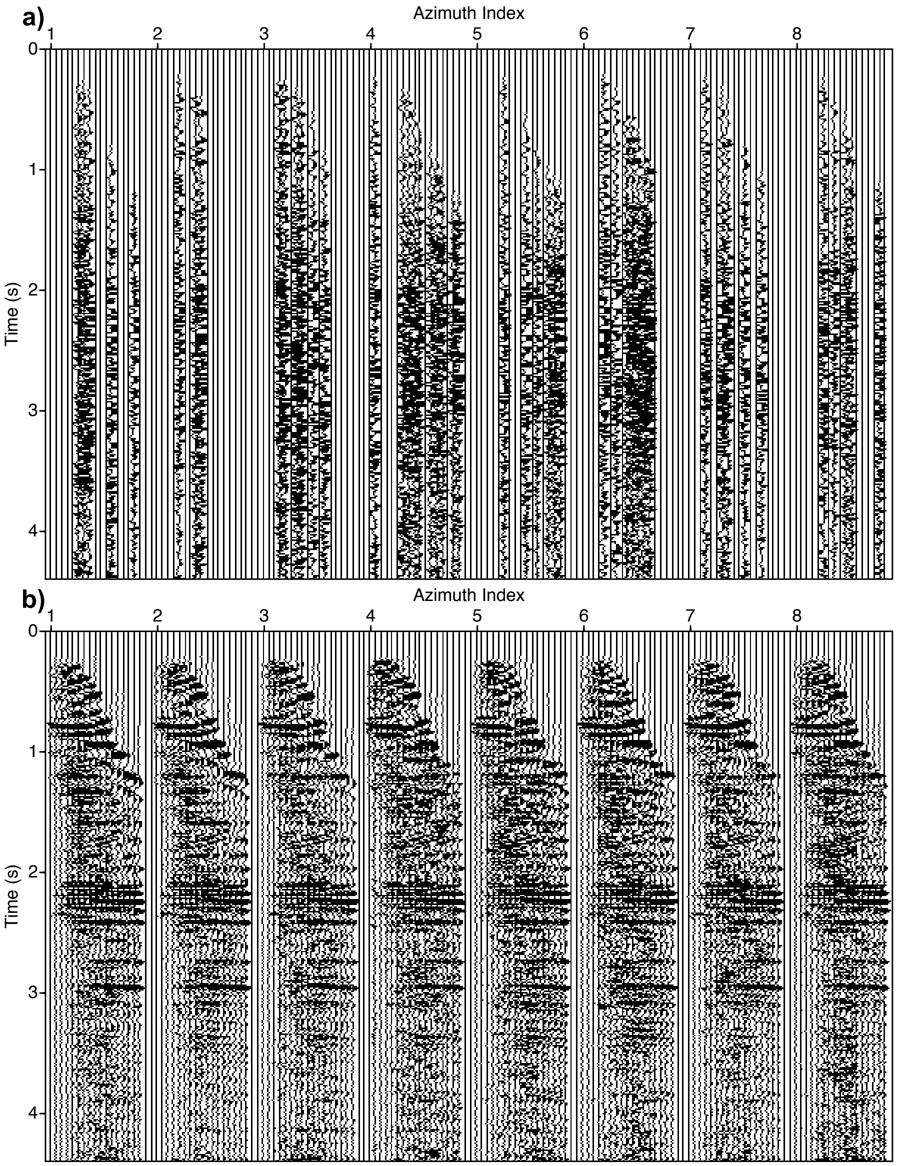


Figure 2.7: The result of 5D reconstruction for the PS-wave data. (a) the raw pre-stack PS-wave data with missing traces. (b) The corresponding reconstructed PS-wave data.

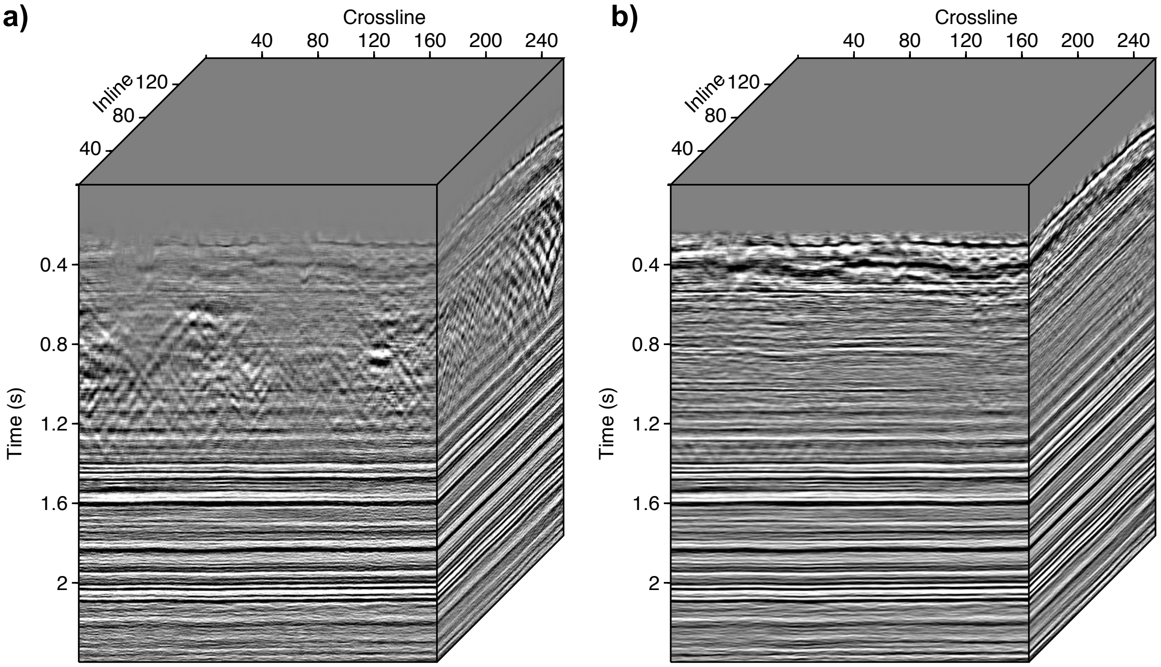


Figure 2.8: Post-stack PP-wave data. (a)The stacking of raw PP-wave data. (b) The stacking of PP-wave data after 5D reconstruction.

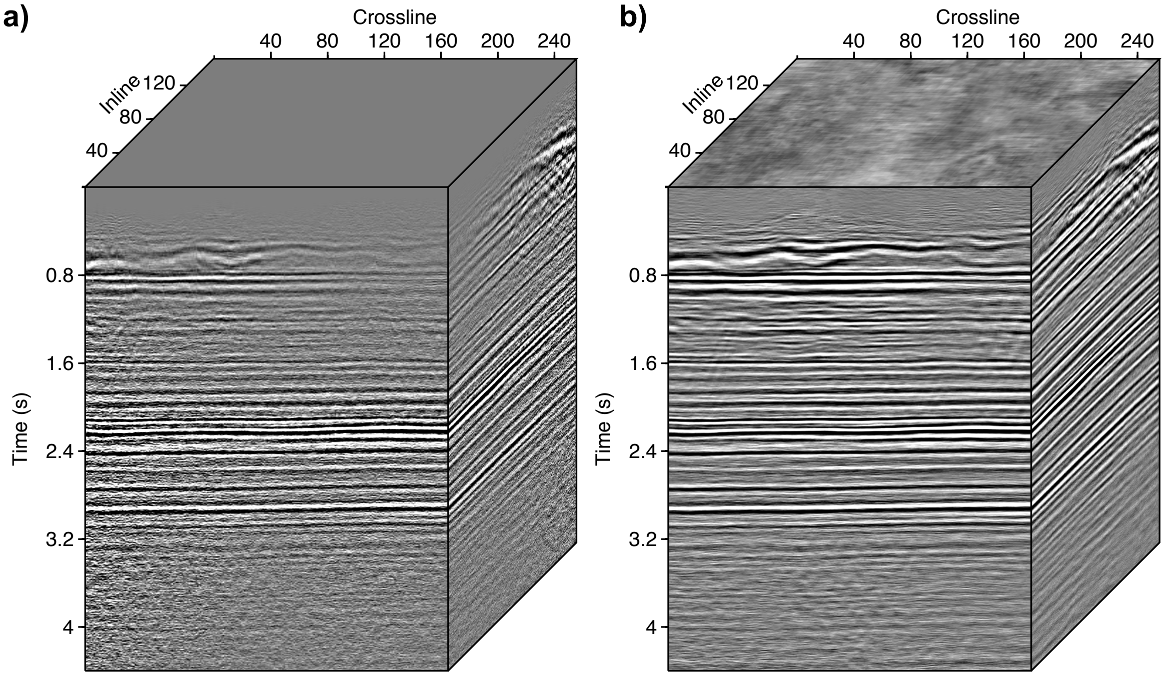


Figure 2.9: Post-stack PS-wave data. (a) Stacking of raw PS-wave data. (b) Stacking of PS-wave data after 5D reconstruction.

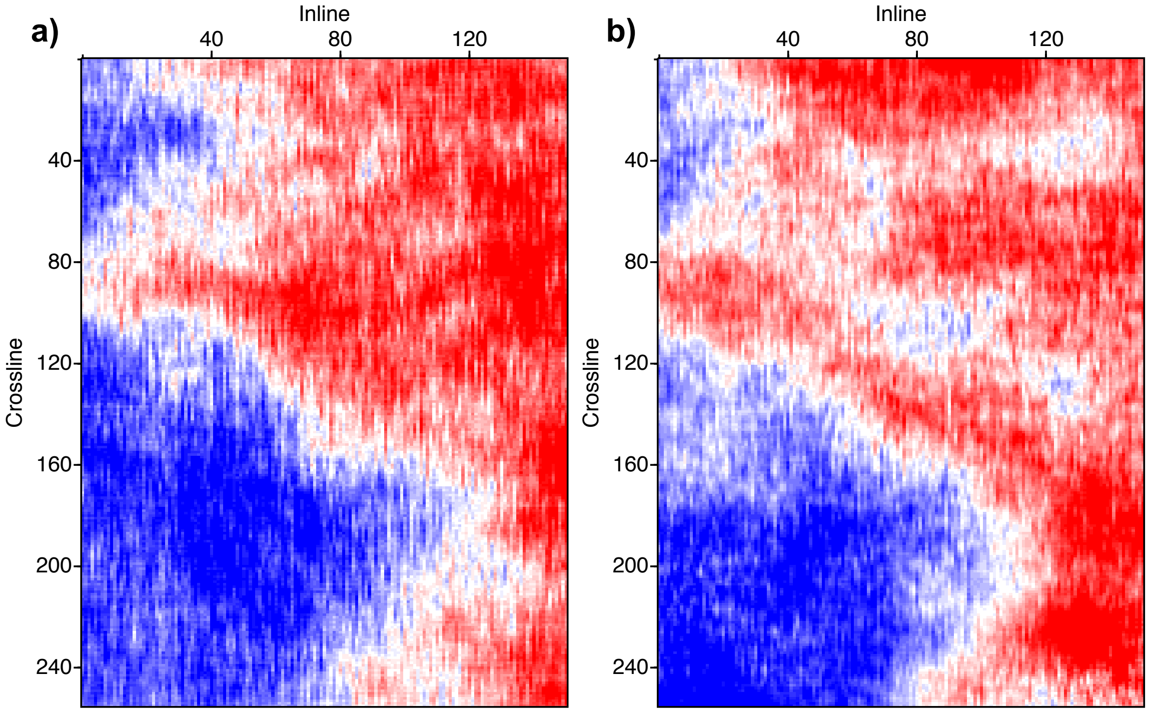


Figure 2.10: The results of 3D post-stack multi-component seismic data registration. (a) The time slice of post-stack PP-wave data at tiem of 2.2 s. (b) The corresponding time slice of the warped PS-wave data.

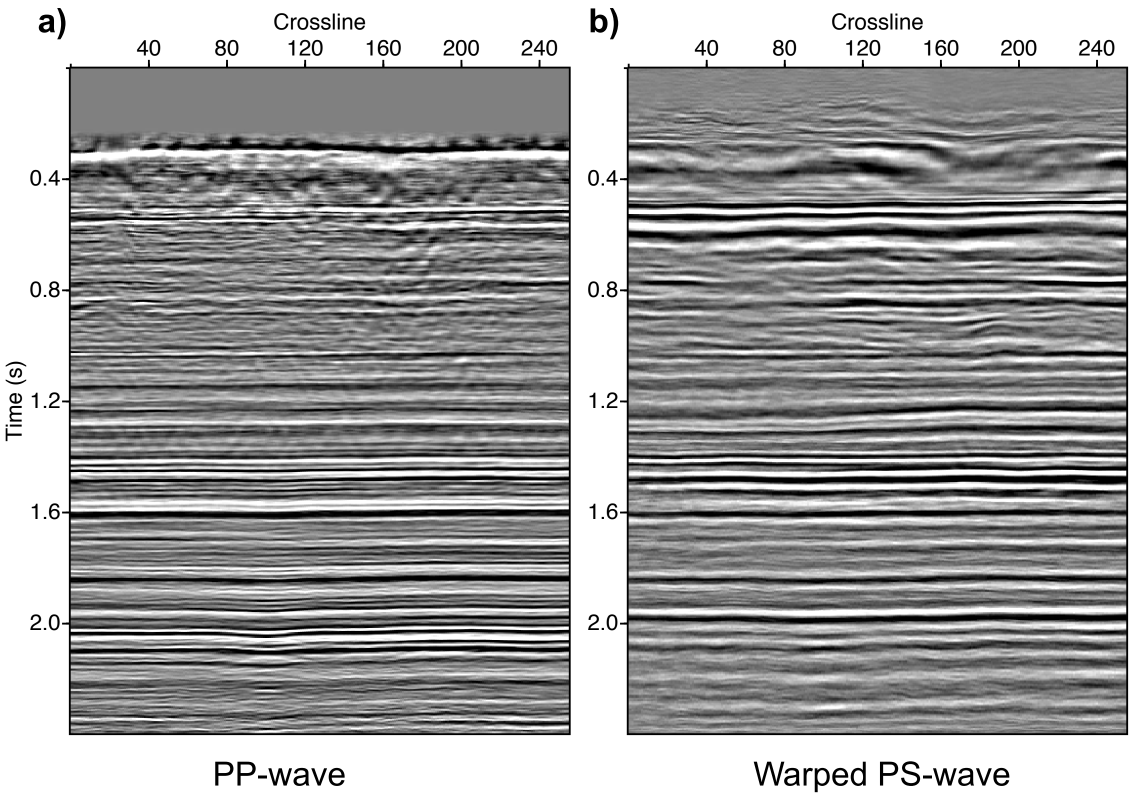


Figure 2.11: The results of 3D post-stack multi-component seismic data registration. (a) The inline sections of post-stack PP-wave data , (b) the corresponding inline section of the warped PS-wave data.

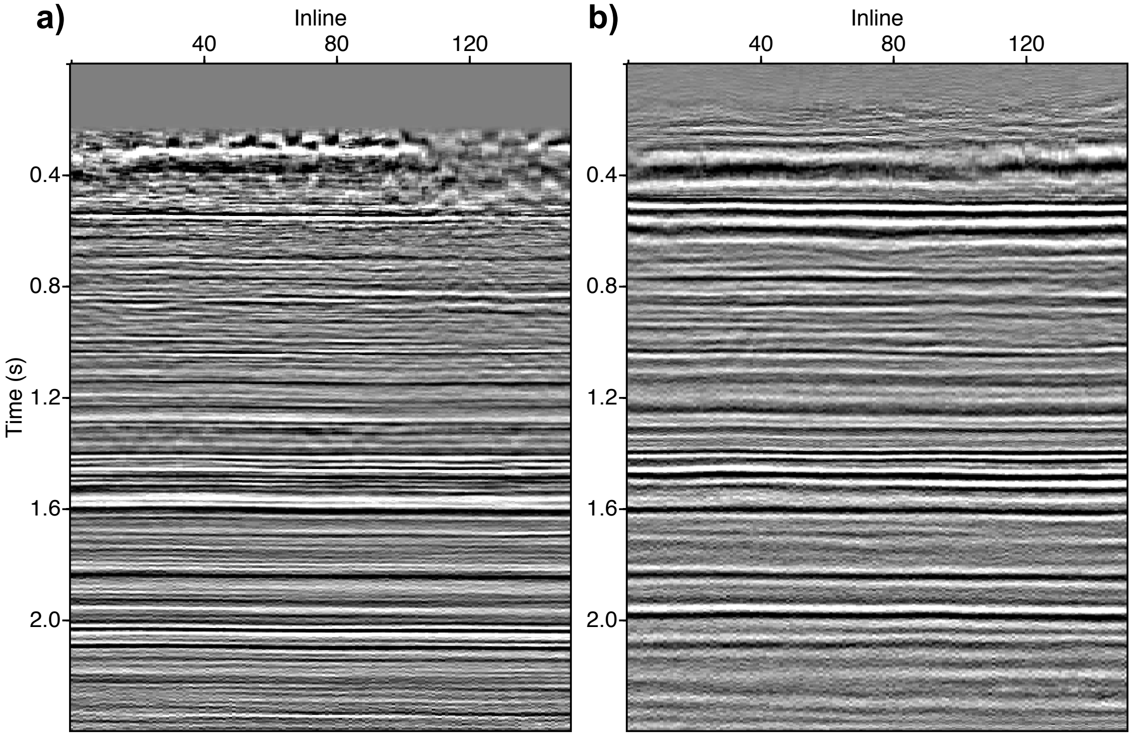


Figure 2.12: The results of 3D post-stack multi-component seismic data registration. (a) cross-line section of post-stack PP-wave data with in-line number 128 (b) The corresponding cross-line section of the warped PS-wave data.

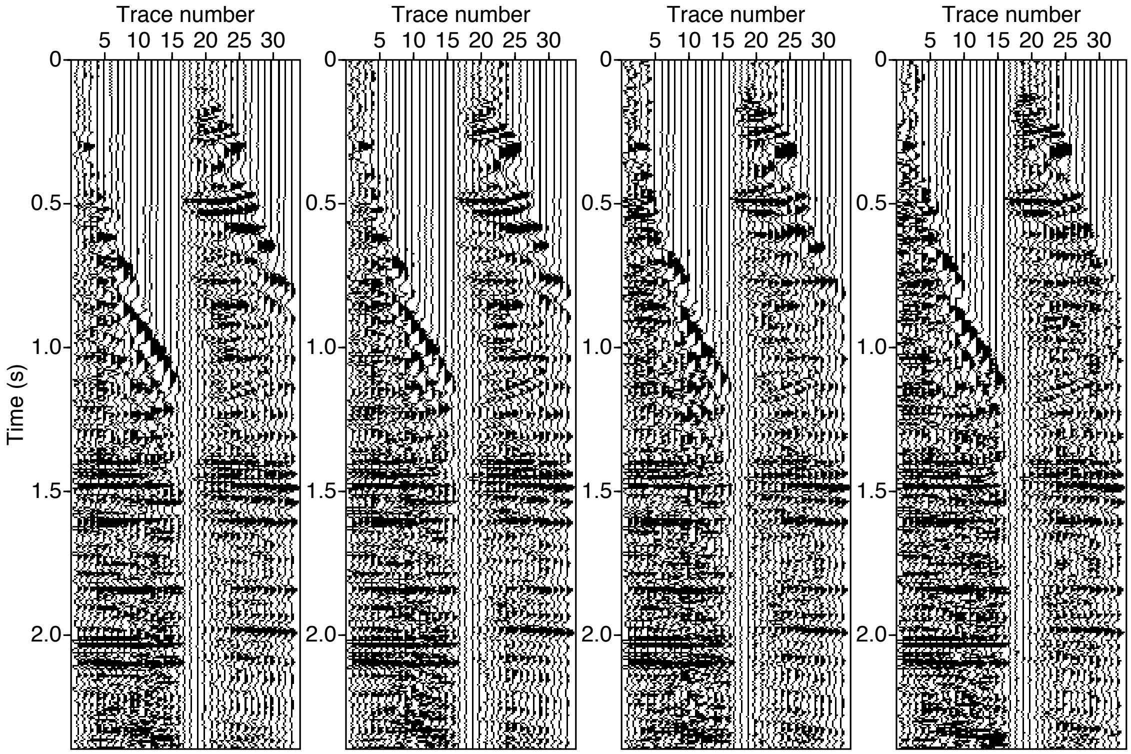


Figure 2.13: The result of the pre-stack registration. In each panel, the left half shows the regularized pre-stack PP-wave data and right half shows the warped pre-stack PS-wave data.

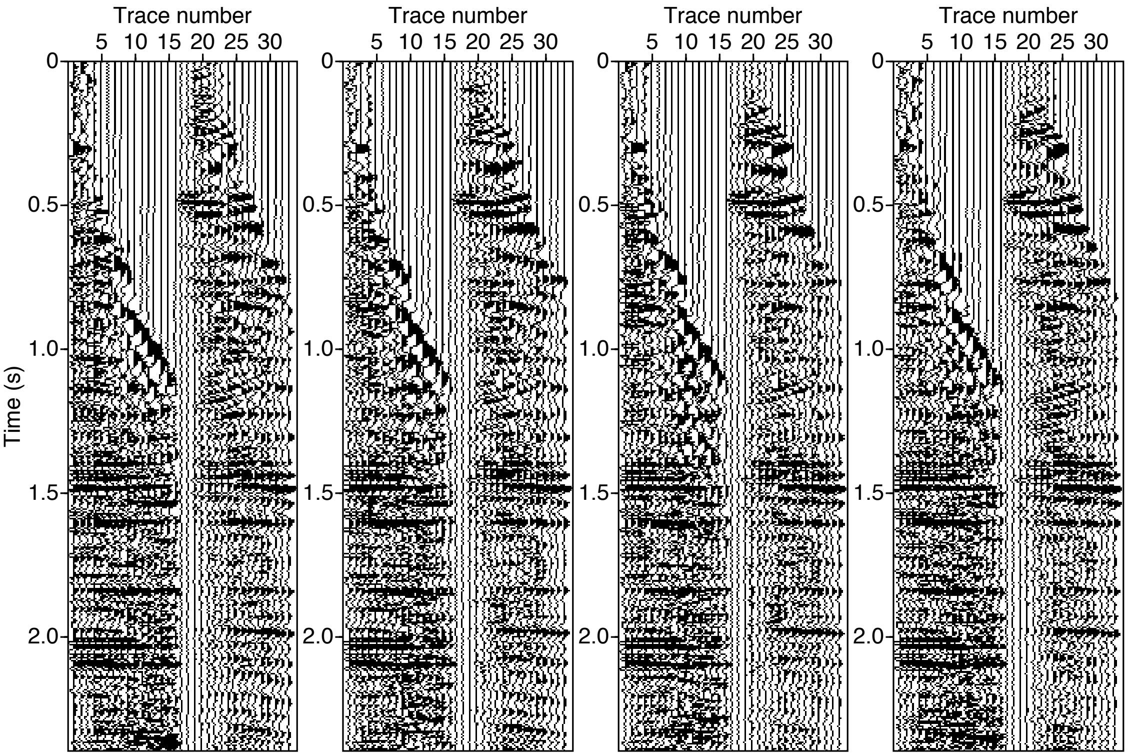


Figure 2.14: The result of pre-stack registration implemented on regularized seismic data. In each panel, the left half shows the regularized pre-stack PP-wave data and right half shows the warped pre-stack PS-wave data.

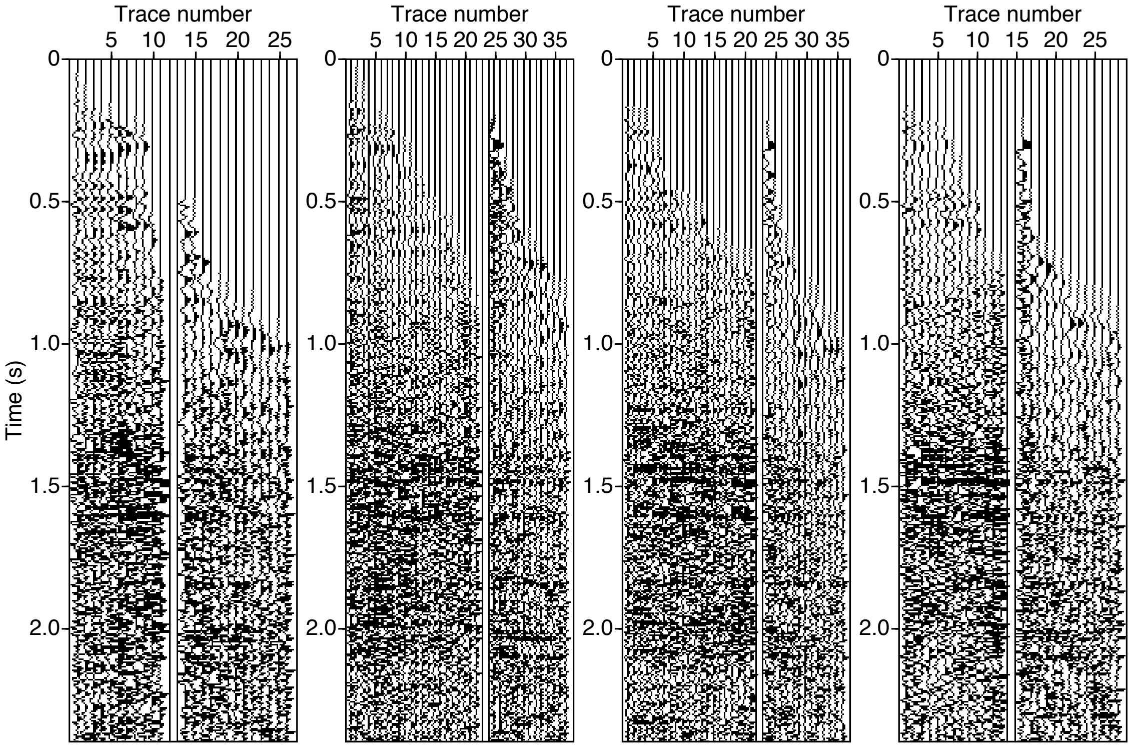


Figure 2.15: The results of pre-stack registration implemented on raw pre-stack seismic data. In each panel, the left half shows the raw pre-stack PP-wave data and right half shows the warped pre-stack PS-wave data.

2.4 Conclusion

We have presented a new workflow for multicomponent pre-stack seismic data registration. This workflow mainly consists of three steps: $5D$ reconstruction is implemented separately to PP-wave and PS-wave data to regularize the pre-stack seismic data and improve the SNR, then the registration algorithm is performed on the post-stack data to obtain a proper initial estimation of the warping function. Finally, a constrained registration is applied to the reconstructed pre-stack seismic data. The proposed algorithm was tested on a field $3D-3C$ data sets and the effectiveness of the workflow is demonstrated.

References

Compton, S., and D. Hale, 2014, Estimating vp/vs ratios using smooth dynamic image warping: Geophysics, **79**, V201–V215.

Fomel, S., 2007, Local seismic attributes: Geophysics, **72**, A29–A33.

Fomel, S., M. Backus, K. Fouad, B. Hardage, G. Winters, et al., 2005, A multistep approach to multicomponent seismic image registration with application to a west texas carbonate reservoir study: 75th annual international meeting: SEG, Expanded Abstracts, 1021.

Fomel, S., M. M. Backus, et al., 2003, Multicomponent seismic data registration by least squares: Presented at the 2003 SEG Annual Meeting, Society of Exploration Geophysicists.

Gaiser, J. E., 1996, Multicomponent vp/vs correlation analysis: Geophysics, **61**, 1137–1149.

Gao, J., A. Stanton, and M. D. Sacchi, 2015, Parallel matrix factorization algorithm and its application to 5d seismic reconstruction and denoising: Geophysics.

Gao, W., and M. D. Sacchi, 2017, Multicomponent seismic data registration by non-linear optimization: Geophysics, **83**, 1–35.

Geis, W. T., R. Stewart, M. Jones, and P. Katapodis, 1990, Processing, correlating, and interpreting converted shear waves from borehole data in southern alberta: Geophysics, **55**, 660–669.

Hale, D., 2009, A method for estimating apparent displacement vectors from time-lapse seismic images: Geophysics, **74**, V99–V107.

Hall, S. A., 2006, A methodology for 7d warping and deformation monitoring using time-lapse seismic data: Geophysics, **71**, O21–O31.

Kolda, T. G., and B. W. Bader, 2009, Tensor decompositions and applications: SIAM review, **51**, 455–500.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, V113–V122.

Kreimer, N., A. Stanton, and M. D. Sacchi, 2013, Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction: Geophysics, **78**, V273–V284.

Liner, C. L., and R. G. Clapp, 2004, Nonlinear pairwise alignment of seismic traces: The Leading Edge, **23**, 1146–1150.

Nickel, M., L. Sonneland, et al., 2004, Automated ps to pp event registration and estimation of a high-resolution vp-vs ratio volume: Presented at the 2004 SEG Annual Meeting, Society of Exploration Geophysicists.

Van Dok, R., P. Kristiansen, et al., 2003, Event registration and vp/vs correlation analysis in 4c processing: Presented at the 2003 SEG Annual Meeting, Society of Exploration Geophysicists.

Yuan, J. J., G. Nathan, A. Calvert, and R. Bloor, 2008, Automated c-wave registration by simulated annealing, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 1043–1047.

Robust 5D tensor completion via randomized QR decomposition

F. Carozzi[1] and M.D. Sacchi

Abstract

5D seismic reconstruction entails the denoising and recovery of missing elements of a multilinear array. Much effort has been done in the past few years in developing these techniques for pre-stack data because they enhance the input image to other techniques as FWI, migration and AVO analysis. We propose a new SVD-free tensor completion method that expands the suite of 5D tensor completion techniques. The technique solves an optimization problem constrained by a low-rank approximation to the different tensor unfoldings. Such approximation is calculated via randomized QR decomposition. We explain the method and analyze results on synthetic and real seismic data including non-gaussian noise.

3.1 INTRODUCTION

Tensors have been studied since the nineteenth century but it was not until the beginning of the twentieth century that their benefits were acknowledged (Cichocki et al., 2015). This change in paradigm can be attributed to the development of tensor decomposition techniques within different fields of research.

[1] Email: carozzi@ualberta.ca

In the past few years, diverse areas have found applications to these techniques, such as social sciences, telecommunications, biomedical engineering, data mining, food industry, computer vision, neuroscience, and machine learning, among others. In particular, signal processing has undergone major developments by adopting multi-linear algebra models (Sidiropoulos et al., 2017). In the field of seismic data processing, simultaneous denoising, and reconstruction of seismic data there has been notable progress in the application of tensor completion techniques.

Regularization of seismic data, that is denoising and reconstruction, can be beneficial for full waveform inversion, migration, AVO and AVAz analysis, multiple suppression, and shear wave splitting analysis. In addition, reconstruction can attenuate artifacts that arise from improper wavefield sampling. The core concept related to seismic data reconstruction using tensor decomposition techniques is that properly sampled, noise free, multidimensional data, can be algebraically represented by a low-rank tensor (Kreimer et al., 2013; Gao et al., 2013). Therefore, missing data and noise increase the rank of the multidimensional array. Therefore, it is possible to regularize real pre-stack seismic data by iteratively finding low-rank data structures that honor the original observations. A similar principle applies to methods based on Hankelization of the observed seismic data in the frequency-space domain (Trickett et al., 2010; Oropeza and Sacchi, 2011).

Recently, many authors have contributed to the development of denoising seismic data methods through tensor rank reduction techniques. In particular, Kreimer and Sacchi (2012) presented a HOSVD method that operates in the frequency-space domain. Also, Kreimer et al. (2013) introduced nuclear norm minimization to reconstruct prestack 5D seismic data. In addition, Gao et al. (2015) adopted the Parallel Matrix Factorization (PMF) algorithm (Xu et al., 2015), which performs low-rank matrix factorization to all the matrices in which one can unfold the tensor. Moreover, Ely et al. (2015) showed the results of applying tensor Singular Value Decomposition (tSVD) (Kilmer and Martin, 2011; Martin et al., 2013) on seismic data. Finally, it is worth mentioning that on the field of matrix rank reduction, López et al. (2016) incorporated a regularization operator into the nuclear norm minimization problem which allows off-the-grid data reconstruction.

In this article, the regularization method for pre-stack seismic volumes is posed as a convex optimization problem, where the low-rank approximation is a constraint to the objective function. The completion problem aims to find the fully sampled, noise-free tensor that best fits the data for each temporal frequency slice. To achieve this goal, the proposed algorithm iteratively estimates a low-rank tensor that simultaneously attenuates incoherent noise and reconstructs the original data. More importantly, the low-rank approximation is based on randomization techniques, where the size of the problem is severely decreased by randomly projecting the data into a subspace that properly preserves variability. For this technique, the determination of the rank of the tensor is no longer an input parameter. This characteristic of the method alleviates a fundamental constraint of SVD-based techniques where precise rank of the unkown data structure needs to be known.

In addition, field data contains many signals not related to the data of interest. Cultural noise, recording and parity errors, uncorrected polarity reversals, isolated noise bursts, misfired shots, scattered shot noise, poor surface conditions, disabled or poorly coupled geophones, to mention some of the most relevant (Trickett et al., 2012). To model these events, erratic noise with non-Gaussian distribution should be explicitly considered in the cost function, (Sternfels et al., 2015). Thus, we propose to model the data using robust objective functions.

The paper is organized as follows. First, we establish the basics of multilinear algebra and the notation that will be used throughout the article. Then, we describe the problem setup and present the cost function to be minimized to solve the problem. Afterward, we present the method used to achieve the low-rank approximation of the tensor and optimization of the cost function. Following, we examine different robust penalty functions and solutions that properly handle erratic noise. Next, we study the behavior of the algorithm with synthetic examples, as well as with field data. In order to work with these examples, preprocessing, binning and patching of the data is necessary. Finally, we present conclusions extracted from this work.

3.2 THEORY

Multilinear Algebra

A tensor is a multidimensional array (Kolda and Bader, 2009). Even though tensors can be interpreted as high order generalizations of vectors and matrices, multilinear algebra presents characteristics that span richer applications. As a consequence, tensor algebra has become a developing field of mathematics since the early 20th century (Cichocki et al., 2015).

Tensor decomposition and models, that is decompositions applied to data arrays that can favor the interpretation of their properties, have seen a major research development since their introduction. In particular, Tucker (1964) and Carroll and Chang (1970), established the Tucker decomposition and Canonical Polyadic Decomposition (CPD) in psychometrics, while Harshman (1970) introduced the concept into linguistics.

Prestack seismic data can naturally be embedded into a 5D structure or tensor. The signal component from the input data presents low rank while the missing data and noise increase the rank of the structure. By performing tensor decomposition to the non low-rank observed data, the complete, noise-free volume can be reconstructed.

Notation and definitions

Tensors are indicated in this paper by calligraphic fonts ($\mathscr{X}$). The number of dimensions of the array is named order, ways or modes. Special cases are tensors of order two or matrices, which are symbolized with boldface capital letters ($\mathbf{X}$). Finally, vectors (order one tensors) are denoted by boldface lowercase letters ($\mathbf{x}$). Scalars are written with lowercase italic letters (x)

The elements of an array are denoted by subscripts. The ith element of a vector is x_i, the (i, j) of a matrix is x_{ij} and the (i, j, k) element of a third order tensor is x_{ijk}.

Unfolding, matricization or flattening refer to the action of reordering the elements of a higher order tensor into a matrix. The mode-k matricization of a Nth order tensor arranges its fibers (higher-order analog of matrix rows and columns) into the columns of a $I_k \times (I_1 I_2 \ldots I_{k-1} I_{k+1} \ldots I_N)$ matrix (Kolda and Bader, 2009). This process is symbolized by $\mathbf{X}_{(n)}$. A fourth order tensor can be unfolded in four different matrices. Similarly, matrices can be folded into tensors. See Figure 3.1 for a graphical explanation of this concept.

The Tensor Frobenius norm is defined as the square root of the sum of the squares of all its elements (Kolda and Bader, 2009). This is analogue to the matrix Frobenius norm and can be mathematically expressed as

$$\|\mathscr{X}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} |x|^2_{i_1 i_2 \ldots i_N}} \;, \tag{3.1}$$

where $i_1, \ldots, I_1$ indicate the range of the indices.

Finally, the concept of Tensor rank is essential to this work. The polyadic decomposition of the nth order tensor $\mathscr{X} \in \mathbb{R}^{I_1 x \ldots x I_N}$ is any representation having the form

$$\mathscr{X} = \sum_{r=1}^{R} \lambda_r \mathbf{b_r^{(1)}} \circ \ldots \circ \mathbf{b_r^{(N)}} \;, \tag{3.2}$$

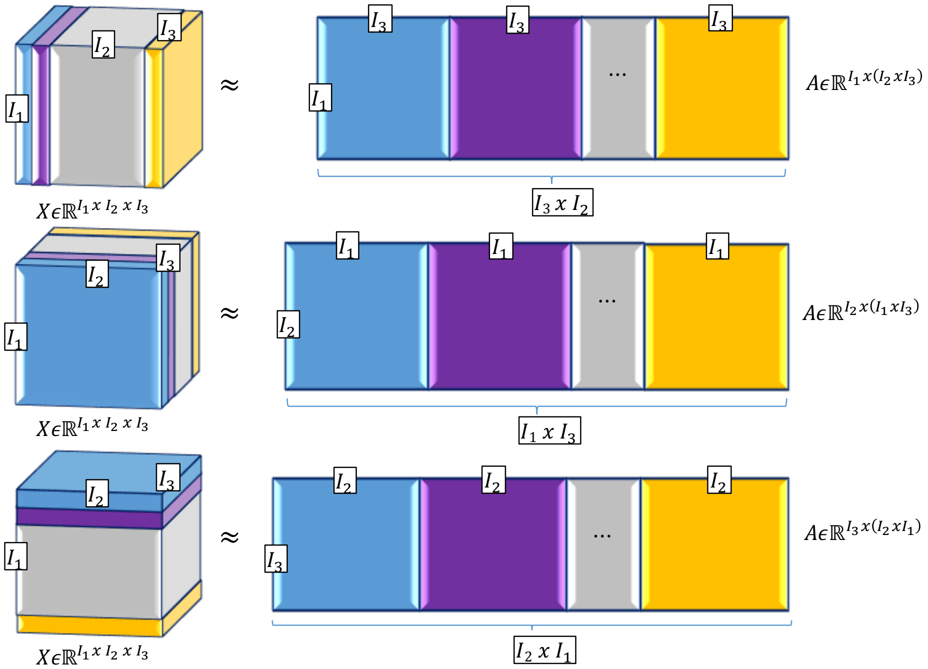

I_1
I_2
I_3
$X\epsilon\mathbb{R}^{I_1 x I_2 x I_3}$
≈
…
$A\epsilon\mathbb{R}^{I_1 x(I_2 x I_3)}$
$I_3 x I_2$
$A\epsilon\mathbb{R}^{I_2 x(I_1 x I_3)}$
$I_1 x I_3$
$A\epsilon\mathbb{R}^{I_3 x(I_2 x I_1)}$
$I_2 x I_1$

Figure 3.1: Tensor unfoldings.

where λ_r is a normalizing scalar, $\mathbf{b}^{(\mathbf{n})}$ represents a vector $\in \mathbb{R}^{I_n}$ and $\circ$ is the standard outer product of vectors. The rank of the tensor is given by the smallest integer R such that the decomposition exists. In this way, the tensor rank concept is related to the minimal representation of a tensor.

It can be concluded from equation 3.2, that the most outstanding difference between linear and multilinear algebra is the definition of rank and the properties related to it. In particular, for a general tensor, the rank depends on the field it is defined, and it can exceed the smallest dimension of the multilinear array.

3.2.1 Cost function and problem setup

Given the nature of the seismic acquisition method, data usually presents missing traces and noise. In contrast, ideal data is regularly gridded, properly sampled and noise free. Such data can be embedded into a low-rank block Hankel matrix (Trickett et al., 2010; Oropeza and Sacchi, 2011) or into a low-rank tensor (Kreimer et al., 2013; Gao et al., 2013). In contrast, field data is represented by a high-rank structure. Thus, pre-stack seismic data can be regularized by iteratively finding low-rank data structures that honor the original observations.

Seismic data is acquired in the source-receiver domain. That is, it can be represented as a function of source and receivers coordinates, s_x, s_y, r_x, r_y. In addition, data has a fifth dimension given by time or frequency. Likewise, spatial coordinates can be represented in the midpoint and offset coordinate system. We denote seismic data in the midpoint-offset, frequency-space domain by $\mathscr{D}^{obs}\left(\omega, x, y, h_x, h_y\right)$, where x, y are the spatial coordinates of the inline and crossline midpoint, and h_x, h_y are coordinates of the inline offset and crossline offset, respectively. Additionally, $\mathscr{X}\left(\omega, x, y, h_x, h_y\right)$ represents the ideal, low rank, seismic field. Temporal frequency axis do not need reconstruction or interpolation, then reconstruction of pre-stack seismic data in the $f-x$ domain bears a 4D problem. Therefore, for one particular frequency, the elements of a seismic volume can be represented by $D^{obs}_{x^i,y^i,h^i_x,h^i_y}$, where x^i, y^i, h^i_x, h^i_y are bins indices for the respective spatial coordinates. The reconstruction has to be performed iteratively for each temporal frequency slice.

Omitting the coordinates dependency for clarity, the model for data reconstruction can be expressed mathematically as

$$\mathscr{P} \circ \mathscr{X} \approx \mathscr{D}^{obs} \ , \tag{3.3}$$

where $\mathscr{P}$ is a sampling operator and $\circ$ is the Hadamard's or elementwise product. $\mathscr{D}^{obs}$, $\mathscr{P}$ and $\mathscr{X}$ are 4th order tensors. The problem is mathematically represented as a linear system of equations where the aim is to estimate $\mathscr{X}$, the complete data volume. As the reconstruction is performed in the $f-x$ domain, $\mathscr{X}$ and $\mathscr{D} \in \mathbb{C}^4$

To solve for $\mathscr{X}$, we minimize the Tensor Frobenius norm (Equation 3.1) of the error

$$\Phi_M = \frac{1}{2}\|\mathscr{P} \circ \mathscr{X} - \mathscr{D}^{obs}\|_F^2 \ . \tag{3.4}$$

Equation 3.4 is an inverse ill-posed problem as there exist an infinite number of solutions that might equally fit the data. To find a physical solution, we add a regularization term or constraint to the objective function. For the reconstruction problem, the regularization term considers a low-rank approximation to the unfoldings of the ideal tensor

$$\Phi_C = \frac{1}{2}\sum_{k=1}^{K}\|\hat{\mathbf{Z}}_{(k)} - \mathbf{Z}_{(k)}\|_F^2 \ . \tag{3.5}$$

Therefore, we express the cost function that allows the calculation of the regularized volume as

$$\begin{aligned}\Phi(\mathscr{X}) &= \Phi_M + \mu\Phi_C \\ &= \frac{1}{2}\|\mathscr{P} \circ \mathscr{X} - \mathscr{D}^{obs}\|_F^2 + \frac{\mu}{2}\sum_{k=1}^{K}\|\hat{\mathbf{Z}}_{(k)} - \mathbf{Z}_{(k)}\|_F^2 \ ,\end{aligned} \tag{3.6}$$

where μ is a regularization parameter.

Hence, tensor completion is a multidimensional optimization problem where an objective function conformed by the misfit between observed data and reconstructed volume, and a measure of the rank of the tensor is minimized. The solution to such problem is the approximated full tensor, free of incoherent noise, that best fits the observed data.

3.2.2 Low rank approximation

Various techniques have been developed for seismic reconstruction via low-rank approximation in the last decades. In particular, Gao et al. (2015) have addressed this problem by proposing a tensor completion method for 5D seismic data that adopts the Parallel Matrix Factorization (PMF) algorithm (Xu et al., 2015). The technique reconstructs the seismic volume by optimizing a cost function that combines a low-rank constraint and a data misfit term via the alternating least-squares (ALS) algorithm (Gabay and Mercier, 1976; Wen et al., 2012; Xu et al., 2015). The low-rank constraint is obtained through matrix factorization to all the matrices in which one can unfold the 5D seismic volume.

Most of the rank-reduction based techniques for reconstruction of seismic data consider the need to define the exact rank of the ideal array as an input parameter. This is a practical overburden for these methods. In this paper, we overcome this constraint by proposing the use of randomized QR decomposition (rQR) (Cheng and Sacchi, 2015; Halko et al., 2011), which does not require the rank information.

Randomization techniques decrease the size of the problem by randomly sampling the data to a subspace that captures most of the information provided. That is, drawing a set of n random vectors $\{\mathbf{x}^i, i = 1, \dots, n\}$, and multiplying them by the kth unfolding of the tensor $\mathscr{X}$ $\left(\mathbf{Z}_{(k)}, k = 1, \dots, K\right)$, we obtain a set of vectors $\{\mathbf{y}^i, i = 1, \dots, n\}$. As the first set is random, it is likely that they form a linearly independent set. As a result, the second set will also be linearly independent. Then, they span the image subspace of the matrix. Halko et al. (2011) present a review of the algebraic theory behind randomizing techniques.

To calculate the low rank approximation of the matrix $\mathbf{Z}_{(k)} \in \mathbb{C}^{m\times n}$, we begin projecting it to $\mathbf{M} \in \mathbb{C}^{m\times p}$

$$\mathbf{M} = \mathbf{Z}_{(k)}\boldsymbol{\Omega} \ , \tag{3.7}$$

where $\mathbf{\Omega} \in \mathbb{C}^{n \times p}$. $\mathbf{\Omega}$ is composed of a set of random and normalized column vectors $[\mathbf{v}_1, \ldots, \mathbf{v}_p]$, such that $p < n$. Each unfolding of the tensor can have a different p-value according to the complexity of the subsurface structure. We refer to this set of K p-values as p_K.

Following, we compute the orthonormal basis $\mathbf{Q} \in \mathbb{C}^{m \times p}$ with the economy-size QR decomposition

$$[\mathbf{Q}, \mathbf{R}] = \mathbf{M} \ , \tag{3.8}$$

where $\mathbf{R} \in \mathbb{C}^{p \times p}$.

Finally, we calculate the approximation of $\mathbf{Z}_{(k)}$ as

$$\hat{\mathbf{Z}}_{(k)} = \mathbf{Q}\mathbf{Q}^H \mathbf{Z}_{(k)} \ , \tag{3.9}$$

where the superindex H indicates the conjugate transpose or Hermitian transpose of the matrix. Since the ideal data is low-rank, the projection spans the reconstructed seismic volume.

To conclude, the main input parameter needed to solve the rank-reduction approximation via randomized QR decomposition is the number of random vectors p_k needed for each unfolding of the data tensor. As stated before, the rank of the method does not require the definition of the rank of the ideal data as an input parameter. This is an important advantage from a practical point of view for this technique in comparison with standard low-rank approximation techniques.

3.2.3 Robust penalty functions

Geophysical field data is usually plagued with noise. Seismic data are not the exception to this. In order to solve for the sought model, Equation 3.4 considers a quadratic norm whose penalty and influence functions are plotted on Figure 3.2 (a) and (b). From both graphs, it can be seen that the greater the error, the bigger is the weight assigned to such value. This means that outliers, which present large errors by definition, contribute more to the model. In this sense, in presence of non-Gaussian or erratic noise, the quadratic norm can yield inaccurate and unstable estimates of the solution.

This issue has been addressed in the past by pre-processing the data with an outlier rejection method, or by applying damping least-squares inversion. These methods have not found many applications as they do not achieve high-quality robust solutions.

In contrast, the objective function can be designed considering smooth, robust solutions. This is achieved through distributions with longer tails than those presented by a Gaussian distribution. In this way, large errors are part of the model yielding a solution that is not heavily affected by them. This characteristic can also be analyzed from Figure 3.2, where penalty functions for non-quadratic norms assign smaller weights to larger errors. In addition, influence functions tend to zero when errors grow to bigger absolute values.

Finally, we propose analyzing a new objective function given by

$$\Phi(\mathscr{X}) = \phi\left(\mathscr{E}\right) + \frac{\mu}{2}\sum_{k=1}^{K} \|\hat{\mathbf{Z}}_{(k)} - \mathbf{Z}_k\|_F^2 \ , \tag{3.10}$$

where the data misfit is measured by a robust penalty function, $\phi\left(\mathscr{E}\right)$, and $\mathscr{E} = \mathscr{P} \circ \mathscr{X} - \mathscr{D}$. Expressions for the $\phi\left(\mathscr{E}\right)$ can be found on Table 3.1. K refers to the order of the tensor which is equal to 4 in our problem. μ is a regularization parameter, σ is a tunning parameter for the penalty function, and $\mathscr{E}$ is the error, previously defined.

Minimizing the cost function

The cost function obtained in the previous section is non-linear and can be minimized using the alternating- least squares algorithm. ALS consists in calculating the best reduced rank approximation to the tensor in each iteration and a subsequent update of the reconstructed data.

In this way, we obtain the approximation to $\mathscr{X}$ as

$$\mathscr{X}^{i+1} = (\mathscr{I} - \mathscr{A} \circ \mathscr{P}) \circ \mathscr{C} + \mathscr{A} \circ \mathscr{D}^{obs} \ , \tag{3.11}$$

where

$$\mathscr{C} = \frac{1}{K}\sum_{k=1}^{K} fold_k\left(\hat{\mathbf{Z}}_{(k)}\right) \ .$$

Tensor $\mathscr{I}$ is a Kth order tensor with all coefficients equal to one. In our problem, $K = 4$.

Equation 3.11 is a weighted combination of the mean of the rank-reduced foldings of the tensor approximated in the previous iteration, and the observed seismic data. The initialization value for $\mathscr{X}$ is $\mathscr{D}^{obs}$. Tensor $\mathscr{A}$ depends on the penalty function used in the model. Table 3.2 shows the expressions for $\mathscr{A}$ for different robust penalty functions.

M-estimator	**Penalty function**	**Influence function**
Frobenius	$\frac{1}{2}\lvert x\rvert_2^2$	$\frac{x}{\sigma}$
l_1/l_2	$\frac{2}{\sigma}\sqrt{\sigma^2+\lvert x\rvert^2}$	$\frac{1}{\sigma}\left(\frac{x}{\sqrt{\sigma^2+\lvert x\rvert^2}}\right)$
Cauchy	$\frac{1}{\sigma}\ln\left(1+\frac{\lvert x\rvert^2}{\sigma^2}\right)$	$\frac{1}{\sigma}\left(\frac{x}{\sigma^2+\lvert x\rvert^2}\right)$
GermanMcClure	$\frac{1}{\sigma^3}\frac{\lvert x\rvert^2}{\sigma^2+\lvert x\rvert^2}$	$\frac{1}{\sigma}\left(\frac{x}{(\sigma^2+\lvert x\rvert^2)^2}\right)$

Table 3.1: Frequently used M-estimators for robust estimation. Mathematical symbols are explained in the text.

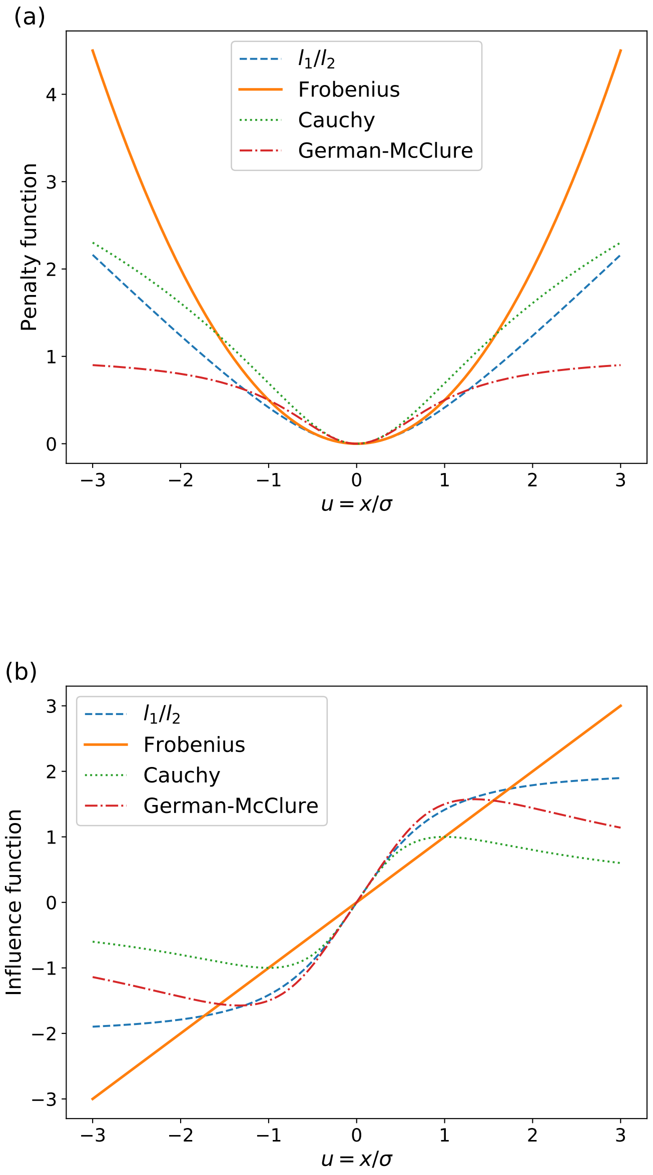


Figure 3.2: Penalty (a) and influence (b) functions of the M-estimators presented in Table 3.1.

M-estimator	Weight tensor
Frobenius	$\frac{1}{1+K\mu}$
l_1/l_2	$\frac{1}{1+K\mu\sqrt{\sigma^2+\lvert\mathscr{E}\rvert^2}}$
Cauchy	$\frac{1}{1+K\mu(\sigma^2+\lvert\mathscr{E}\rvert^2)}$
GermanMcClure	$\frac{1}{1+K\mu(\sigma^2+\lvert\mathscr{E}\rvert^2)^2}$

Table 3.2: Weight tensor expressions (Equation 3.11). Mathematical symbols are explained in the text.

In addition, the final expression for reconstruction is analog to the Cadzow-based denoising and reconstruction algorithm used by Oropeza and Sacchi (2011), where the rank reduction is applied to Hankel matrices constructed from the seismic data. Also, the POCS reconstruction method (Abma and Kabir, 2006) presents a similar expression, but in this case reconstruction is achieved through frequency-wavenumber thresholding. Finally, PMF (Gao et al., 2013) presents a similar expression.

3.2.4 Final Tensor Completion algorithm

The numerical solution of the objective function in equation 3.10 reduces to Algorithm 3.1.

Algorithm 3.1 Low-Rank tensor completion via rQR decomposition.

```
$\mathscr{D}^{obs}, \mu, \mathscr{P}, p_K, N_{iter}, rel, tol$
for $i = 1 : N_{iter}$ do
Projection operator via rQR:
    for $k = 1 : K$ do
        $\mathbf{Z}_k^i \leftarrow unfold_k\left(\mathscr{X}^i\right)$
        $\mathbf{M} \leftarrow \mathbf{Z}_k^i \boldsymbol{\Omega}$
        $\mathbf{Q}, \mathbf{R} \leftarrow qr[\mathbf{M}]$
        $\hat{\mathbf{Z}}_k^i \leftarrow \mathbf{Q}\mathbf{Q}^H \mathbf{Z}_k^i$
    end for
    $\mathscr{C} \leftarrow \frac{1}{K}\sum_{k=1}^{K} fold_k \hat{\mathbf{Z}}_k^i$
    $\mathscr{X}^{i+1} \leftarrow (\mathscr{I} - \mathscr{A} \circ \mathscr{P}) \circ \mathscr{C} + \mathscr{A} \circ \mathscr{D}$
end for
```

We remind the reader that the expression of the tensor $\mathscr{A}$ depends on the penalty function used to measure the error in the model.

3.3 EXAMPLES

3.3.1 Synthetic data

Noiseless examples

We begin by exploring a simple 5D model with four dipping planes to assess the behavior of the proposed algorithm. The objective of this simple experiment is to assess the behaviour of the proposed method. As a consequence, we only use Frobenius norm for the error functional in our cost function.

The synthetic volume corresponds to a spatial tensor of size $12 \times 12 \times 12 \times 12$, and 1000 time samples with a time sampling rate of 2 ms. The intersection of the planes with the vertical axis occurs at 0.5 s, 0.7 s, 1.0 s and 1.43 s.

To quantitatively compare the results among tests run with different parameters, we define a quality reconstruction parameter, QR, in dB units as

$$QR\ [dB] = 10 \log \frac{\|\mathscr{D}^{true}\|_F^2}{\|\mathscr{X} - \mathscr{D}^{true}\|_F^2}\ ,$$

where $\mathscr{D}^{true}$ is the complete, noise free data and $\mathscr{X}$ is the final reconstructed volume. If $QR = 2$ dB, $\|\mathscr{D}^{true}\|_F^2$ is 1.6 times bigger than $\|\mathscr{X} - \mathscr{D}^{true}\|_F^2$; while if $QR = 10$ dB, then the numerator is 10 times bigger than the denominator. Overall, we consider any result with a QR larger than 10 dB as an acceptable reconstruction.

In addition, we set an error less than 0.004 at each frequency for the convergence of the algorithm or a maximum number of 20 iterations. The band of frequencies used for reconstruction is 1 to 70 Hz.

Figure 3.3 shows the results for the reconstruction of noise-free data, 40% of which is decimated. The Frobenius penalty function is used with reconstruction parameters $\mu = 0.001$ and $p = 4$ for all unfoldings. Results show an effective performance of the method. In addition, the quality factor for this experiment is $Q = 26.5$ dB. Therefore, we can conclude that the low rank approximation via randomized techniques in the PMF method conveys appropiate results.

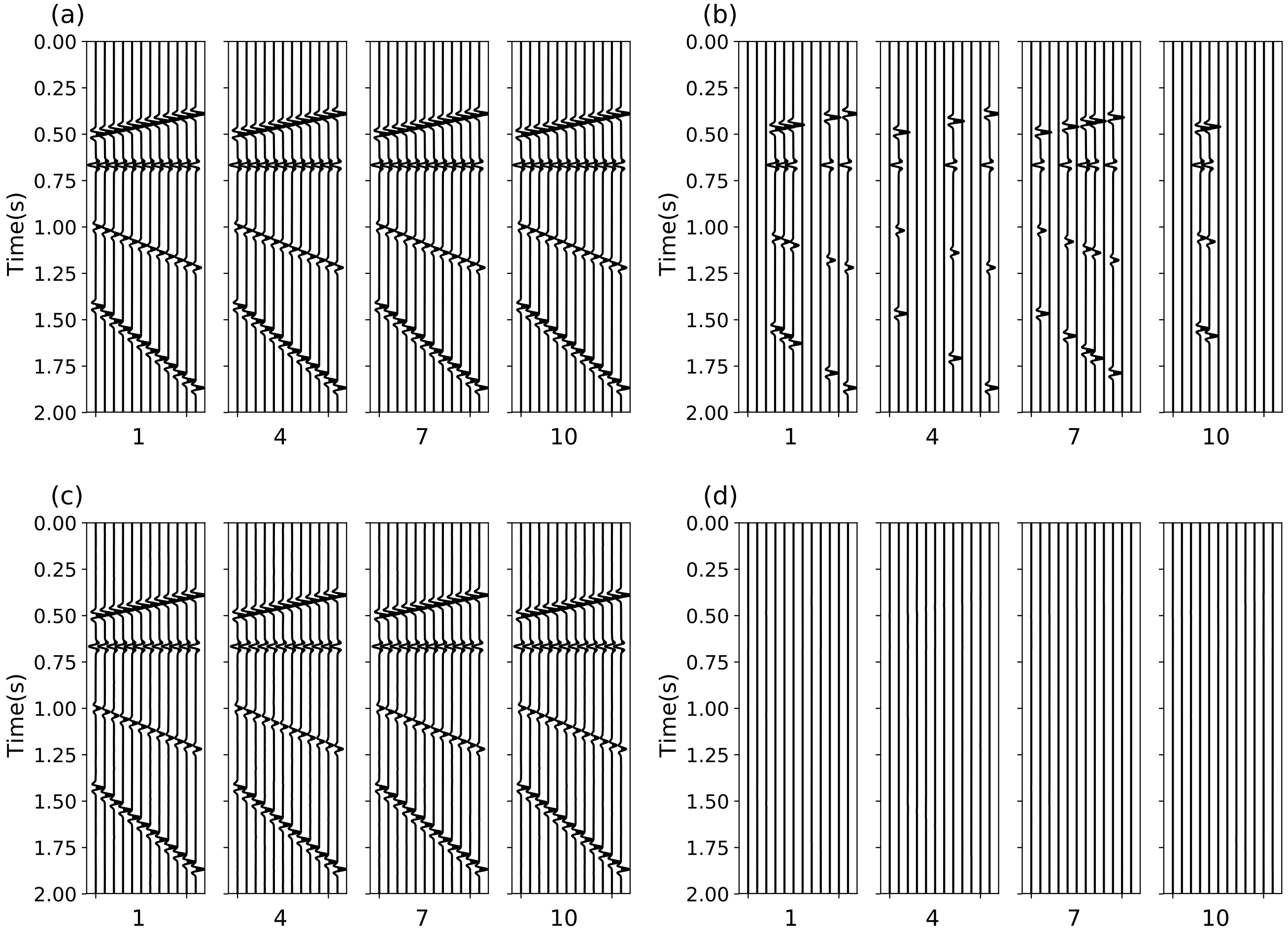


Figure 3.3: Reconstruction of noise-free linear events. (a) Original data. (b) Decimated data with 40% missing traces. (c) Reconstructed data. (d) Error. Each subplot shows four common midpoint gathers, each of which is enumerated with the corresponding inline number.

In their work, Stanton et al. (2012) and Gao et al. (2015) discuss that curved events preserve their low-rank property in a similar trend to linear events. In this sense, low-rank approximation methods could be used with the objective of denoising and reconstructing seismic volumes. With the aim of testing our method in such scenario, we propose the analysis of synthetic curved events. Considering this, we perform an experiment with the same parameters as those in Figure 3.3. Results are shown in Figure 3.4. From this figure, we can also conclude that PMF with randomized QR decomposition yields proper reconstruction results no only in windows were linear approximations can be considered but also in scenarios presenting curved events.

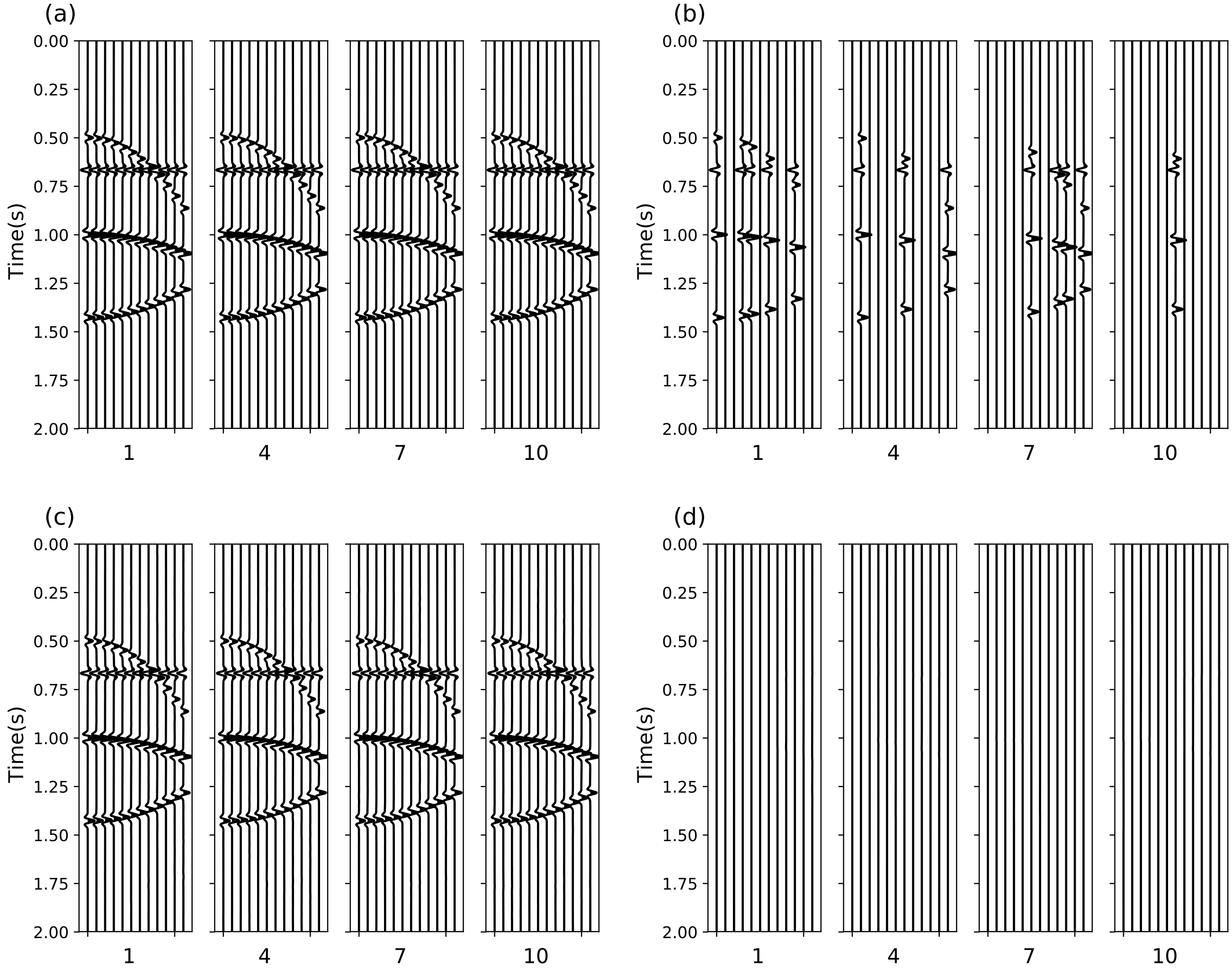


Figure 3.4: Reconstruction of noise-free hyperbolic events. (a) Original data. (b) Decimated data with 40% missing traces. (c) Reconstructed data. (d) Error.

Synthetic data contaminated with noise

Having tested the proposed method in noise-free synthetic volumes, we continue testing the reconstruction quality of the method for noisy traces. One of the main contributions of this paper is the evaluation of traces presenting non-Gaussian noise distribution. Thus, we design two different experiments presenting this kind of noise, as background noise or as outlier traces.

We begin by considering non-Gaussian background noise. Following the structure of the previous section, we design a synthetic seismic volume with the same parameters as those considered for the noise-less linear experiment. In addition, 40 % of the data is decimated. Following, a Gaussian mixture with $SNR = 1.5$, where $SNR = \frac{\|signal\|_F^2}{\|noise\|_F^2}$ is generated. The proposed gaussian mixture considers two random series each of which presents gaussian distribution with different variance. In addition, these series are blended in accordance to a pre-defined mixing parameter. In this sense, we design simple background noise that follows a non-Gaussian distribution, and we added it to the synthetic seismic volume.

As non-Gaussian noise is being considered, we perform the reconstruction under the Frobenius penalty function and different robust penalty functions, all of which are listed in Table 3.1. As expected, all robust calculations present analog results, then only those obtained using l_1l_2 penalty function are shown in the following figures. The selected parameters for these experiments are $\mu = 5$, $p = 6$ for all the unfoldings and a maximum of 100 iterations per calculated frequency. For robust penalty functions, there is an extra parameter, σ that has to be pre-defined. In this experiment and the following, $\sigma = 0.0001 * \|D^{obs}(\omega)\|_F$ for each considered frequency, ω. At the end of the present section we present a brief discussion on the selection of this parameter.

Results are shown on Figure 3.5 and Figure 3.5 continuation. In subplot (a), four common midpoint gathers of the ideal volume presenting four dipping events is portrayed. Subplot (b) presents the same gathers after decimation and addition of noise. From this subplot it can be observed that the added noise represents an important percentage of the total energy of the traces. Following, (c) and (d) show the reconstruction perform with the proposed method, considering Frobenius and l_1l_2 penalty functions, respectively. Next, subplots (e) and (f) are presented. These subplots reveal the error of the reconstruction with respect to the initial ideal volume. Detailed analysis of these images shows that Frobenius norm is not able to completely reconstruct the events embedded in the noisy traces. As previously showed in Figure 3.2, this effect is expected from this norm as it penalty function assigns bigger weights to greater errors. For a better visualization of this effect, subplot (g) is included, showing the difference between the errors calculated in subplot (e) and (f). In this visualization, the removal of part of the signal using Frobenius penalty function, with respect to l_1l_2 penalty function is more noticeable. From these results, it can be concluded that robust penalty functions outperform the standard Frobenius one for non-Gaussian noise distributions.

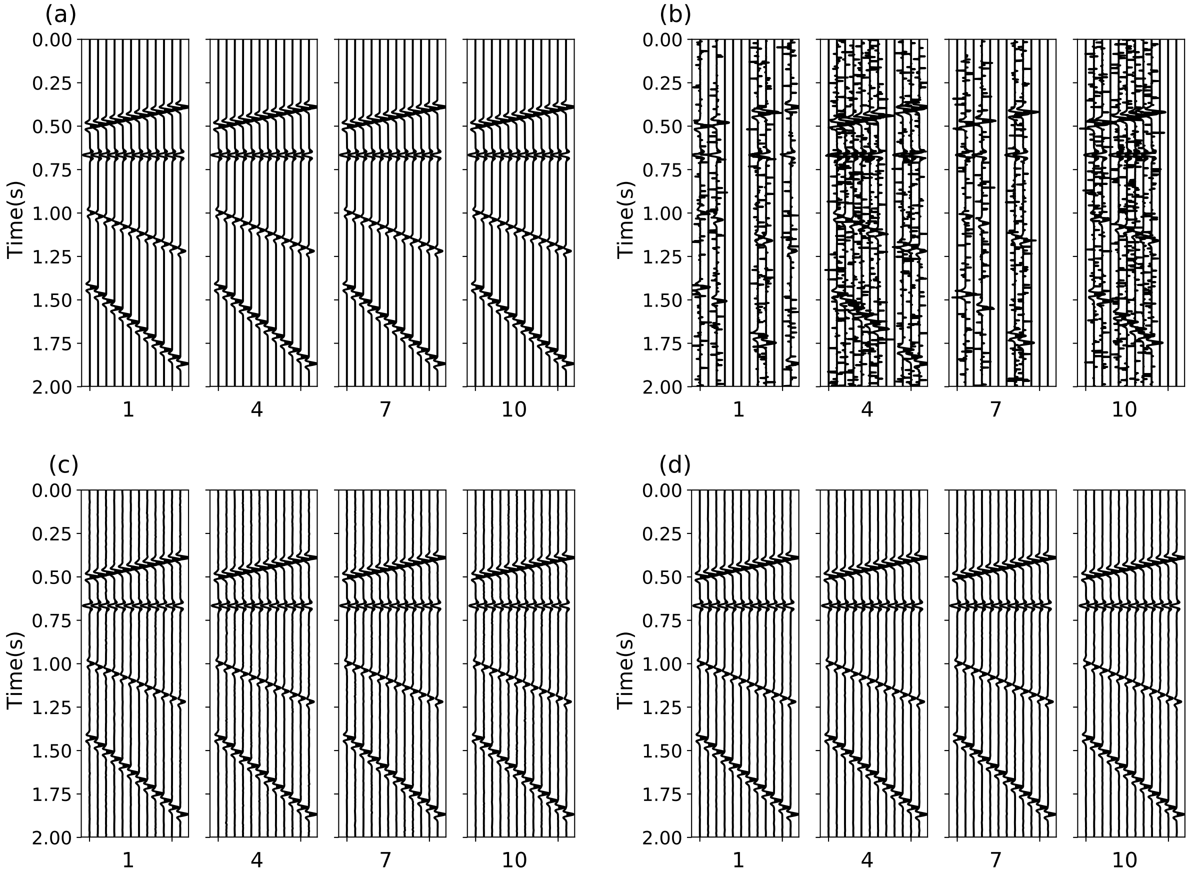


Figure 3.5: Reconstruction of linear events with non-Gaussian noise. (a) Original data. (b) Decimated data with 40% missing traces. Reconstructed data considering Frobenius (c) and l_1/l_2 (d) penalty functions, respectively.

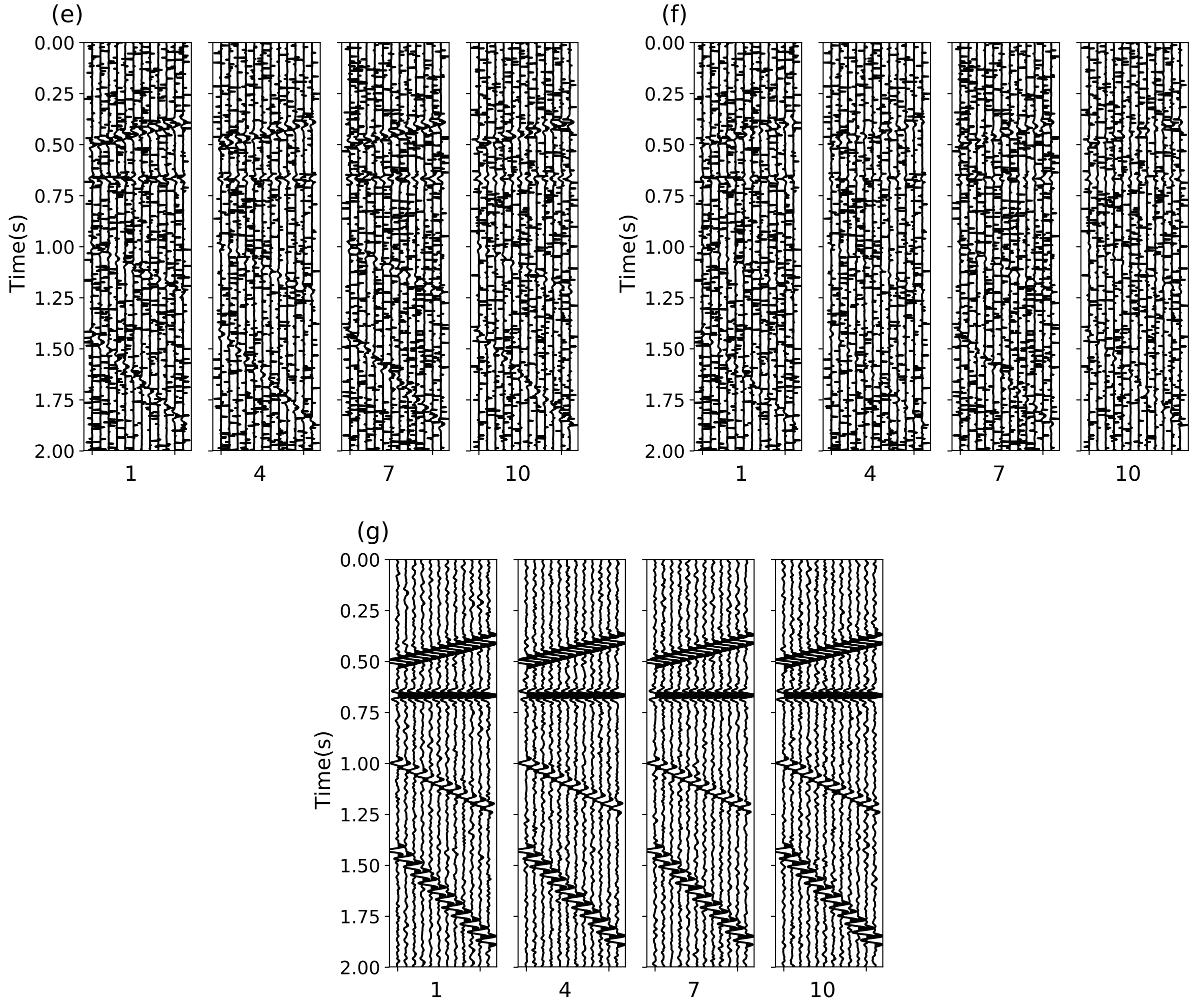


Figure 3.5 continuation. (e) and (f) error corresponding to the reconstructions in (c) and (d), respectively. (g) Difference between (e) and (f).

A final test is considered to evaluate the method in the presence of corrupted or outlier traces. The initial data is again modelled from 4 linear dipping events where non-gaussian noise is added to the ideal traces with $SNR = 0.5$. In addition, 5 % of the traces have been replaced by sinusoidal time series simulating incorrect measurements. Finally, 40 % of the data is decimated.

The reconstruction is performed considering $\mu = 1$, $p = 4$ for all the unfoldings and a maximum of 100 iterations per calculated frequency. For robust penalty functions, $\sigma = 0.0001 * \|D^{obs}(\omega)\|_F$ for each considered frequency, ω.

Results are shown in Figure 3.6. As in the previous examples, each subplot shows a common midpoint gather of the analyzed synthetic volume, two of them consider a constant CMP-x coordinate and varying CMP-y, while the other two consider constant CMP-y coordinate with moving CMP-x coordinate. Subplot (a) shows the ideal synthetic volume. Subplot (b) illustrates the input incomplete, noisy volume. This subplot shows the character of the outlier traces included in the data. As in the previous example, from these images, it can be seen that the noise represents a high percentage of the energy of each trace. Subplots (c) and (d) show the reconstructed result achieved via Frobenius and l_1/l_2 penalty functions, respectively. From these subplots (c) and (d), it can be concluded that the reconstruction with Frobenius penalty function is deficient in terms of denoising corrupted traces. Analyzing the errors, presented in subplots (e) and (f), it can be analyzed that once more, a robust penalty function is needed in those cases where non-Gaussian noise is present in the seismic volume, in particular if such noise is originated from outlier traces. In this case, the reconstructing algorithm is not able to denoise outlier traces when the standard Frobenius penalty function is used. In constrast to the previous example, where the non-Gaussian noise is present as a constant background component, it can be deduced that the Quality Reconstruction with both penalty functions is lower when outlier or local problematic traces are present. In any case, given that the time and computing resources needed to reconstruct the considered data are not affected by choice of penalty function, it is concluded that always a robust penalty function should be preferred over a Frobenius one, when non-Gaussian noise is present in the volume of interest.

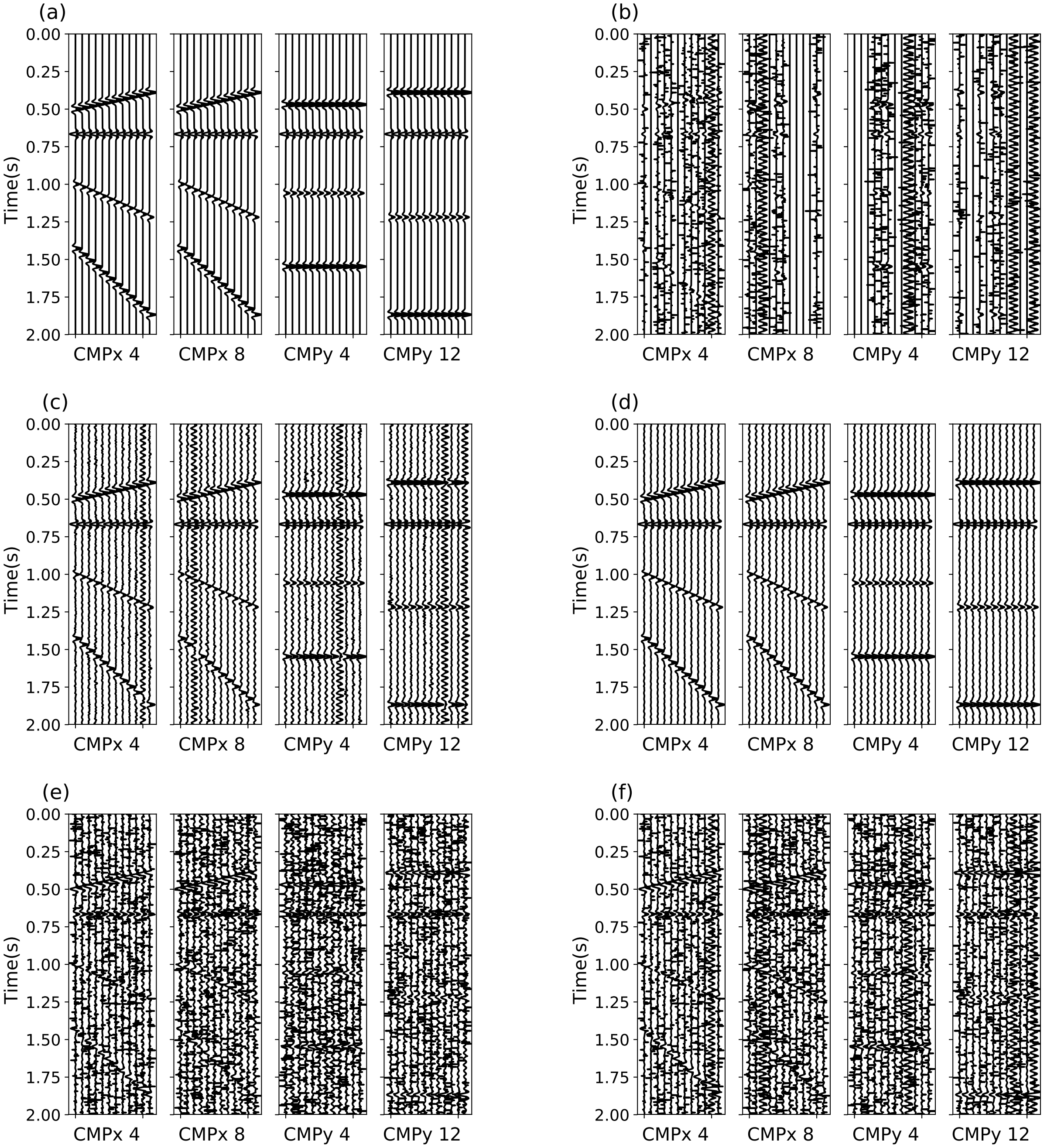


Figure 3.6: Reconstruction of linear events with non-Gaussian noise and corrupted traces. (a) Original data. (b) Decimated data with 40% missing traces. Reconstructed data considering Frobenius (c) and l_1/l_2 (d) penalty functions, respectively. (e) and (f) error.

SNR	**Decimation**	**Frobenius**		**l_1/l_2**		
		μ	QR [dB]	μ	σ	QR [dB]
0.5	10	5	19.1	5	0.001	24.4
0.5	40	1	15.5	1	0.0001	20
0.5	90	0.2	13.6	0.08	0.0001	13.3
1	10	4	19.7	4	0.001	25.8
1	40	1	16.9	0.9	0.0001	20.8
1	90	0.005	14.5	0.005	0.0001	14.5
10	10	1.1	22.5	1.1	0.0001	29.3
10	40	1.1	16.9	1.1	0.0001	22.7
10	90	0.005	15.4	0.005	0.0001	15.5

Table 3.3: Parameter selection for optimal QR considering Frobenius and l_1/l_2 penalty functions.

Discussion on parameter selection

Finally, the difficulty on how to practically determine σ for robust inversion has to be discussed. In order to do so, we design an experiment with 4 linear events. The ideal volume is analog to the one used in the previous experiments. The maximum number of iterations for each frequency is set to 100 and 4 random projection vectors were used in the rQR decomposition. The band frequency is 1 to 70 Hz. Non-Gaussian noise is added to the synthetic data using a Gaussian mixture. SNR varies from 0.5 to 20. Using line search, we aim to find the parameters μ and σ that optimize the final QR of the experiment. Table 3.3 shows QR for different SNR and decimation scenarios when reconstruction is performed considering Frobenius or l_1/l_2 penalty functions. From the results, it can be concluded that, even though σ has an impact on the final QR results, its variation is minimal. Most importantly, from this Table as from Figure 3.5, it can be concluded that robust penalty functions outperform Frobenius one.

3.3.2 Field data

Following, we test the algorithm on seismic data. The data was acquired to monitor a heavy oil field in Alberta, Canada. Sources and receivers were distributed following an orthogonal survey. Their locations are shown on Figure 7.3. The fold before binning is also shown in Figure 3.8. The data is NMO corrected to avoid spectral wrapping in the frequency-wavenumber domain. Moreover, a low-pass filter is applied to remove operation noise.

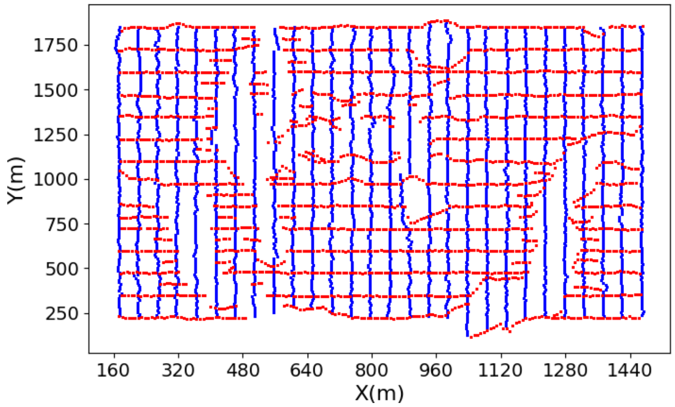


Figure 3.7: Survey acquisition geometry. Sources follow East-to-West lines. Receivers form North-to-South lines.

3D seismic data acquisition naturally yields to volumes that depend of 5 dimensions. The process call *Binning* permits to save the data in a 5D structure (a tensor).

Binning implies assigning each trace of a 3D survey to an appropriate cell of a 5D volume grid defined by the coordinates of the midpoint and offset and time of the corresponding trace (Vermeer, 2002). Reconstruction methods perform poorly on highly decimated data. In this sense, the parameters used to bin field data become of paramount relevance. Therefore, the main objective is to tailor a 5D grid that has a fewer number of redundancy traces per bin but, at the same time, presents most of its bins filled with information.

Proper binning parameters for the considered data are 5×5 m common midpoint (CMP) and 100 m offset by 45 degree in azimuth. Such grid presents approximately 80% decimation of traces. The binned area includes 274 CMPx bins and 27 CMPy bins. Binning is performed sequentially, that is, the trace kept for any given bin is the last trace evaluated for the corresponding cell.

Following, we define overlapping windows or *patches* in space and time. This technique is used to ensure stationarity of the wavefield. In addition, given the size of the considered volume, it benefits the computational effort. Each window consists of 45 CMPx bins by 27 CMPy bins. This configuration makes a total of 7 patches.

The reconstruction is performed on every patch with the same parameters. We consider a band frequency ranging from 1 Hz to 100 Hz, a maximum number of 50 iterations per considered frequency, $\mu = 1$ and $p = 5, 5, 18, 30$ for each unfolding, respectively.

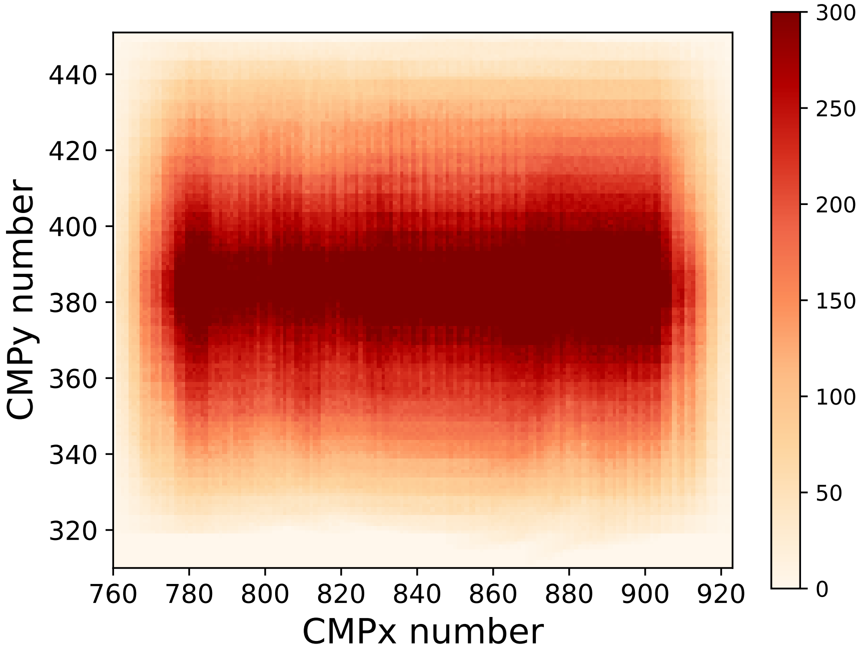


Figure 3.8: Fold map before binning.

Finally, an ensemble of the patches is done to recover the full 5D reconstructed volume. Figures 3.10 to 3.12 show slices of such volume before and after reconstruction considering Frobenius and robust norms.

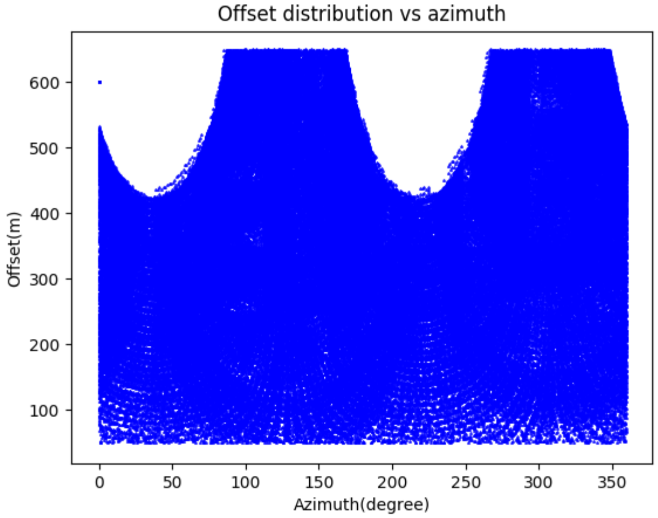


Figure 3.9: Offset vs. Azimuth distribution after binning.

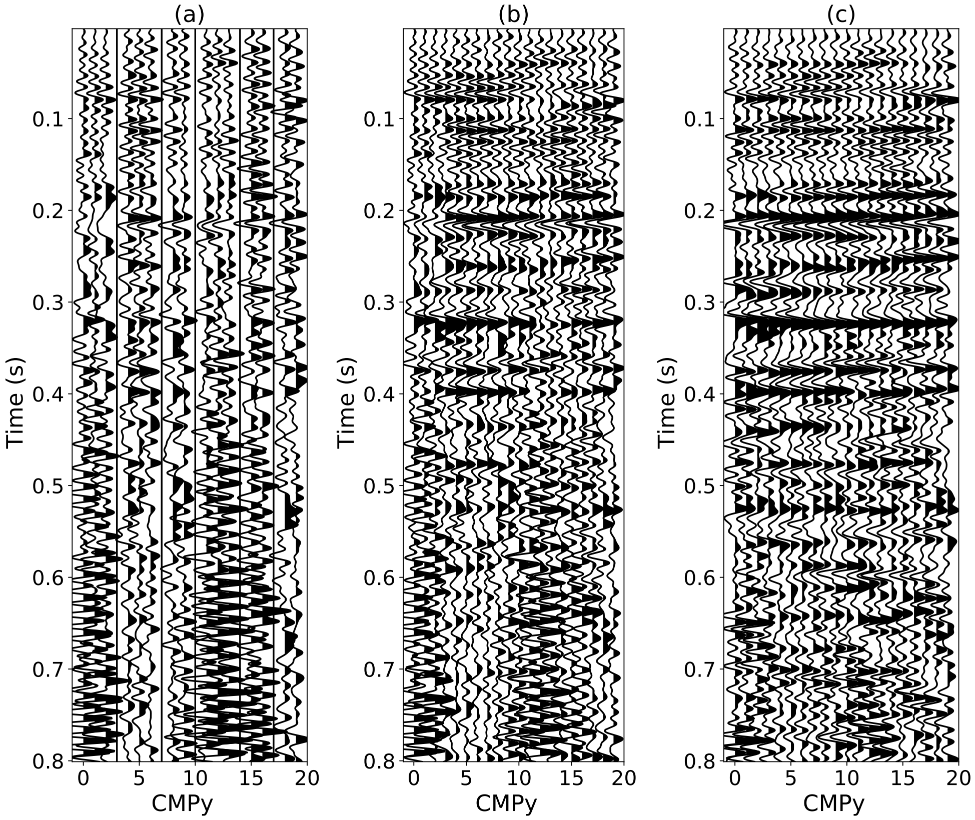


Figure 3.10: Reconstructed field seismic data for fixed CMPx. (a) Original data. (b) Reconstructed data using Frobenius penalty function. (c) Reconstructed data using l_1/l_2 penalty function.

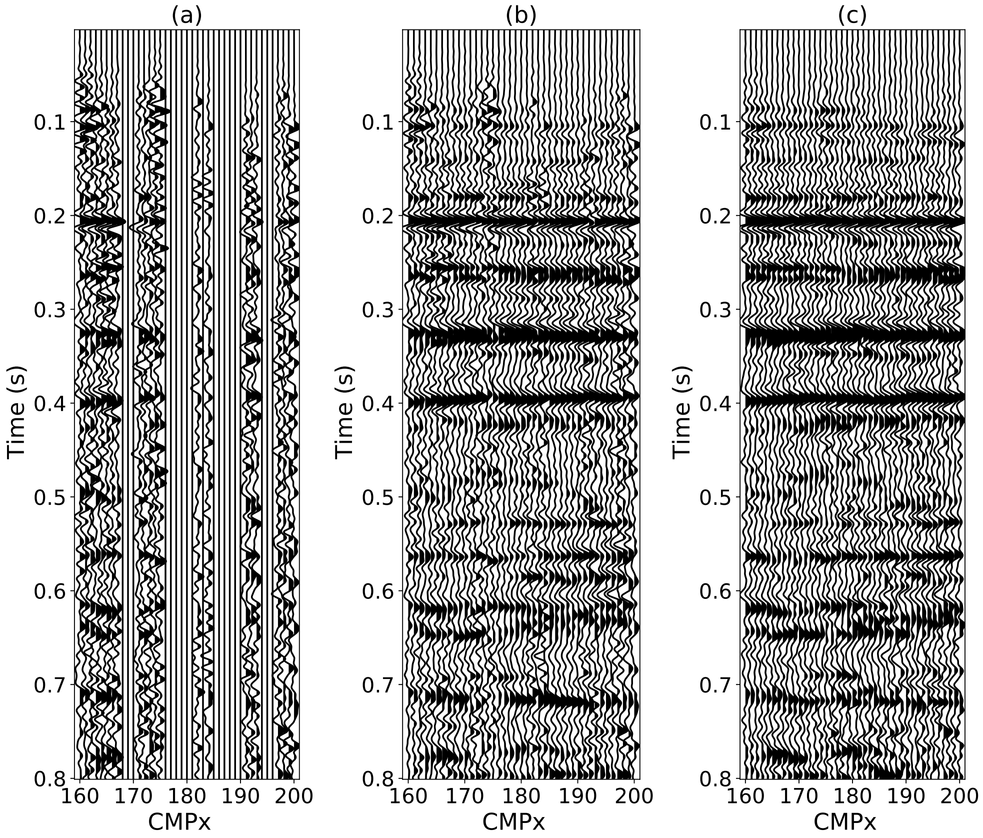


Figure 3.11: Reconstructed field seismic data for fixed CMPy. (a) Original data. (b) Reconstructed data using Frobenius penalty function. (c) Reconstructed data using l_1/l_2 penalty function.

It can be concluded that the reconstruction method improves overall SNR and provides continuity to events without generating bogus data. In addition, in the presence of non-Gaussian or outlier noise, the consideration of robust penalty functions for the evaluation of the model provides better and more robust results. These characteristics facilitate qualitative and quantitative seismic interpretation and improve seismic attribute extraction.

3.4 CONCLUSIONS

In this article, we study the regularization of prestack seismic data as an inverse problem. We propose an objective function composed of a misfit term and a constraint that considers a low-rank

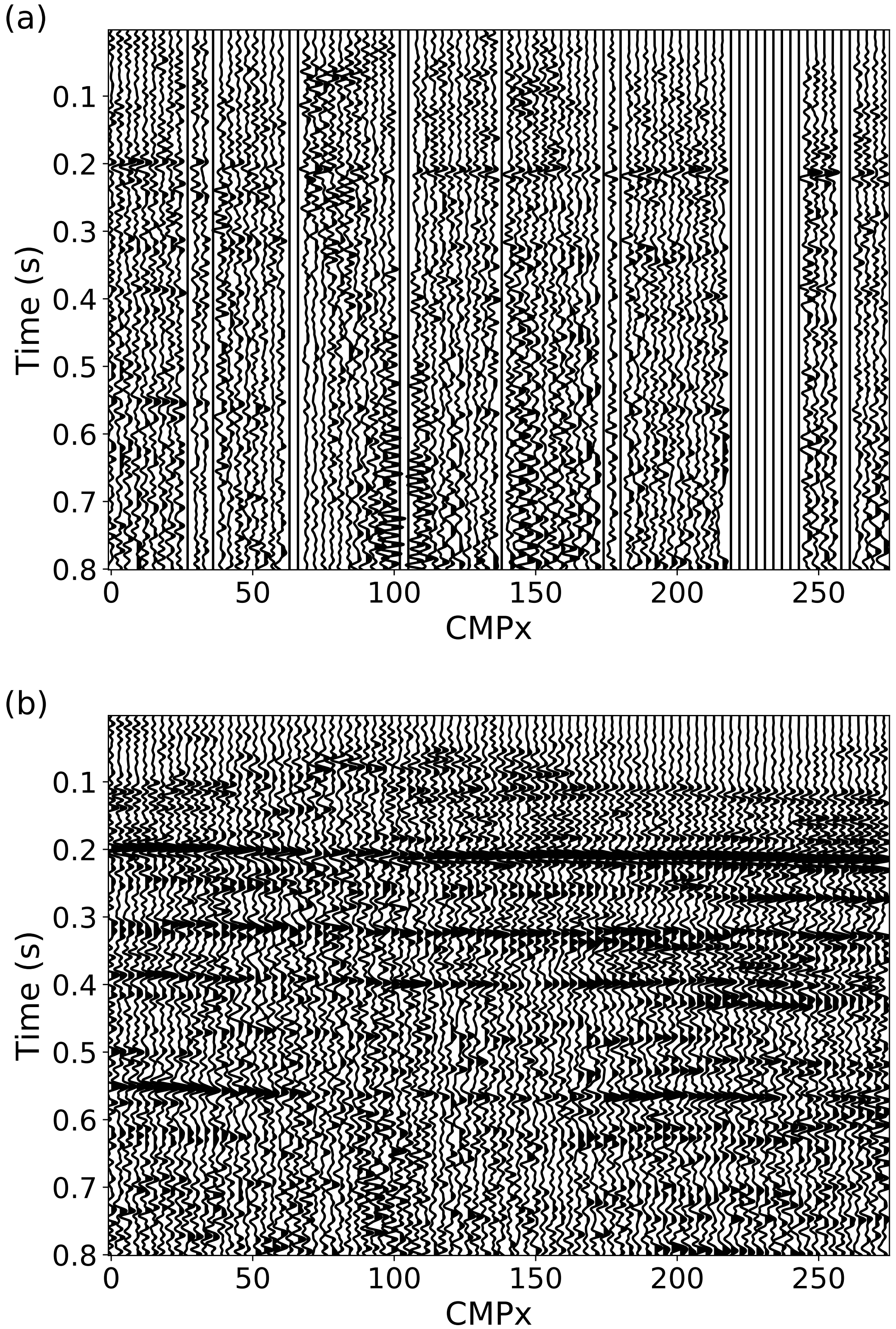


Figure 3.12: Reconstructed field seismic data for fixed CMPy. (a) Original data. (b) Reconstructed data using l_1/l_2 penalty function.

approximation to the data tensor. The solution is obtained through optimization of the objective function.

In order to achieve low-rank tensor completion, the method applies low-rank matrix factorization to every mode unfolding of the data tensor via randomized QR decomposition. The advantage of the proposed technique is twofold: first, we avoid the computational cost of the SVD, secondly and most important, the method does not require the definition of the rank of the ideal data. This is an important advantage from the application point of view.

Following, the presence of erratic or non-gaussian noise is acknowledged by modelling the misfit term with robust penalty functions such as l_1/l_2, Cauchy or GermanMcClure, instead of the popular Frobenius one. The results obtained on synthetic and land data show the performance of the algorithm, even in the presence of curved events and low SNR.

References

Abma, R., and N. Kabir, 2006, 3d interpolation of irregular data with a pocs algorithm: Geophysics, **71**, E91–E97.

Carroll, J. D., and J.-J. Chang, 1970, Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition: Psychometrika, **35**, 283–319.

Cheng, J., and M. D. Sacchi, 2015, A fast rank-reduction algorithm for 3d deblending via randomized qr decomposition, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 3830–3835.

Cichocki, A., D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, 2015, Tensor decompositions for signal processing applications: From two-way to multiway component analysis: IEEE Signal Processing Magazine, **32**, 145–163.

Ely, G., S. Aeron, N. Hao, and M. E. Kilmer, 2015, 5d seismic data completion and denoising using a novel class of tensor decompositions: Geophysics, **80**, V83–V95.

Gabay, D., and B. Mercier, 1976, A dual algorithm for the solution of nonlinear variational problems via finite element approximation: Computers & Mathematics with Applications, **2**, 17–40.

Gao, J., M. D. Sacchi, and X. Chen, 2013, A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions: Geophysics, **78**, V21–V30.

Gao, J., A. Stanton, and M. D. Sacchi, 2015, Parallel matrix factorization algorithm and its application to 5d seismic reconstruction and denoising: Geophysics, **80**, V173–V187.

Halko, N., P.-G. Martinsson, and J. A. Tropp, 2011, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions: SIAM review, **53**, 217–288.

Harshman, R. A., 1970, Foundations of the parafac procedure: Models and conditions for an explanatory multi-modal factor analysis: UCLA Working Papers in Phonetics, **16**, 1–84.

Kilmer, M. E., and C. D. Martin, 2011, Factorization strategies for third-order tensors: Linear Algebra and its Applications, **435**, 641–658.

Kolda, T. G., and B. W. Bader, 2009, Tensor decompositions and applications: SIAM review, **51**, 455–500.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, V113–V122.

Kreimer, N., A. Stanton, and M. D. Sacchi, 2013, Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction: Geophysics, **78**, V273–V284.

López, O., R. Kumar, Ö. Yılmaz, and F. J. Herrmann, 2016, Off-the-grid low-rank matrix recovery and seismic data reconstruction: IEEE Journal of Selected Topics in Signal Processing, **10**, 658–671.

Martin, C. D., R. Shafer, and B. LaRue, 2013, An order-p tensor factorization with applications in imaging: SIAM Journal on Scientific Computing, **35**, A474–A490.

Oropeza, V., and M. D. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Sidiropoulos, N. D., L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, 2017, Tensor decomposition for signal processing and machine learning: IEEE Transactions on Signal Processing, **65**, 3551–3582.

Stanton, A., N. Kreimer, D. Bonar, M. Naghizadeh, and M. Sacchi, 2012, A comparison of 5d reconstruction methods, *in* SEG Technical Program Expanded Abstracts 2012: Society of Exploration Geophysicists, 1–5.

Sternfels, R., G. Viguier, R. Gondoin, and D. Le Meur, 2015, Multidimensional simultaneous random plus erratic noise attenuation and interpolation for seismic data by joint low-rank and sparse inversion: Geophysics, **80**, WD129–WD141.

Trickett, S., L. Burroughs, and A. Milton, 2012, Robust rank-reduction filtering for erratic noise, *in* SEG Technical Program Expanded Abstracts 2012: Society of Exploration Geophysicists, 1–5.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3829–3833.

Tucker, L. R., 1964, The extension of factor analysis to three-dimensional matrices: Contributions to mathematical psychology, 110–127.

Vermeer, G. J., 2002, 3-D seismic survey design: Society of Exploration Geophysicists.

Wen, Z., W. Yin, and Y. Zhang, 2012, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm: Mathematical Programming Computation, 1–29.

Xu, Y., R. Hao, W. Yin, and Z. Su, 2015, Parallel matrix factorization for low-rank tensor completion: Inverse Problems and Imaging, **9**, 601–624.

F-X Quaternion singular spectrum analysis

B. Bahia[1]

Abstract

The quaternion signal model can naturally represent multi-component seismic data. In this paper, this model is employed to generalize the Singular Spectrum Analysis (SSA) filter to the case of vector measurements. The necessary tools (quaternion Fourier transform and quaternion singular value decomposition) for generalization of such technique to the quaternion domain are reviewed. Quaternion vector autoregressive models are used to justify the rank-reduction operation in the quaternion domain. Simultaneous denoising and reconstruction of synthetic multicomponent seismic records are illustrated and compared against the scalar version of the SSA filter.

4.1 Introduction

Multicomponent (MC) sensors have allowed the vector wavefield to be acquired by recording vibrations in three orthogonal directions, acquiring, therefore, a three-dimensional (3D) signal. The processing of such signals is said to be more intensive due to, for example, the volume of data to be processed. In this light, different approaches for processing multidimensional signals can be taken. For instance, processing can be carried out component-wise, where each component is processed per time. Another option is to embed the vectorial signal into a long-vector, allowing for its simultaneous processing. However, both approaches have drawbacks which have been gaining attention among

[1] Email: breno.bahia@ualberta.ca

researchers lately. Namely, component-wise processing of vector data disregards possible relations between its components whereas the long-vector notation accounts only for linear relations between them, which could result in a loss of information (Le Bihan and Ginolhac, 2001; Le Bihan and Mars, 2004). A more concise approach to handle vector data is to adopt a quaternionic model for multidimensional signals (Le Bihan and Mars, 2004). The reasoning is that the quaternion domain ($\mathbb{H}$) offers a natural way of representing three- and four-dimensional signals, such as three- and four-component (3C and 4C) seismic data, while accounting and preserving mutual relations between components (Le Bihan and Mars, 2004; Took and Mandic, 2011; Stanton and Sacchi, 2013). The usage of quaternions, therefore, has been increasing among the signal processing community, which retools the field with new techniques based on quaternion algebra. Examples of known techniques that have been extended to the quaternion domain are QR decomposition (Bunse-Gerstner et al., 1989), Fourier transforms (Ell, 1992), and singular value decomposition (SVD) (Le Bihan and Mars, 2004). Such extensions allow the formulation of further signal processing techniques, as subspace methods for signal decomposition, in the quaternion domain. For example, singular spectrum analysis (SSA) has been applied to the multichannel analysis of electroencephalograms based on a quaternion-valued approach (Enshaeifar et al., 2016). In the geophysical community, applications of quaternion signal processing have already been presented; some examples regarding the processing of MC seismic signals are time-lapse analysis and edge detection (Witten and Shragge, 2006), velocity analysis (Grandi et al., 2007), deconvolution (Menanno and Mazzotti, 2012), reconstruction (Stanton and Sacchi, 2013), and wavefield separation (Sajeva and Menanno, 2017).

The goal of this paper is to validate the quaternion-valued SSA (QSSA) (Enshaeifar et al., 2016) as a tool for simultaneous denoising and reconstruction of multicomponent seismic data in a holistic approach. SSA is usually regarded as a model-free (data-driven) method, and has, therefore, applications in diverse areas of knowledge such as climatology (Ghil et al., 2002), seismic data processing (Trickett, 2008; Oropeza and Sacchi, 2011), and medicine (Enshaeifar et al., 2016). The method is mainly described as a subspace decomposition algorithm where the original signal is represented as a sum of components such as trend, periodic components, and noise. Trend and periodic components are often approximated by time series of small rank (Golyandina and Zhigljavsky, 2013) whereas the noise subspace is usually ignored. SSA is often described in two complementary stages: decomposition and reconstruction. The decomposition stage is based on the SVD of what is known as trajectory matrix. SSA has already been successfully applied to denoising and reconstruction of single component seismic data by Trickett (2008)[2] and Oropeza and Sacchi (2011), and recently extended to MC seismic data (Sacchi et al., 2017; Janzen et al., 2017). For the

[2] Cadzow filtering and SSA are equivalent algorithms arising from different fields.

MC data case, however, the authors employed long-vector notation to represent their data samples in frequency-space ($f - x$) domain. Here, the potential of the quaternion field for representation of 3D (or 4D) signals is exploited by adopting the quaternion signal model to represent 3C (or 4C) data within the SSA context. Two versions of SSA were defined by Enshaeifar et al. (2016) in the quaternion domain; one is based on the straightforward extension of its real counterpart by proper replacement of real-valued techniques with their quaternionic equivalents, and the other exploits augmented second-order statistics of quaternion random signals (Took and Mandic, 2011).

The underlying idea for the application of SSA in quaternion-valued MC data is the same as in Oropeza and Sacchi (2011) and Sacchi et al. (2017), where denoising and reconstruction are applied to the signal at monochromatic frequency slices, thus requiring the application of the Fourier transform (FT) to the data. As already mentioned, FT has been extended to $\mathbb{H}$ by Ell (1992), and has found applications in the processing of color images (Ell and Sangwine, 2007) and MC seismic data reconstruction (Stanton and Sacchi, 2013). Likewise, SVD has also been extended to $\mathbb{H}$, paving the way for applications as wavefield separation (Le Bihan and Mars, 2004; Sajeva and Menanno, 2017). Together with vector autoregressive models for quaternion variables (Ginzberg and Walden, 2013), these techniques allow for the formulation of simultaneous MC seismic data denoising and reconstruction based on rank-reduction methods, as it is going to be discussed. The paper is then organized as follows: section 2 is devoted to a brief description of quaternion algebra, quaternion model of vector signals, quaternion SVD (QSVD) and quaternion Fourier transform (QFT). QSSA is discussed, with mention to its augmented version, and its applications to synthetic data are illustrated in section 3, followed by conclusions in section 4.

4.2 Quaternions

Quaternions are an extension of complex numbers, being also regarded as one case of hypercomplex numbers. Its inherent 4D structure provides an intuitive framework for the representation of three- and four-dimensional signals, yielding a unified processing approach for vector data. In the following, some basic definitions of quaternion algebra are reviewed. This will provide enough background to allow the definition of quaternion signals and possible techniques for its processing. Thorough descriptions of quaternions can be found elsewhere in the literature, as Ward (2012) and, focusing on signal and image processing, Hitzer and Sangwine (2013) and Ell et al. (2014). The description given here mainly follows Ell et al. (2014).

Quaternion algebra

Definitions

A quaternion variable, $q \in \mathbb{H}$, can be defined as a four-dimensional hypercomplex number with cartesian form given by

$$q = a + bi + cj + dk, \tag{4.1}$$

where the tuple $\{a, b, c, d\} \in \mathbb{R}$ are its components. A quaternion can be decomposed as a scalar (or real) part plus a vector (or imaginary) part, which consists of three imaginary components, as follows

$$S(q) = a, \tag{4.2}$$

$$V(q) = bi + cj + dk = q - S(q), \tag{4.3}$$

and finally

$$q = S(q) + V(q). \tag{4.4}$$

A quaternion is said to be pure if $S(q) = 0$, and the set of pure quaternions can be denoted as $V(\mathbb{H})$. Quaternions with null vector part are simply identified as elements of $\mathbb{R}$. The imaginary units (i, j, k) follow the rules[3]

$$i^2 = j^2 = k^2 = -1 \tag{4.5}$$

and

$$ijk = -1, \tag{4.6}$$

and also describe the three orthogonal axes of the vector part in q. Furthermore, together with the unitary operator, the imaginary units form a 4D basis in $\mathbb{H}$ $(\{1, i, j, k\})^4$. Finally, it is possible to notice that the sets of real and complex numbers are special cases of quaternions, i.e., $\mathbb{R} \in \mathbb{C} \in \mathbb{H}$.

[3] Other relations, such as $ij = -ji = k$, can be drawn from 4.6 by left or right multiplications by i or k, respectively.

[4] This is one specific 4D basis in $\mathbb{H}$. Others are also possible.

Following the rules given in 4.5 and 4.6, one can get to the conclusion that multiplication in $\mathbb{H}$ is, in general, not commutative. Using the words of Ell and Sangwine (2007), that is the main difference between quaternion and usual algebra. In fact, the operations of addition and multiplication of quaternion variables are defined in analogy to its complex counterpart, always keeping in mind that the ordering of operators is not arbitrary due to the field's property of non-commutativity. This property has important implications in the development of quaternion-based techniques, as the example of the QFT. On the other hand, provided that the non-commutativity of quaternion algebra is kept in mind, Ell and Sangwine (2007) describe the implementation of quaternion-based operations and techniques as being straightforward. Actually, when manipulating quaternion variables, one hopes that classical definitions for real or complex numbers to be trivial extensions to the quaternion field. However, there are specific properties for the quaternion domain which do not arise in its complex counterpart (see, e.g., Took and Mandic (2011)), and some extensions might not be that straightforward.

Properties

Some properties of quaternions that are interesting for the scope of this paper are the conjugate of a quaternion, which is given by

$$q^* = a - bi - cj - dk, \tag{4.7}$$

and its modulus

$$|q| = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{4.8}$$

If $|q| = 1$, q is a unitary quaternion.

Crucial for the development of the augmented QSSA is the concept of quaternion involutions (self-inverse mappings) about the imaginary axes

$$q^i = -iqi = a + bi - cj - dk, \tag{4.9}$$

$$q^j = -jqj = a - bi + cj - dk, \tag{4.10}$$

and

$$q^k = -kqk = a - bi - cj + dk. \tag{4.11}$$

One can relate a quaternion and its involutions by

$$q^* = \frac{1}{2}(q^i + q^j + q^k - q), \tag{4.12}$$

and express the four components of q as

$$a = \frac{1}{2}(q + q^*) \qquad b = \frac{1}{2i}(q - q^{i*}) \tag{4.13}$$

$$c = \frac{1}{2j}(q - q^{j*}) \qquad d = \frac{1}{2k}(q - q^{k*}). \tag{4.14}$$

Representations

There are different representations of quaternions which are particularly useful in the development and study of hypercomplex techniques. Examples are polar and Cayley-Dickson (CD) forms of a quaternion. The polar form of a quaternion is equivalent of the Euler formula, and is given by

$$q = |q|e^{\mu\phi} \tag{4.15}$$

where its modulus ($|q|$) is given by 4.8, and the phase ($\phi \in \mathbb{R}^+$) and eigenaxis ($\mu \in V(\mathbb{H})$) are given by

$$\phi = \tan^{-1}\frac{|V(q)|}{S(q)} \tag{4.16}$$

and

$$\mu = \frac{V(q)}{|V(q)|}, \tag{4.17}$$

respectively. From 4.17, note that $|\mu| = 1$, being called a pure unitary quaternion.

A quaternion can also be represented as a pair of complex numbers. This is known as Cayley-Dickson form, and it splits the quaternion in two different (orthogonal) planes. In this representation, q is written as

$$q = z_1 + z_2 j, \tag{4.18}$$

where $z_1 = a + bi$ and $z_2 = c + di$. Notice that $\{z_1, z_2\} \in \mathbb{C}_i$, the set of complex numbers defined with $i^2 = -1$. Therefore, the CD form allows the interpretation of a quaternion in terms of complex numbers. This representation can be generalized to what is called symplectic decomposition of quaternions, where q is written in another basis $\{1, \mu, \nu, \mu\nu\} \in \mathbb{H}$ as q'

$$q' = a' + b'\mu + c'\nu + d'\mu\nu, \tag{4.19}$$

with decomposition given by

$$q' = (a' + b'\mu) + (c' + d'\mu)\nu. \tag{4.20}$$

The complex units $\{\mu, \nu, \mu\nu\}$ follow the same rules as in 4.5 and 4.6

$$\mu^2 = \nu^2 = (\mu\nu)^2 = -1, \tag{4.21}$$

and $S(\mu\nu) = 0$, which imples $\mu \perp \nu$, $\mu\nu \perp \mu$ and $\mu\nu \perp \nu$ (orthonormal basis in $V(\mathbb{H})$). The new set of components $\{a', b', c', d'\}$ is obtained through the change of basis

$$a' = a \tag{4.22}$$

$$\begin{pmatrix} b' \\ c' \\ d' \end{pmatrix} = \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 \\ \nu_1 & \nu_2 & \nu_3 \\ \mu\nu_1 & \mu\nu_2 & \mu\nu_3 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \end{pmatrix}. \tag{4.23}$$

The quaternion given by 4.20 is decomposed into its simplex and perplex parts

$$q' = q_s + q_p\nu, \tag{4.24}$$

with $q_s = (a' + b'\mu)$ and $q_p = (c' + d'\mu)$. Again, notice that $\{q_s, q_p\} \in \mathbb{C}_\mu$. It is important to pay attention to the fact that the simplex and perplex parts of a quaternion in its symplectic (and CD) form are isomorphic to complex numbers since it allows quaternion signals to be used as complex signals. This, in turn, facilitates the development of quaternion-based techniques (Ell and Sangwine, 2007).

Vectors and matrices of quaternions

The definitions above facilitates the description of vectors and matrices of quaternions. Vectors and matrices of quaternions will be used in the representation of quaternion signals recorded by vector sensors. Again, while several classical definitions can be easily extended to the quaternion domain as, for instance, scalar product, norm and the distance between two vectors, some operations for real and complex arrays do not hold in the quaternion domain. The reader can refer to Le Bihan and Mars (2004) to a signal-processing-oriented description of such operations, and to Zhang (1997) for a more complete study of quaternion matrices.

Intuitively, a vector of quaternions $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_N]^T$ ($\mathbf{x} \in \mathbb{H}^N$) with N dimensions is such that its elements are quaternions ($x_i \in \mathbb{H}, i = [1, N]$). The scalar product between two quaternion vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\triangleleft}\mathbf{y}, \tag{4.25}$$

where the superscript $^{\triangleleft}$ denotes the conjugate transpose of a quaternion vector. The norm of a quaternion vector is then given by

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^{\triangleleft}\mathbf{x}}. \tag{4.26}$$

Likewise, a matrix $\mathbf{X} \in \mathbb{H}^{N\times M}$ also has quaternions as its elements $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_M]$, with $\mathbf{x}_i \in \mathbb{H}^N, i = [1, M]$.

Representations of quaternions can also be applied to vectors and matrices of quaternions. The complex representation (CD form) of vectors and matrices of quaternions is rather useful for the concepts of rank and SVD in this field.

The CD representation of a quaternion vector $\mathbf{x} \in \mathbb{H}^N$ is given by $\mathbf{x} = \mathbf{x}_s + \mathbf{x}_p j$, where $\{\mathbf{x}_s, \mathbf{x}_p\} \in \mathbb{C}_i^N$. For a quaternion matrix $\mathbf{X} \in \mathbb{H}^{N\times M}$, the CD form is no different: $\mathbf{X} = \mathbf{X}_s + \mathbf{X}_p j$, where $\{\mathbf{X}_s, \mathbf{X}_p\} \in \mathbb{C}_i^{N\times M}$. The complex representation is the most common way of studying quaternion matrices. This is done with the definition of the complex adjoint matrix, $\chi_X \in \mathbb{C}^{2N\times 2M}$, of a quaternion matrix $\mathbf{X}$, which is given by

$$\chi_X = \begin{pmatrix} \mathbf{X}_s & \mathbf{X}_p \\ -\mathbf{X}_p^* & \mathbf{X}_s^* \end{pmatrix}. \tag{4.27}$$

It is through this notation of quaternion matrices that, for instance, complex decomposition algorithms can be used to compute decomposition of quaternion matrices (Le Bihan and Mars, 2004).

The definition of rank of a quaternion matrix is essential in the development of SSA in the quaternion domain. Just as for real and complex matrices, the rank of a quaternion matrix $\mathbf{X}$ can be defined as the maximum number of columns that are right linearly independent. It can also be understood as the minimum number of rank-1 quaternion matrices that result in $\mathbf{X}$ by linear combination. Other useful properties for the discussion of QSVD in the following sections are: the rank of a quaternion matrix $\mathbf{X}$ is k if and only if the rank of its complex adjoint χ_X is $2k$. Finally, the rank of a quaternion matrix $\mathbf{X}$ is k if $\mathbf{X}$ has k non-zero singular values (Zhang, 1997).

Quaternion signals

As stated before, the objective of this paper is to avoid the usage of component-wise or long-vector notation when processing vector signals. The approach adopted here follows Le Bihan and Mars (2004) and Ell and Sangwine (2007), where the vector signals are defined by means of quaternions, which allows for its simultaneous and concise processing. For the case of 2D-3C seismic data, where an array of vector sensors records three orthogonal components, here denoted by $(\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z) \in \mathbb{R}^{nt\times nx}$, at nt discrete time samples and nx discrete space positions, its possible to encode the three corresponding signals into the components of a pure quaternion matrix $\mathbf{Q} \in V(\mathbb{H}^{nt\times nx})$ such as

$$\mathbf{Q}[n,x] = \mathbf{U}_x[n,x]i + \mathbf{U}_y[n,x]j + \mathbf{U}_z[n,x]k, \tag{4.28}$$

with $n = [1, nt]$ and $x = [1, nx]$. As it is possible to notice, $\mathbf{Q}$ represents the multicomponent seismic data, and $\mathbf{U}_x$, $\mathbf{U}_y$, and $\mathbf{U}_z$ are the two horizontal and vertical components, respectively. Of course, extension for 4C and data with higher dimensions is possible. As it is defined, $\mathbf{Q}$ requires quaternion-based techniques for its processing, as for instance the QFT and QSVD which are discussed next.

It is important to stress, though, that quaternions can be used to accommodate two pairs of complex signals, as it follows from its symplectic decomposition, and process them simultaneously (Stanton and Sacchi, 2013). In this case, two components in the frequency domain, say x and y, could be represented by means of the Cayley-Dickson form of quaternions as

$$\mathbf{Q}[\omega,x] = \tilde{\mathbf{U}}_x[\omega,x] + \tilde{\mathbf{U}}_y[\omega,x]j, \tag{4.29}$$

where $\{\tilde{\mathbf{U}}_x, \tilde{\mathbf{U}}_y\} \in \mathbb{C}_i^{nt\times nx}$ are obtained through the classical Fourier transform of the real components $\mathbf{U}_x$ and $\mathbf{U}_y$.

4.3 Quaternion tools for signal processing

The definition of quaternion signals as a “new” way to handle multidimensional signals calls for generalization of classical tools in signal processing, such as Fourier and Laplace transforms, to their hypercomplex versions. In fact, although a recent growing number of applications of hypercomplex numbers in signal processing are being documented, usage of quaternions in this field dates back to the 80s, with applications to Fourier transforms in nuclear magnetic resonance imaging by Ernst et al. (1987) and Delsuc (1988). In the following sections, the quaternionic extension of two signal processing techniques vastly used in its real and complex versions are discussed: the Singular Value Decomposition and the Fourier Transform. Furthermore, the open-source Quaternion toolbox for MATLAB (Sangwine, 2005) makes computation with quaternion matrices convenient by offering the implementation of such techniques and others more.

Quaternion Singular Value Decomposition

Quaternion singular value decomposition is a fundamental tool when treating vector signals. Zhang (1997) provides the theoretical basis for the existence of the SVD for quaternion matrices but its computation is given in two ways by Le Bihan and Mars (2004) and Sangwine and Le Bihan (2006). QSVD can be computed through the complex representation of quaternion-valued matrices, but its description is analogous to the real or complex case. The possibility of computing the QSVD allows for the definition of quaternionic signal processing techniques as wave separation based on subspace methods (Le Bihan and Mars, 2004) and QSSA (Enshaeifar et al., 2016).

A given quaternion matrix $\mathbf{X} \in \mathbb{H}^{N\times M}$ of rank r has a SVD written as

$$\mathbf{X} = \mathbf{U}\begin{pmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{V}^{\triangleleft} \tag{4.30}$$

where $\mathbf{\Sigma}_r \in \mathbb{R}^{r\times r}$ contains the r singular values of $\mathbf{X}$, arranged in decreasing magnitude order, on its main diagonal. $\mathbf{U} \in \mathbb{H}^{N\times N}$ and $\mathbf{V} \in \mathbb{H}^{M\times M}$ are unitary quaternion matrices (so $S(\mathbf{U}\mathbf{U}^{\triangleleft}) = S(\mathbf{V}\mathbf{V}^{\triangleleft}) = \mathbf{I}$, and $V(\mathbf{U}\mathbf{U}^{\triangleleft}) = V(\mathbf{V}\mathbf{V}^{\triangleleft}) = \mathbf{0}$) which contain the left and right singular vectors of $\mathbf{X}$, respectively. Proof of existence of QSVD is given in Zhang (1997) or Loring (2012), and it is based on the polar decomposition of quaternion matrices. QSVD is rank revealing, i.e., the number of non-null singular values equals the rank of the matrix. Hence, likewise real or complex cases, QSVD can be written as a sum of r rank-1 quaternion matrices as

$$\mathbf{X} = \sum_{n=1}^{r} \mathscr{X}_n = \sum_{n=1}^{r} \mathbf{u}_n \mathbf{v}_n^{\triangleleft} \sigma_n, \tag{4.31}$$

where $\mathbf{u}_n$ (nth column of $\mathbf{U}$) and $\mathbf{v}_n$ (nth column of $\mathbf{V}$) are the left and right singular vectors, and σ_r represents the (real) singular values of $\mathbf{X}$.

One way to compute the QSVD uses the isomorphism between quaternion matrices ($\mathbb{H}^{N\times M}$) and their Cayley-Dickson representation ($\mathbb{C}^{2N\times 2M}$) (Le Bihan and Mars, 2004), which allows the SVD of a quaternion matrix $\mathbf{X}$ to be obtained from the SVD of its complex adjoint matrix χ_X (equation 4.27) using classical algorithms. The SVD of χ_X is given by

$$\chi_X = \mathbf{U}^{\chi_X} \begin{pmatrix} \mathbf{\Sigma}_{2r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} (\mathbf{V}^{\chi_X})^{\triangleleft} = \sum_{n'=1}^{2r} \mathbf{u}_{n'}^{\chi_X} \mathbf{v}_{n'}^{\chi_X \triangleleft} \sigma_{n'}, \tag{4.32}$$

with left and right singular vectors given by $\mathbf{u}_n^{\chi_X}$ and $\mathbf{v}_n^{\chi_X}$, respectively. The diagonal matrix $\mathbf{\Sigma_{2r}}$ has a special structure where its singular values ($\sigma_{n'}$) appear by pairs along its main diagonal (Zhang, 1997), which are also the singular values of $\mathbf{X}$[5]. Once computed the complex SVD of χ_X, obtaining the singular values and vectors of $\mathbf{X}$ is a matter of associating the correct indexes: the nth singular elements of $\mathbf{X}$ ($\sigma_n, \mathbf{u}_n, \mathbf{v}_n$) are associated with the $n' = (2n-1)$th singular elements of χ_X. This happens because the matrix is redundant in the way it is built to preserve the quaternion properties (Sangwine and Le Bihan, 2006).

It is also possible to obtain a rank-p ($p < r$) approximation of a given quaternion matrix, in a least-squares sense, through its QSVD. As in real and complex cases, the best rank-p estimate to a quaternion matrix can be obtained by truncating QSVD at its first p singular elements in equation 4.31 (Le Bihan and Mars, 2004). It is, therefore, possible to use low-rank techniques in quaternion-based signal processing provided that these approximations are justified.

Quaternion Fourier Transform

The importance of Fourier analysis in signal processing is incontestable. The extension of Fourier transforms to the field of quaternions is, therefore, a natural step in the processing and analysis of

[5] As it can be noticed, χ_X has twice more singular values than $\mathbf{X}$ due to its construction and size (Le Bihan and Mars, 2004).

signals defined employing quaternions. A formal definition of hypercomplex spectral transformations can be found in Ell (1992) but earlier applications of quaternions together with FTs had already been documented, as aforementioned. Other applications of the QFT already found in the literature include the definition of multidimensional analytic signals (Bulow and Sommer, 2001), processing of color images (Ell and Sangwine, 2007), and vector reconstruction of MC seismic data (Stanton and Sacchi, 2013). Good references on the topic are Hitzer and Sangwine (2013) and Ell et al. (2014).

Quaternion Fourier transforms have allowed the holistic treatment of vector-signals defined with quaternion variables. The fact that quaternion multiplication is non-commutative has important implications in the QFT definition. Specifically, due to non-commutativity, the positioning of the exponential kernel in the transform definition is meaningful, and a variety of QFTs is possible. Moreover, the fact that a generalized complex operator $\mu^2 = -1$ can be defined increases the number of possible QFT definitions even further[6]. The 1D right-sided forward[7] FT of a quaternion-valued function $f : \mathbb{R} \rightarrow \mathbb{H}$ is defined as (Ell et al., 2014)

$$\mathscr{F}_{\mu}^{R}\{f\}(\omega) = F^{R}(\omega) = \kappa_{-} \int_{-\infty}^{+\infty} f(t) e^{-\mu\omega t} dt, \tag{4.33}$$

with inverse given by

$$\mathscr{F}_{\mu}^{-R}\{F^{R}\}(t) = f(t) = \kappa_{+} \int_{-\infty}^{+\infty} F(\omega) e^{\mu\omega t} d\omega, \tag{4.34}$$

where μ is an arbitrary pure unit quaternion ($\mu^2 = -1$), and $\kappa_{+}\kappa_{-} = (2\pi)^{-1}$. This definition is said to be the *right-sided* QFT because the exponential kernel is placed to the right of the function to be transformed. In a similar fashion, therefore, one can define the *left-sided* QFT by plancing the exponential kernel by the left side of $f(t)$ or $F(\omega)$. For vector signals of higher dimensons, as color images or 5D seismic data, the number of QFT definitions is even greater because non-commutativity also implies that $e^{ia+jb} \neq e^{ia}e^{jb}$, and each of these kernels can be placed on either side of the function to be transformed. This gives rise to sandwiched, as well as factored and unfactored multi-axis definitions of the QFT as shown in table 4.1 for the forward 2D case (Hitzer and Sangwine, 2013).

It is important to notice that, since the complex units (i, j, k) are special cases of quaternions, QTFs actually offer a generalization of the Fourier transform definition where the complex FT is a special case in which $\mu = i$.

[6] To infinity (since there are infinite square roots of -1 in $\mathrm{V}(\mathbb{H})$).

[7] The authors also define a reverse transform, where the sign of the exponent is positive. This transform is not the same as the inverse transform.

Table 4.1: 2D QFT definitions

	Left	Right	Sandwich
Single-axis	$e^{-\mu(\omega x+\eta y)}f(x,y)$	$f(x,y)e^{-\mu(\omega x+\eta y)}$	$e^{-\mu\omega x}f(x,y)e^{-\mu\eta y}$
Dual-axis	$e^{-(\mu\omega x+\nu\eta y)}f(x,y)$	$f(x,y)e^{-(\mu\omega x+\nu\eta y)}$	NA
Factored	$e^{-\mu\omega x}e^{-\nu\eta y}f(x,y)$	$f(x,y)e^{-\mu\omega x}e^{-\nu\eta y}$	$e^{-\mu\omega x}f(x,y)e^{-\nu\eta y}$

The importance of the isomorphism between complex numbers and complex subfields (defined in the symplectic decomposition of quaternions) becomes more apparent when the question of what are the criteria for quaternion functions to be Fourier-transformable arises. Existence and invertibility are inherited, due to this isomorphism, from the complex case. Also, numerical implementation is greatly facilitated with the use of already existing complex Fourier transforms, such as complex FFTs (Ell et al., 2014). Regarding computational aspects of QFTs, it is first necessary to define the discrete QFT (DQFT), as it is done for the 1D left-sided case[8] (Ell et al., 2014)

$$F\left[\omega\right] = \frac{1}{\sqrt{N}} \sum_{n=0}^{nt-1} e^{-\mu 2\pi \frac{n\omega}{N}} f[n], \tag{4.35}$$

with inverse

$$f\left[n\right] = \frac{1}{\sqrt{N}} \sum_{\omega=0}^{nt-1} e^{\mu 2\pi \frac{n\omega}{N}} F[\omega]. \tag{4.36}$$

Instead of writing a direct code for computation of DQFT, or even a FFT code for it, Ell and Sangwine (2007) propose to take advantage of already existing FFT codes by using symplectic decomposition of quaternions and isomorphisms with the complex case. First, notice that the DQFT of a quaternion function f along a given eigenaxis μ requires the definition of a basis $\{1, \mu, \nu, \mu\nu\}$. According to Ell and Sangwine (2007), the choice of μ and ν is arbitrary but these cannot be parallel to each other. If ν is not perpendicular to μ, an uptaded value for ν can be found by taking the cross-product between μ and $\mu\nu$. Having defined the 4D basis in $\mathbb{H}$, the 1D DQFT is as follows (Ell and Sangwine, 2007):

- Apply a change of basis from f to f' following 4.23;

[8] The superscripts are being dropped to avoid clutter. If necessary, distinction shall be evident from the context.

- Decompose the function into its simplex and perplex parts following 4.24;
- Perform two 1D complex FFTs of f_s and f_p to obtain $F_s[\omega]$ and $F_p[\omega]$;
- Construct the full quaternion in frequency domain by $F'[\omega] = F_s[\omega] + F_p[\omega]\nu$;

These steps can be generalized to a N-D QFT (Stanton and Sacchi, 2013).

The choice of the eigenaxis μ for the QFT deserves attention. Although earlier described as arbitrary, the choice of the eigenaxis has its consequences, which seems to be more understood in the case of color images (see, e.g., Ell and Sangwine (2007)). In a general sense, the choice of a different basis in $\mathbb{H}$ defines a preferential association of components axes, which could be described as a certain mixing between them (Stanton and Sacchi, 2013). The fact that there is no apparent preferential association of the signals in 3-space with the eigenaxis (or the basis) allows the usage of any pure quaternion μ in the QFT definition (Hitzer and Sangwine, 2013). In the case of RGB images, for instance, defining the eigenaxis $\mu = \frac{1}{\sqrt{3}}(i, j, k)$, corresponds to define it as the gray-line[9] in color-space. This choice of μ is said to be obvious in color-image vector processing as the simplex and perplex parts of the quaternion in the basis $\{1, \mu, \nu, \mu\nu\}$ results in the luminance and chrominance information, respectively, of the image. So far, no analogy could be found in the case of quaternion representation of MC seismic data (Stanton and Sacchi, 2013). Also, note that this degree-of-freedom when defining the QFT eigenaxis permits practitioners to define it according to their convenience. A trivial but perhaps advantageous choice could be $\{1, \mu, \nu, \mu\nu\} = \{1, i, j, k\}$, where the coefficients are kept the same after the change of basis (Stanton and Sacchi, 2013).

By knowing how to apply the QFT, it is now possible to describe the Fourier analysis of vector signals defined with quaternion variables. Using the definition of quaternionic signal given by 4.28, application of QFT yields another quaternion signal, which is usually a full quaternion (all four components are non-zero), represented by the QFTed simplex and perplex parts of the quaternion as

$$\tilde{\mathbf{Q}}[\omega, x] = \tilde{\mathbf{Q}}_s[\omega, x] + \tilde{\mathbf{Q}}_p[\omega, x]j, \tag{4.37}$$

with $\tilde{\mathbf{Q}} \in \mathbb{H}^{nt \times nx}$ obtained through the QFT with respect to time. The frequency domain coefficients in the hypercomplex spectrum have the same layout as the classical spectrum and are divided into positive and negative frequencies, also including the DC and Nyquist terms, as usually defined (Ell et al., 2014). Therefore, even though there are no conjugate symmetries for the QFT of a quaternion signal as in the real case, a symmetry of the coefficients indexes exists, i.e., the same frequency

[9] Diagonal of the RGB color cube connecting the colors black $(0, 0, 0)$ and white $(1, 1, 1)$.

component is represented by the indexes N and $nt - N$[10]. It is possible to illustrate the above by encoding the 2D 3C seismic data shown in figure 4.1 into the components of the quaternion signal. The seismic data was generated using a second-order elastic finite-differences simulation (Boyd, 2006) considering a medium consisting of three flat homogeneous layers where the middle one is an HTI medium. The physical properties of the medium follow Stanton and Sacchi (2013).

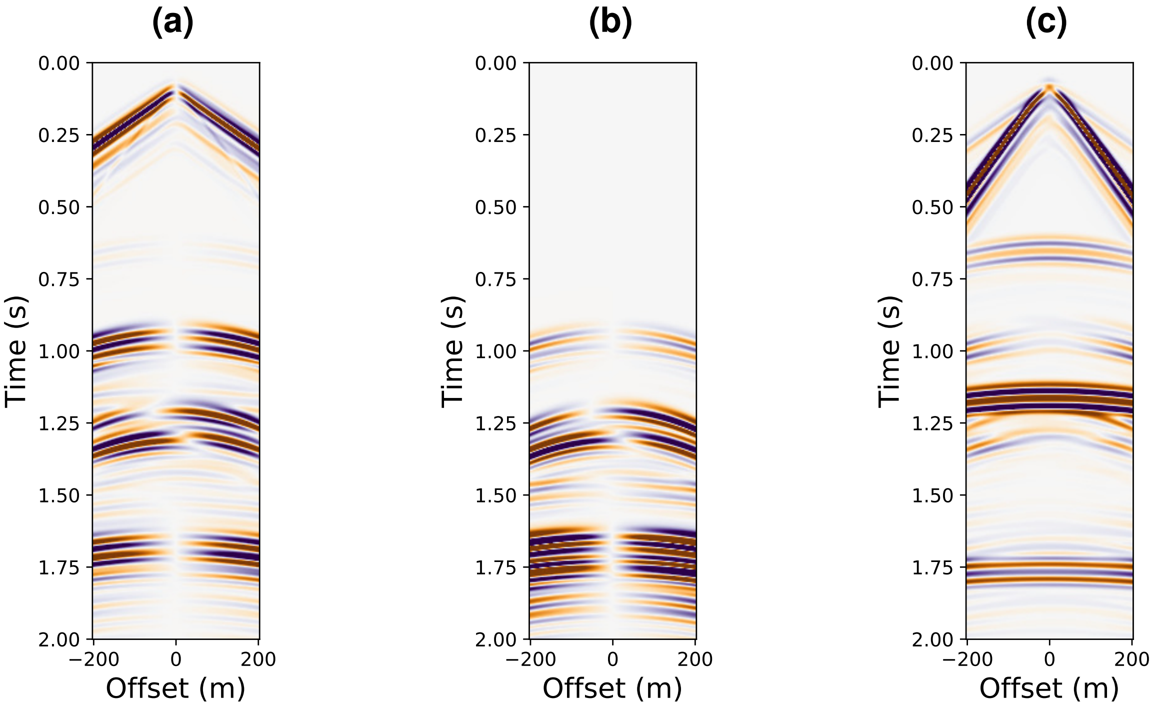


Figure 4.1: Elastic finite-difference synthetic data.

Analogously to its complex counterpart, a signal which has been QFTed is written in terms of cosinusoidal basis functions (the hypercomplex exponentials) and hypercomplex amplitudes. Its possible, therefore, to write the hypercomplex representation of the signal in terms of its amplitude and phase, analogously to what is usually done in classical processing, and the additional concept of orientation (the eigenaxis), as it follows from equation 4.15 (Ell and Sangwine, 2007). The amplitude and FK spectra of the above seismic data are shown in figure 4.2. Both of these spectra are computed by using the polar representation of quaternions, after applying the left-sided QFT with eigenaxis $\mu = \frac{1}{\sqrt{3}}(1, 1, 1)$ to the data. Although the frequency content being shown in figure

[10] One should pay attention for numerical implementation using one-based indexing. DC and Nyquist coefficients should be considered.

4.2(a) seems to be the same on both sides of the spectrum, they actually differ in value since there are no conjugate symmetries for quaternion signals (just as in the complex case) in the Fourier domain. As an implication, operations carried out in monochromatic frequency slices, as classically done in several filtering techniques (e.g., SSA, prediction, and projection filters), have to be applied to both sides of the spectrum separately. On the other hand, the hypercomplex spectrum offers the advantage of representing the amplitude of all of its elements in a single term given by its modulus (equation 4.8) (Stanton and Sacchi, 2013).

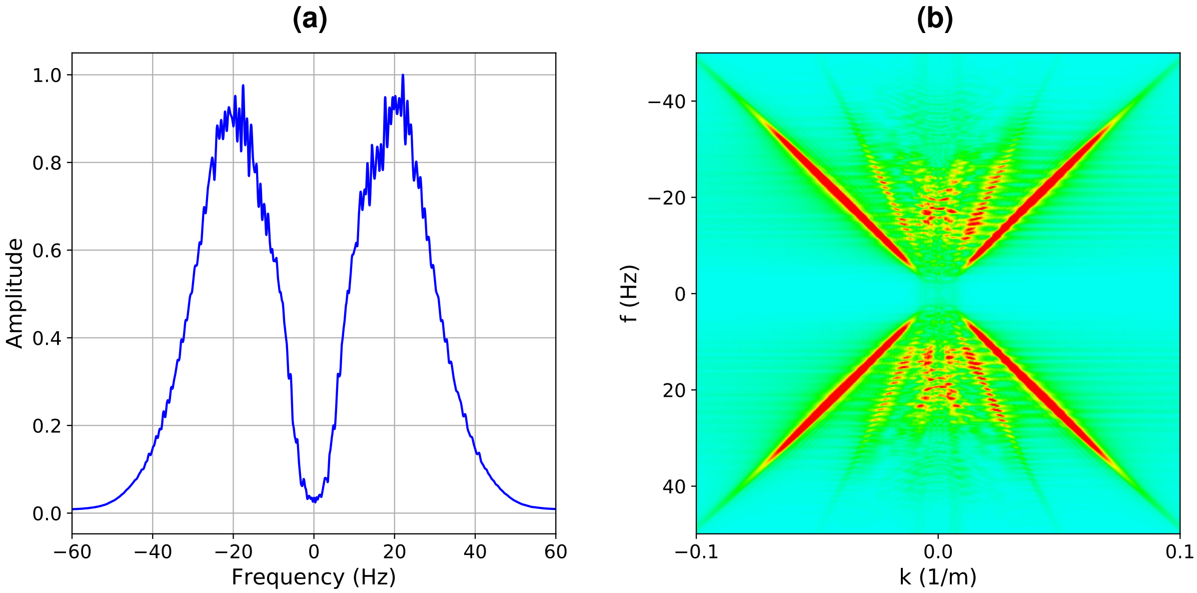


Figure 4.2: (a) Amplitude spectrum and (b) FK spectrum of the quaternion-based seismic signal.

4.4 (Quaternion) Singular Spectrum Analysis

SSA is a data-driven subspace decomposition algorithm often used for analysis of 1D time series, where the original signal is represented as the sum of components usually classified as belonging to signal or noise subspaces. SSA can be described, regardless of the domain it is being applied ($\mathbb{R}$, $\mathbb{C}$ or $\mathbb{H}$), in two stages: decomposition and reconstruction.

Considering the 2D 3C dataset in figure 4.1, represented as a quaternion signal, in frequency domain

$\tilde{\mathbf{Q}} \in \mathbb{H}^{nt \times nx}$, it is possible to represent a monochromatic frequency $\mathbf{q} \in \mathbb{H}^{nx}$ as

$$\mathbf{q} = \begin{pmatrix} q[1] & q[2] & \dots & q[nx] \end{pmatrix}^T . \tag{4.38}$$

In the decomposition stage of SSA, it is first necessary to construct a Hankel matrix $\mathbf{T} \in \mathbb{H}^{L \times nx-L+1}$, which columns are one-sample lagged versions of a length-L window of the signal $\mathbf{q}$. In the SSA literature, this matrix is named the trajectory matrix, and it is given by

$$\mathbf{T} = \begin{pmatrix} q[1] & q[2] & \dots & q[K] \\ q[2] & q[3] & \dots & q[K+1] \\ \vdots & \vdots & \ddots & \vdots \\ q[L] & q[L+1] & \dots & q[nx] \end{pmatrix}, \tag{4.39}$$

where $K = nx - L + 1$. Following Oropeza and Sacchi (2011), the mapping of the data to the trajectory matrix is going to be denoted by the operator $\mathscr{P}_{\mathscr{H}}$ such that

$$\mathbf{T} = \mathscr{P}_{\mathscr{H}} \mathbf{q}. \tag{4.40}$$

According to equation 4.31, $\mathbf{T}$ can be represented as a sum of rank-1 matrices using its QSVD

$$\mathbf{T} = \sum_{n=1}^{r} \mathscr{T}_n = \sum_{n=1}^{r} \mathbf{u}_n \mathbf{v}_n^{\triangleleft} \sigma_n, \tag{4.41}$$

where, again, r is its rank, $\mathbf{u}_n$ and $\mathbf{v}_n$ are the n-th right and left singular vectors, respectively, and σ_n its n-th singular value. The reconstruction stage starts with the truncation of the QSVD of $\mathbf{T}$ to its largest p singular elements. As already discussed, this approximates $\mathbf{T}$ to $\mathbf{T}_p$ in a least-squares sense. The rank-reduction (truncation) operation can be represented by the operator $\mathscr{P}_{\mathscr{R}}$ so one has

$$\mathbf{T}_p = \sum_{n=1}^{p} \mathscr{T}_n = \mathscr{P}_{\mathscr{R}} \mathbf{T}. \tag{4.42}$$

Finally, the reconstructed time series is achieved from $\mathbf{T}_p$ by averaging the elements along its antidiagonals, which is represented by the operator $\mathscr{P}_{\mathscr{A}}$

$$\hat{\mathbf{q}} = \mathscr{P}_{\mathscr{A}} \mathbf{T}_p. \tag{4.43}$$

Combining equations 4.40, 4.42 and 4.43, it is possible to write

$$\hat{\mathbf{q}} = \mathscr{P}_{\mathscr{A}}\mathscr{P}_{\mathscr{R}}\mathscr{P}_{\mathscr{H}}\mathbf{q} = \mathscr{P}\mathbf{q}, \tag{4.44}$$

where $\mathscr{P} = \mathscr{P}_{\mathscr{A}}\mathscr{P}_{\mathscr{R}}\mathscr{P}_{\mathscr{H}}$ represents the SSA filter (Oropeza and Sacchi, 2011).

Enshaeifar et al. (2016) argues that second-order statistical information from the signal can be used to enhance the quaternion-valued SSA (QSSA) by incorporating information from the three involutions (equations 4.9 to 4.11) in an augmented trajectory matrix. This is the difference between AQSSA and QSSA. In AQSSA, the standard trajectory matrix $\mathbf{T}$ (equation 4.39) is augmented by all three involutions to generate $\mathbf{T}^a \in \mathbb{H}^{4L\times K}$ as

$$\mathbf{T}^a = [\mathbf{T}^T \quad \mathbf{T}^{iT} \quad \mathbf{T}^{jT} \quad \mathbf{T}^{kT}]^T. \tag{4.45}$$

By doing so, one expects to capture complete second order information of the signal by exploiting the covariance between components of the quaternion signal (see Took and Mandic (2011)).

As it is known, p linear events in $t-x$ can be represented by the superposition of p complex sinusoids in x by adopting autoregressive models of order p (Oropeza and Sacchi, 2011; Sacchi et al., 2017). This is at the core of the justification to approximate the trajectory matrix to its rank-p version. Using quaternion vector autoregressive (VAR) models (Ginzberg and Walden, 2013), extension of this argument to the quaternion domain can be tested. Following Ginzberg and Walden (2013), the VAR process of order p can be written as

$$q[n] = A_1 q[n-1] + A_2 q[n-2] + ... + A_p q[n-p], \tag{4.46}$$

where $A_i \in \mathbb{H}$ are the AR coefficients, and it is assumed that the process has no innovation and it is zero-mean. By substitution of equation 4.46 into 4.39, one can show that this matrix has rank p, where p is the order of the AR process, i.e., the number of dips in the data. This results, therefore, seems to justify the rank-p approximation of the trajectory matrix in the quaternion domain. However, as stated by Ginzberg and Walden (2013), this process describes proper[11] quaternion signals. In the case of improper signals, widely linear quaternion VAR processes, which takes in consideration the involutions of the signal, should be used. This process is written as (Ginzberg and Walden, 2013)

$$q[n] = A_1 q[n-1] + B_1 q^i[n-1] + C_1 q^j[n-1] + D_1 q^k[n-1] + ... + D_p q^k[n-p]. \tag{4.47}$$

[11] Proper signals are such that have vanishing complementary covariance matrices, i.e., covariance between data components is zero. These are also called circular signals.

where A_i, B_i, C_i, and $D_i \in \mathbb{H}$ are the AR coefficients. Equation 4.47 should be used to justify the low-rank approximation when using AQSSA. It is expected that the addition of linear operations on the involutions will increase the rank of the trajectory matrix.

In order to simultaneously reconstruct and denoise multicomponent data, one can use the iterative imputation algorithm described in Oropeza and Sacchi (2011)

$$\mathbf{q}^{iter} = \alpha^{iter}\mathbf{q}^{obs} + (\mathbf{1} - \alpha^{iter})\mathscr{S}\mathscr{P}\mathbf{q}^{iter-1} + (\mathbf{1} - \mathscr{S})\mathscr{P}\mathbf{q}^{iter-1}, \tag{4.48}$$

where $\mathbf{q}^{obs}$ is the noisy quaternion signal with missing observations, the number of iterations is denoted by $iter = [1, \dots, iter_{max}]$, $\mathscr{S}$ denotes the sampling operator, $\mathbf{1}$ is a $nx \times 1$ vector of ones, and α is a iteration dependent trade-off parameter set to linearly decrease from 1 to 0 as iteration number increases (Oropeza and Sacchi, 2011).

The prediction gain R_i is used to quantify (in dB) and evaluate the performance of the algorithms, which is given by

$$R_i = 10\log_{10}\frac{\|\mathbf{U}_i^o\|_2^2}{\|\mathbf{U}_i - \mathbf{U}_i^o\|_2^2}, \tag{4.49}$$

where $\mathbf{U}_i^o$ and $\mathbf{U}_i$ are the true (noise-free and complete) and reconstructed data, respectively, with $i = (x, y, z)$ representing the (real) component of the data.

Examples

An example will be used to test the application of QSSA and AQSSA in denoising and reconstruction of multicomponent 2D seismic data. The results of denoising and reconstruction are compared against single-component SSA (denoted by SSA).

In order to approximate the condition of linear events in $t - x$, a window of the data shown in figure 4.1 is used (Figure 4.3). This window has 2 to 3 quasi-linear events, accounting for approximately 0.22 seconds of data in 51 traces, showing different polarities and also offset-dependent variations in amplitude. The number of time samples is $nt = 111$ and in space $nx = 51$. Figure 4.4 shows the noisy windowed data with SNR$= 2$ for all components.

By varying the eigenaxis of the QFT, it is possible to gain some further insight on its importance and, perhaps, how to properly define it. Figures 4.5 and 4.6 show the frequency slices at 20 Hz and 40

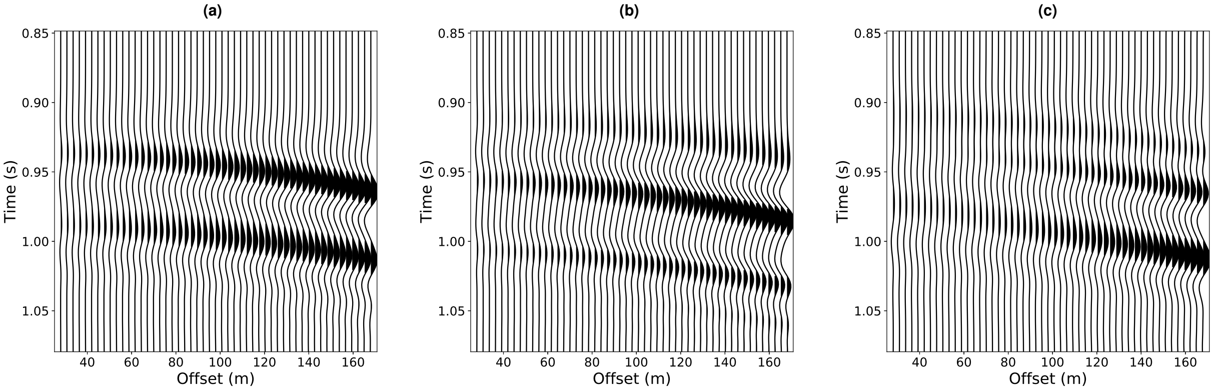


Figure 4.3: Window of 2D 3C synthetic seismic data.

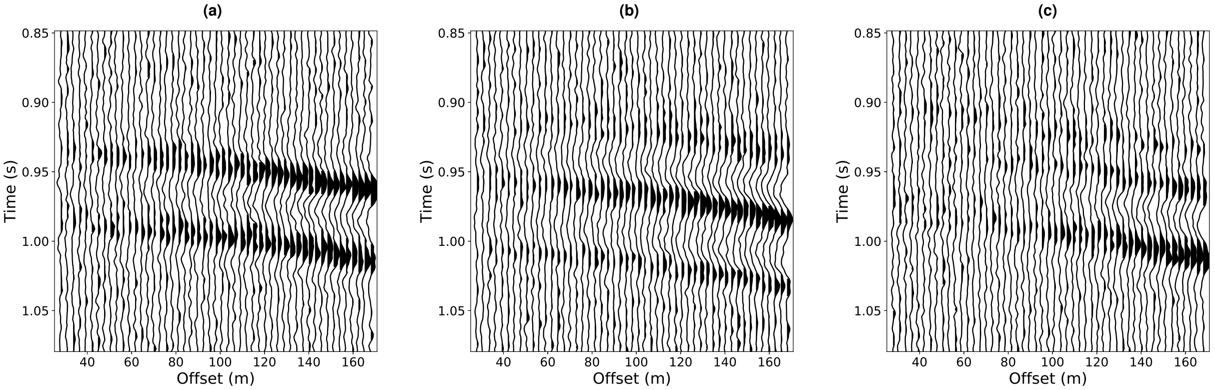


Figure 4.4: Window of 2D 3C synthetic seismic data contamined with noise.

Hz, respectively, for both noise-free and noisy cases, to illustrate the consequences of the eigenaxis choice in the QFT. The axis were set to (a)$\mu = i$, (b)$\mu = j$, (c)$\mu = k$, and (d)$\mu = \frac{1}{\sqrt{3}}(1, 1, 1)$. Observe that the components of all basis are very similar to each other. Also, notice that the scalar component of (a), (b) and (c) represent the real part of the complex FT of the purely imaginary signal defined by the respective component (i,j,k). That happens because the signal is initially defined as a pure quaternion, and no mixing happens with the component in its simplex part since $S(q) = 0$. In addition, observe that by setting the eigenaxis to $\mu = \frac{1}{\sqrt{3}}(1, 1, 1)$ indeed mixes the quaternion components. That seems to reduce the noise level in each one of them, as more easily observed for the frequency slice of 20 Hz (Figure 4.5(d)). That is the reason why this eigenaxis is employed in this paper. Also, the left-sided QFT is always employed.

Noise and missing observations will increase the rank of the trajectory matrix, just as in the single-component case. The spectrum of singular values of the trajectory matrix and its augmented version, obtained for the frequency of 20 Hz, are shown in figure 4.7. It is important to notice that the curvature observed in the windowed data increases the rank of the trajectory matrix, i.e., the small singular values do not vanish as in the ideal case of linear events. The same would be observed in the case of real data. Hence, the number of singular values used in the rank-reduction operator should be increased to two or three times the number of expected dips in the data. Nevertheless, the noise-free singular value distribution illustrates the approximate low-rank character of both matrices, which further justifies the development of rank-reduction algorithms for processing multicomponent seismic data in the quaternion domain.

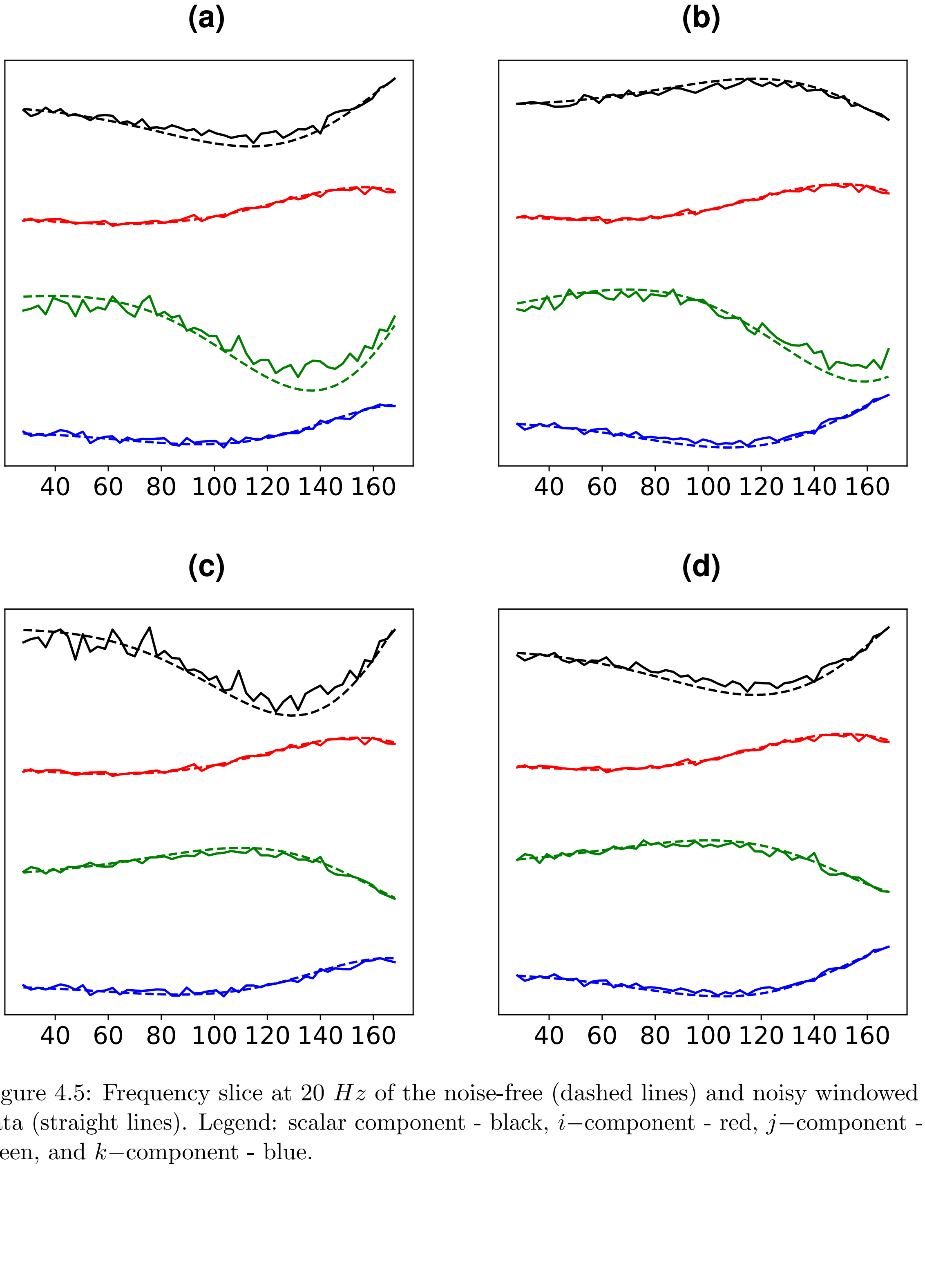


Figure 4.5: Frequency slice at 20 Hz of the noise-free (dashed lines) and noisy windowed data (straight lines). Legend: scalar component - black, $i-$component - red, $j-$component - green, and $k-$component - blue.

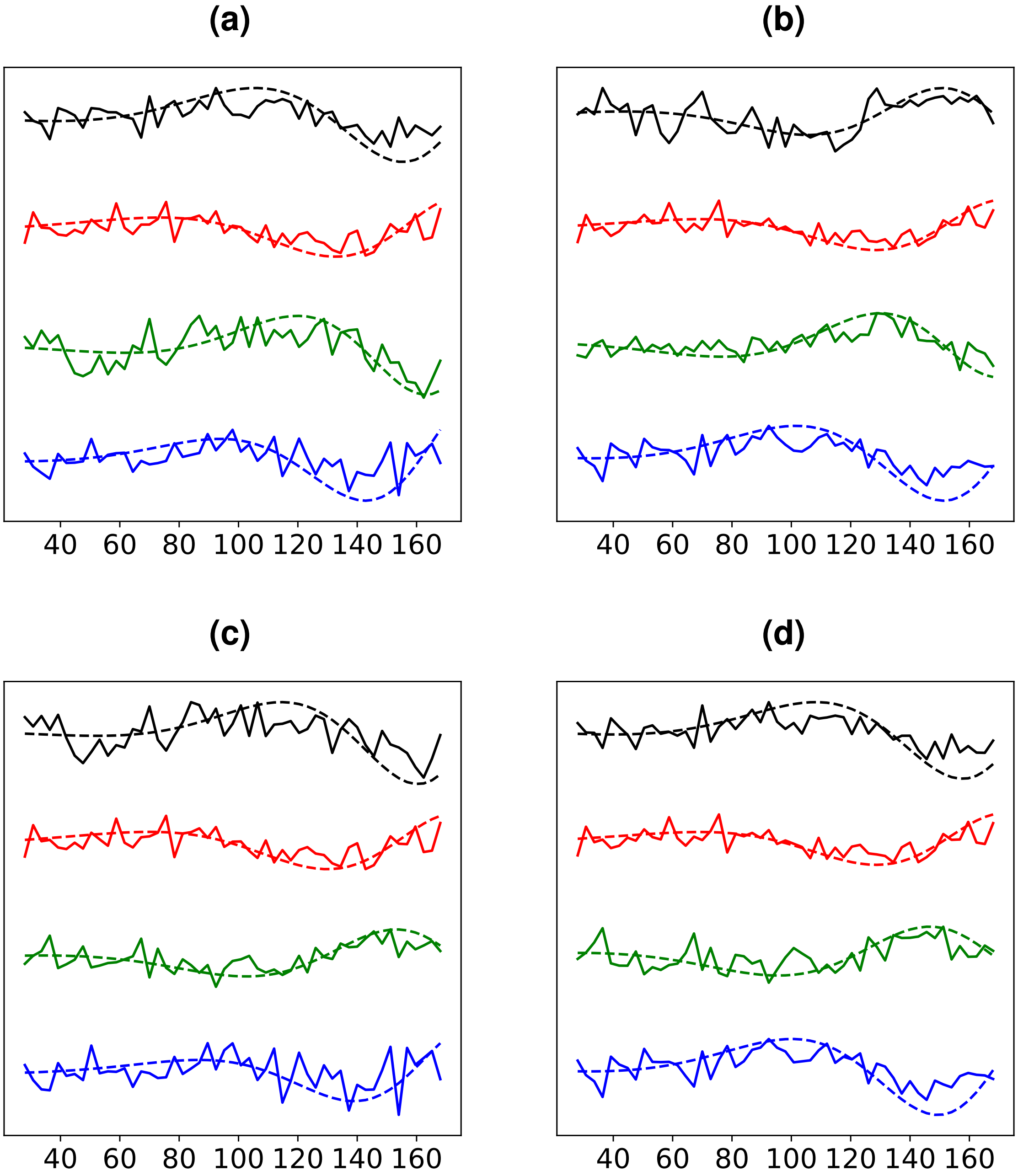


Figure 4.6: Frequency slice at 40 Hz of the noise-free (dashed lines) and noisy windowed data (straight lines). Legend: scalar component - black, $i-$component - red, $j-$component - green, and $k-$component - blue.

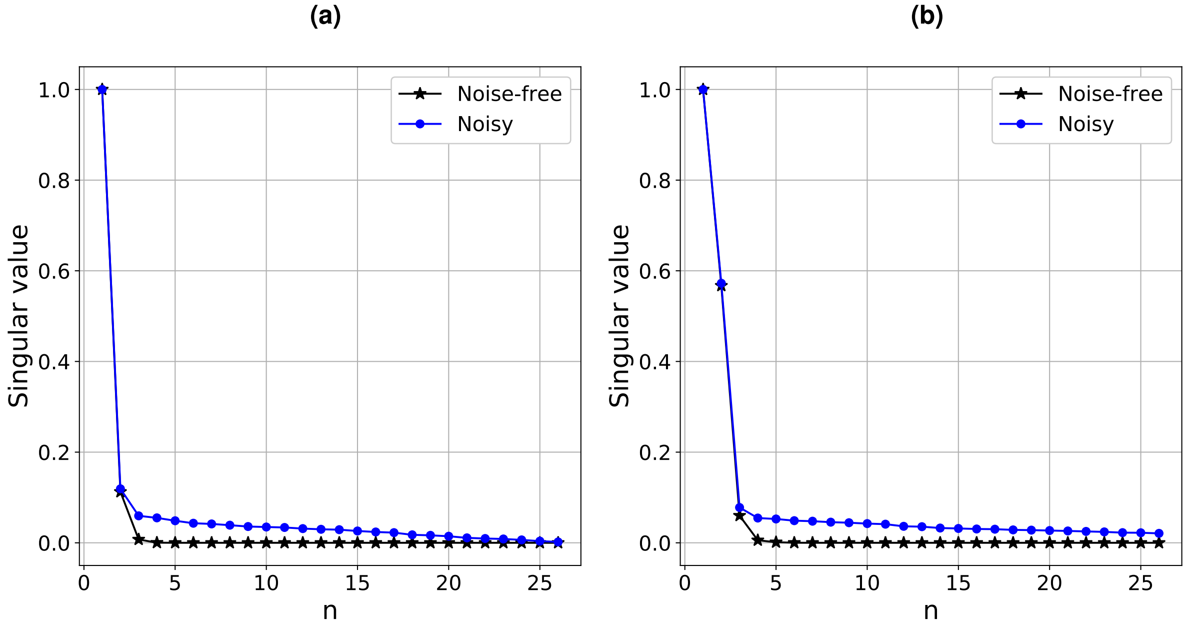


Figure 4.7: Singular values of trajectory matrix for frequency slice at 20 Hz of the noise-free and noisy windowed data.

For all algorithms, the frequency range used in the simulations is 0 to 60 Hz. The window length is set to $L = \lfloor \frac{nx}{2} \rfloor + 1$, where $\lfloor\rfloor$ denotes the integer part of its argument, so that the trajectory matrix is closest to a square matrix (Oropeza and Sacchi, 2011). The rank is set to $p = 4$ for all algorithms. The number of iterations for the imputation algorithm is set to 30 in all cases.

Figures 4.8 to 4.10 show the denoising results for SSA, QSSA and AQSSA, respectively. For this example, the reconstruction gains indicate that single-component SSA can be outperformed by the quaternionic approaches but the results are yet not consistent among components.

Table 4.2: Denoising quality factors.

	SSA	QSSA	AQSSA
$R_x(dB)$	9.2257	9.1901	9.3532
$R_y(dB)$	8.7778	7.1951	7.5778
$R_z(dB)$	8.2872	8.1111	8.7950

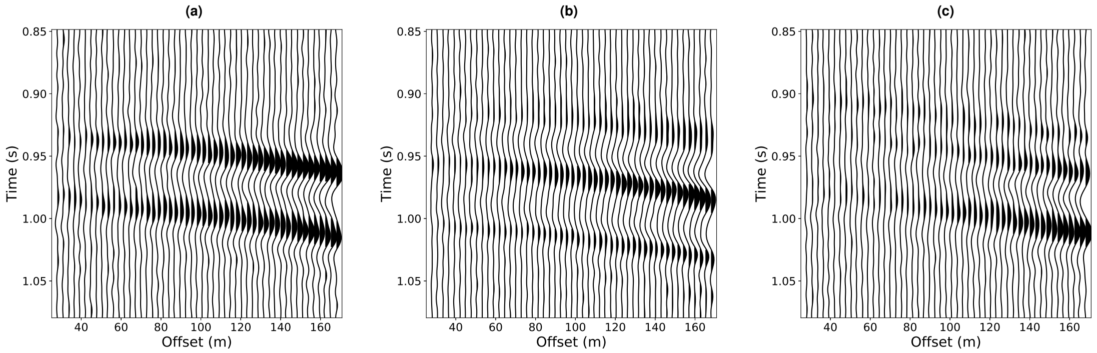


Figure 4.8: Results for single-component SSA denoising.

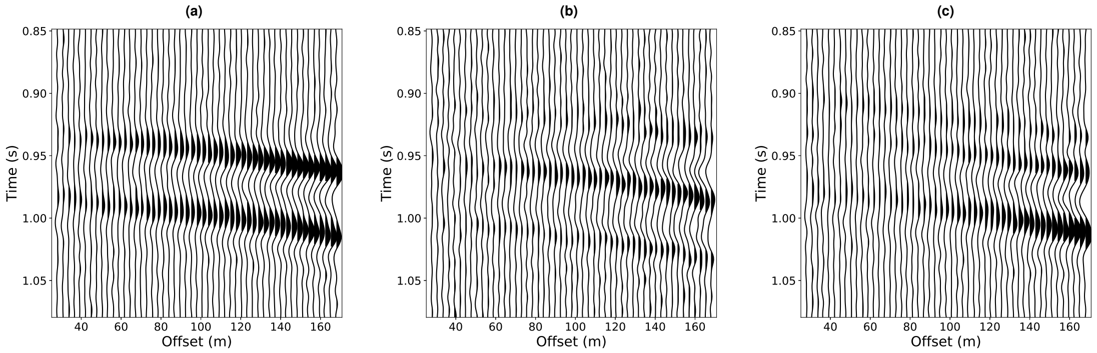


Figure 4.9: Result for QSSA denoising.

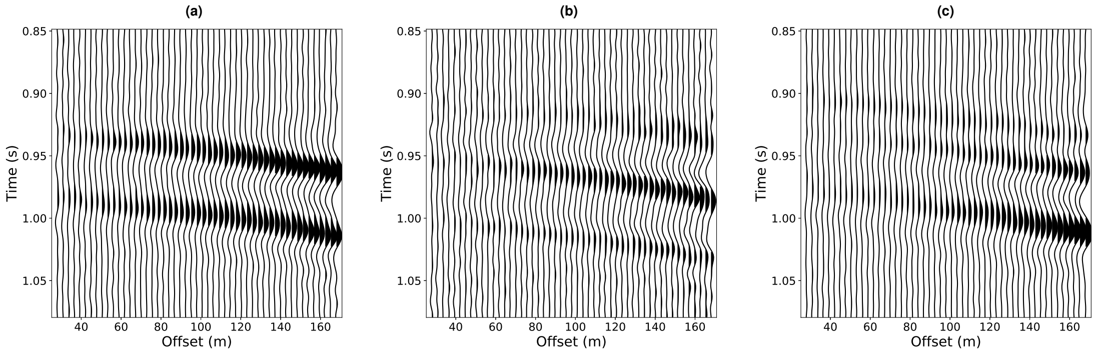


Figure 4.10: Result for AQSSA denoising.

It is also possible to test QSSA and AQSSA for the case where missing observations disrupt the data (Figure 4.11). The data has 40% of its traces randomly decimated. Figures 4.12 to 4.14 show the reconstruction results for SSA, QSSA and AQSSA, respectively, in the case of noise-free data. For this example, both QSSA and AQSSA were able to outperform SSA for all components.

Table 4.3: Reconstruction quality factors for noise-free decimated data.

	SSA	QSSA	AQSSA
$R_x(dB)$	14.9801	15.7723	19.2908
$R_y(dB)$	14.7861	15.0882	20.0428
$R_z(dB)$	8.5857	11.8549	11.8776

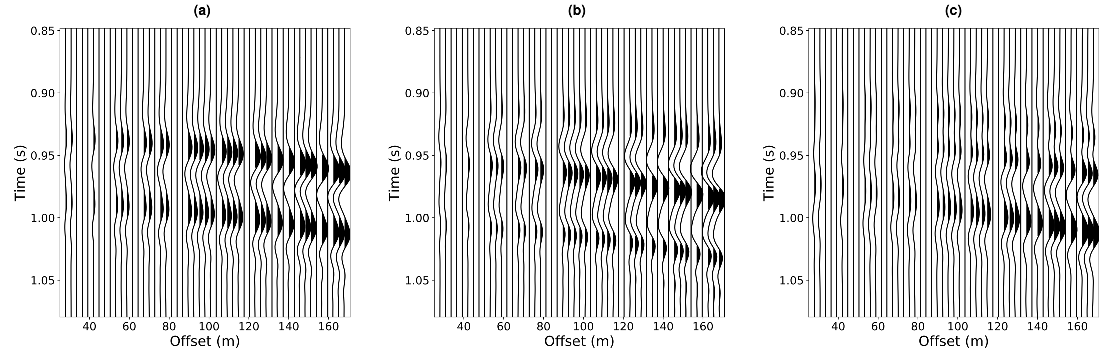


Figure 4.11: Noise-free decimated data with 40% of traces missing

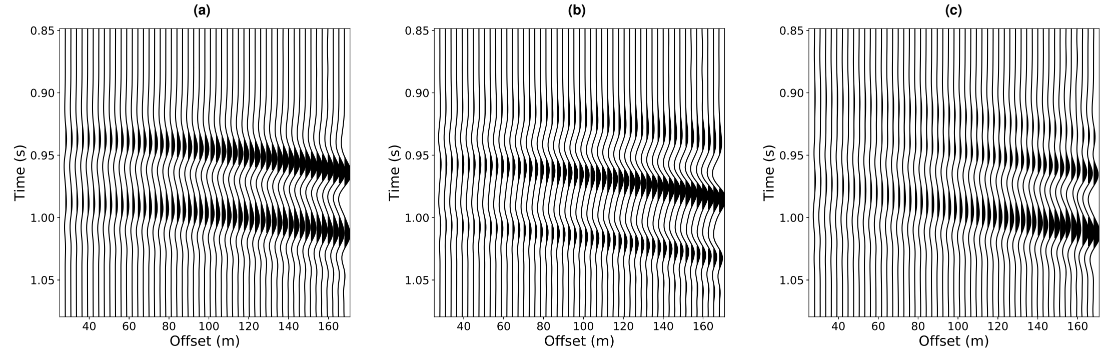


Figure 4.12: Results for single-component SSA reconstruction.

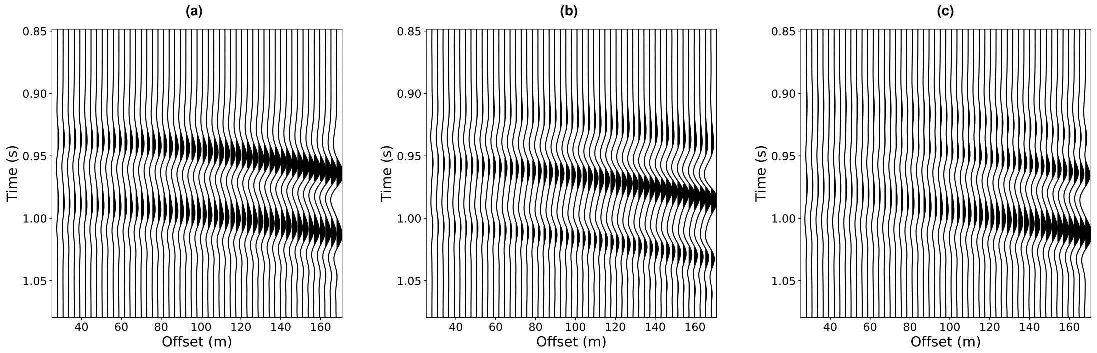


Figure 4.13: Result for QSSA reconstruction.

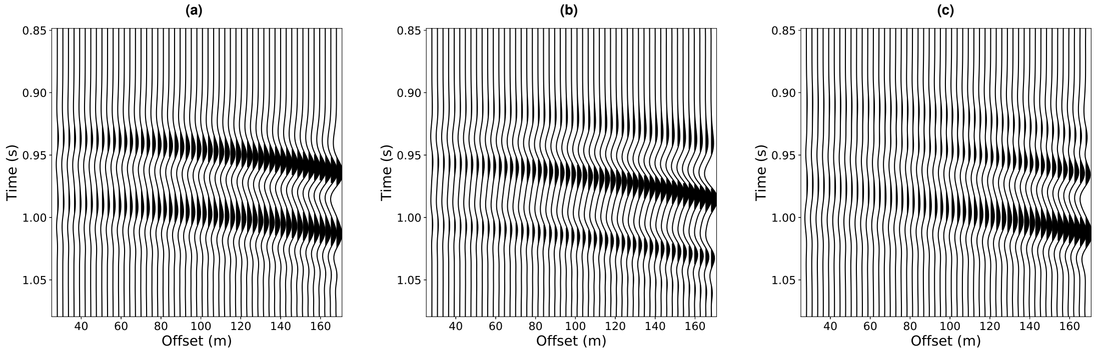


Figure 4.14: Result for AQSSA reconstruction.

The final test is to simultaneously denoise and reconstruct the data using QSSA and AQSSA. Figure 4.15 shows the data with SNR=2 and randomly decimated to 40% of its traces. Figures 4.16 to 4.18 shows the results for SSA, QSSA and AQSSA, respectively. As indicated in table 4.4, superiority in the quaternionic approaches over component-wise processing can be observed in this example for only for the x component.

Table 4.4: Reconstruction quality factors for noisy decimated data.

	SSA	QSSA	AQSSA
$R_x(dB)$	4.9872	5.529	6.9222
$R_y(dB)$	5.8574	4.2748	4.9912
$R_z(dB)$	6.4979	4.4450	5.7709

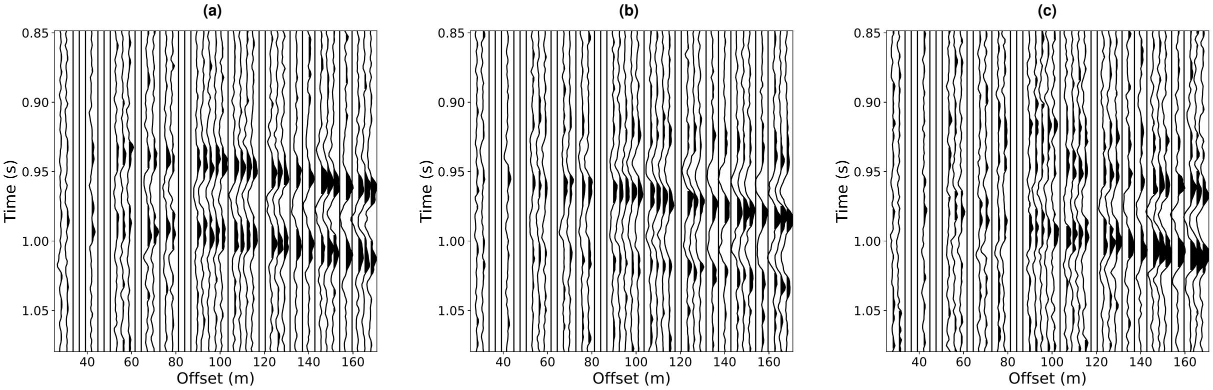


Figure 4.15: Noise-free decimated data with 40% of traces missing.

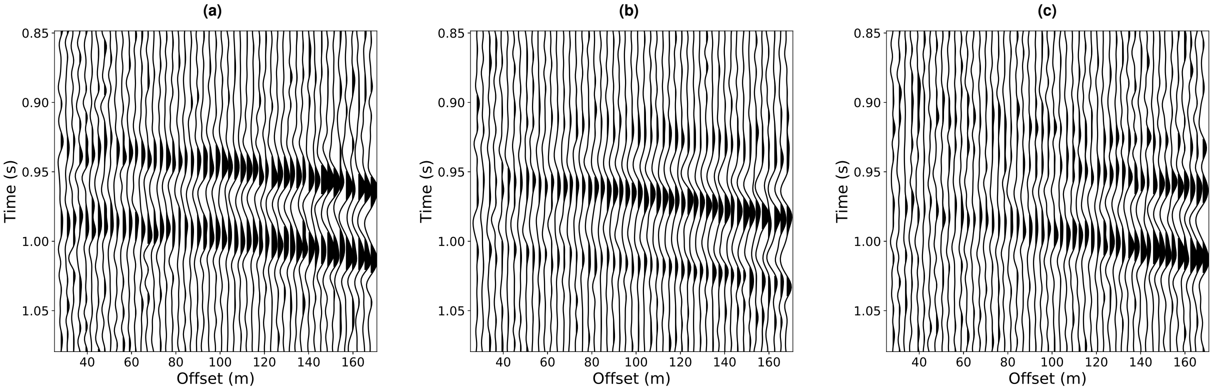


Figure 4.16: Results for single-component SSA reconstruction.

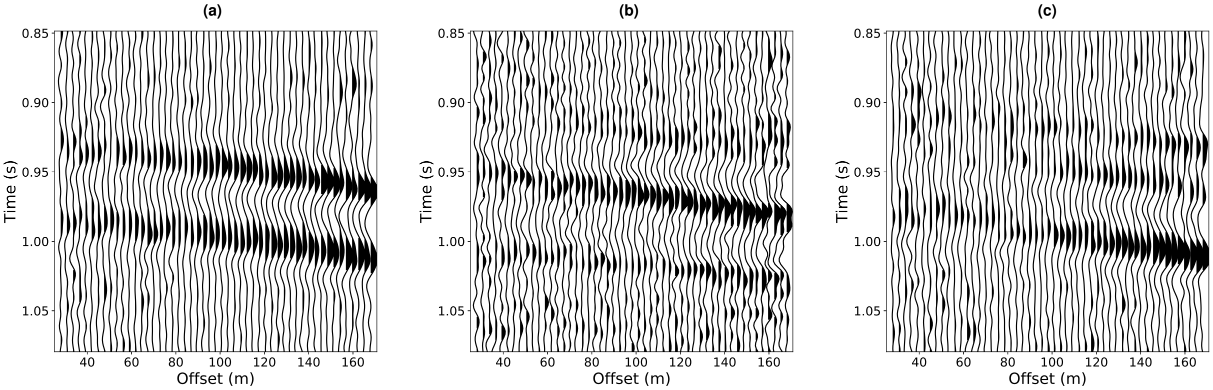


Figure 4.17: Result for QSSA reconstruction.

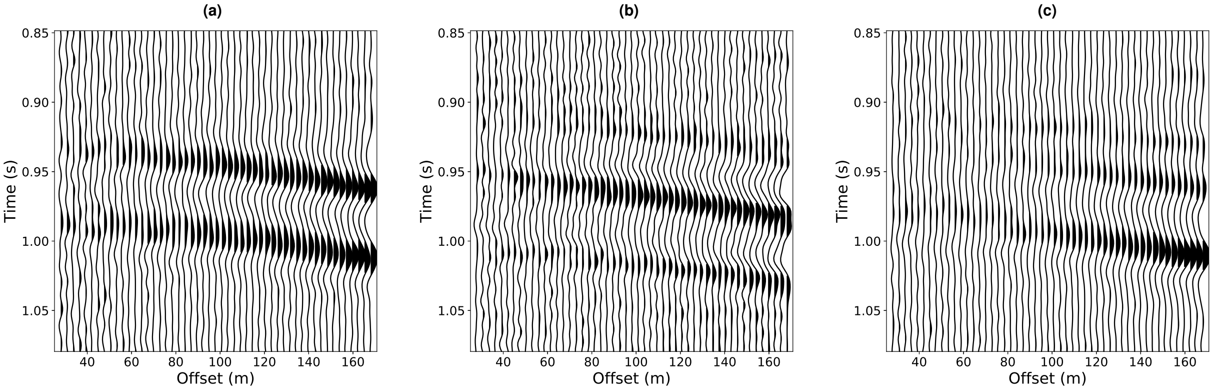


Figure 4.18: Result for AQSSA reconstruction.

4.5 Conclusions and Discussion

In this paper, SSA denoising and reconstruction in the vector field case is investigated using the quaternion signal model. The MC data is defined as a pure quaternion signal which can then be treated with quaternionic signal processing tools. QSVD and QFT are briefly presented as such tools. QSSA and its augmented version are introduced as possible techniques for simultaneous denoising and reconstruction of multicomponent seismic data, in a holistic approach, through rank-reduction. As in its real counterpart, the trajectory matrix in the ideal case, where the data is not corrupted by noise and missing observations, is of low rank. The problem of noise reduction and data completion is, therefore, posed as a rank reduction problem since noise and dead traces will increase the rank of the trajectory matrix. The iterative imputation algorithm is also generalized to the quaternion domain, allowing for simultaneous MC seismic data denoise and reconstruction.

Employment of the field of quaternions, as shown in this paper, seems promising. On the other hand, QSSA and AQSSA are still not consistent with the tests in the sense that its performances are, at times, inferior to the scalar case. The same is not observed when the comparison is made between both quaternionic approaches, where AQSSA outperforms QSSA in every test. More has to be investigated regarding the quaternion vector autoregressive models so a better understanding can be achieved for the parameter setting, especially in AQSSA. Furthermore, the question of a possible preferential eigenaxis (or basis in $\mathbb{H}$) for the QFT is still open for seismic signals. A thought is that directional information from multicomponent data could perhaps be obtained using polarization filters.

Regarding the computational aspects of quaternion-based techniques, it should be kept in mind that the possibility of a concise approach to process vector signals comes with a price. The computational load of hypercomplex techniques is heavier than its real-valued counterparts due to its 4D structure. For instance, addition in the quaternion domain requires four real-valued additions, and multiplication requires 16 real-valued multiplications (Alfsmann et al., 2007). Thus, techniques already known for being computationally heavy, such as SVD, also impose a difficulty in the development of such techniques. On the other hand, this also opens the way to other exciting topics such as the extension of fast rank-reduction algorithms to the quaternion-domain, such as randomized SVD or QR.

4.6 Acknowledgments

The sponsors of the Signal Analysis and Imaging Group at the University of Alberta are appreciated for their support. The developers of the QTFM are also acknowledged.

References

Alfsmann, D., H. G. Göckler, S. J. Sangwine, and T. A. Ell, 2007, Hypercomplex algebras in digital signal processing: Benefits and drawbacks: Signal Processing Conference, 2007 15th European, 1322–1326.

Boyd, O. S., 2006, An efficient matlab script to calculate heterogeneous anisotropically elastic wave propagation in three dimensions: Computers & geosciences, **32**, 259–264.

Bulow, T., and G. Sommer, 2001, Hypercomplex signals-a novel extension of the analytic signal to the multidimensional case: IEEE Transactions on signal processing, **49**, 2844–2852.

Bunse-Gerstner, A., R. Byers, and V. Mehrmann, 1989, A quaternion QR algorithm: Numerische Mathematik, **55**, 83–95.

Delsuc, M. A., 1988, Spectral representation of 2D NMR spectra by hypercomplex numbers: Journal of Magnetic Resonance (1969), **77**, 119–124.

Ell, T. A., 1992, Hypercomplex spectral transformations: PhD Thesis.

Ell, T. A., N. Le Bihan, and S. J. Sangwine, 2014, Quaternion Fourier transforms for signal and image processing: John Wiley & Sons.

Ell, T. A., and S. J. Sangwine, 2007, Hypercomplex Fourier transforms of color images: IEEE Transactions on image processing, **16**, 22–35.

Enshaeifar, S., S. Kouchaki, C. C. Took, and S. Sanei, 2016, Quaternion singular spectrum analysis of electroencephalogram with application in sleep analysis: IEEE Transactions on Neural Systems and Rehabilitation Engineering, **24**, 57–67.

Ernst, R. R., G. Bodenhausen, A. Wokaun, et al., 1987, Principles of nuclear magnetic resonance in one and two dimensions.

Ghil, M., M. Allen, M. Dettinger, K. Ide, D. Kondrashov, M. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, et al., 2002, Advanced spectral methods for climatic time series: Reviews of geophysics, **40**.

Ginzberg, P., and A. T. Walden, 2013, Quaternion VAR modelling and estimation: IEEE Transactions on Signal Processing, **61**, 154–158.

Golyandina, N., and A. Zhigljavsky, 2013, Singular spectrum analysis for time series: Springer Science & Business Media.

Grandi, A., A. Mazzotti, and E. Stucchi, 2007, Multicomponent velocity analysis with quaternions: Geophysical Prospecting, **55**, 761–777.

Hitzer, E., and S. J. Sangwine, 2013, Quaternion and Clifford Fourier transforms and wavelets: Springer.

Janzen, S., J. Cheng, and M. D. Sacchi, 2017, SSA for multicomponent data: GeoConvention2017.

Le Bihan, N., and G. Ginolhac, 2001, Subspace methods for 3D arrays: Workshop on Physics in Signal and Image Processing (PSIP), 359–364.

Le Bihan, N., and J. Mars, 2004, Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing: Signal processing, **84**, 1177–1199.

Loring, T. A., 2012, Factorization of matrices of quaternions: Expositiones Mathematicae, **30**, 250–267.

Menanno, G. M., and A. Mazzotti, 2012, Deconvolution of multicomponent seismic data by means of quaternions: Theory and preliminary results: Geophysical Prospecting, **60**, 217–238.

Oropeza, V., and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Sacchi, M. D., J. Cheng, and S. Janzen, 2017, Theory and application of vector singular spectrum analysis (SSA) for multicomponent seismic data reconstruction: GeoConvention2017.

Sajeva, A., and G. Menanno, 2017, Characterisation and extraction of a Rayleigh-wave mode in vertically heterogeneous media using quaternion SVD: Signal Processing, **136**, 43–58.

Sangwine, S., 2005, Quaternion toolbox for Matlab: http://qtfm. sourceforge. net/.

Sangwine, S. J., and N. Le Bihan, 2006, Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion Householder transformations: Applied mathematics and computation, **182**, 727–738.

Stanton, A., and M. D. Sacchi, 2013, Vector reconstruction of multicomponent seismic data: Geophysics, **78**, V131–V145.

Took, C. C., and D. P. Mandic, 2011, Augmented second-order statistics of quaternion random signals: Signal Processing, **91**, 214–224.

Trickett, S., 2008, F-xy Cadzow noise suppression, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 2586–2590.

Ward, J. P., 2012, Quaternions and Cayley numbers: Algebra and applications: Springer Science & Business Media.

Witten, B., and J. Shragge, 2006, Quaternion-based signal processing, *in* SEG Technical Program Expanded Abstracts 2006: Society of Exploration Geophysicists, 2862–2866.

Zhang, F., 1997, Quaternions and matrices of quaternions: Linear algebra and its applications, **251**, 21–57.

Controlling signal leakage in FX Deconvolution: A discussion on a paper presented at SEG by Chiu et al.

Mauricio D Sacchi [1]

Abstract

A recent paper entitled ”Residual-signal recovery using a pattern-based method” by Chiu et al. (2017) presents a solution to a long-standing problem in seismic data processing. The authors adopt FX deconvolution and present an algorithm that is capable of recovering the signal that has leaked into the error panel. I will investigate the proposed algorithm and attempt to understand its premises and limitations.

5.1 Introduction

FX deconvolution is one of the traditional tools for enhancing seismic records. The method is often adopted to remove incoherent noise (Canales, 1984; Gulunay, 1986). FX deconvolution poses denoising as a linear prediction problem in the frequency-space domain and it corresponds to assuming an autoregressive (AR) model in space. In other words, an AR model is used to predict the spatial signal, and the prediction error identifies the offending noise in the data. A problem with FX deconvolution is signal leakage. The latter is primarily due to the inability of the FX filter to properly model additive noise. In fact, the AR model representation adopted by FX deconvolution can only incorporate the noise in the form of a random innovation. In a recent article, Chiu et al.

[1] Email: msacchi@ualberta.ca

(2017) proposed to attenuate signal leakage by reapplying the FX deconvolution operator to the error and, by this way, they propose to estimate the coherent portion of the signal that leaked into the error. The algorithm of Chiu et al. (2017) can be summarized as follows:

1 Estimate the prediction filter $\mathbf{a}$ from the signal $\mathbf{d} \rightarrow \mathbf{a}$

2 Apply the prediction filter to the signal to obtain a denoised signal $\rightarrow \hat{\mathbf{d}}$

3 Compute error/noise $\rightarrow \mathbf{e} = \mathbf{d} - \hat{\mathbf{d}}$

4 Apply the prediction filter[2] to the error $\mathbf{e}$ to extract the coherent part of signal in the error $\rightarrow \mathbf{r}$

5 Improve the estimation by adding $\mathbf{r}$ to the estimated signal in [**2**] $\rightarrow \hat{\mathbf{d}} + \mathbf{r}$

In next sections, we will try to bring to light this intriguing algorithm.

5.2 Theory

5.2.1 Prediction filtering

In the frequency-space (FX) domain we will denote the data as $d(\omega, x)$. To avoid notational clutter, we will drop the dependency on ω, and we understand that the process is carried out for all frequencies ω. We will also assume data evenly discretized in space $d(\omega, n-1\Delta x) \equiv d_n, n = 0 \dots N-1$ where N is the number of traces. In FX deconvolution, one assumes the following linear prediction model

$$d_n = a_1\, d_{n-1} + a_2\, d_{n-2} + \dots a_L\, d_{n-L} + e_n\,. \tag{5.1}$$

Consider, for instance, the case where the order of the AR model is given by $L = 3$ and $N = 8$. The AR model can be written in terms of the following system of equations

[2] The filter is already computed in [**1**]

$$\underbrace{\begin{pmatrix} d_2 & d_1 & d_0 \\ d_3 & d_2 & d_1 \\ d_4 & d_3 & d_2 \\ d_5 & d_4 & d_3 \\ d_6 & d_5 & d_4 \end{pmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}}_{\mathbf{a}} - \underbrace{\begin{pmatrix} d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{pmatrix}}_{\mathbf{d}} \approx \underbrace{\begin{pmatrix} e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix}}_{\mathbf{e}} . \tag{5.2}$$

The above system of equations can be written in compact form via the following expression

$$\mathbf{M}\,\mathbf{a} - \mathbf{d} = \mathbf{e} . \tag{5.3}$$

The coefficients of the AR model are estimated by minimizing the Euclidean norm of the error (Gulunay, 2017). At this point it is important to point out that we also need to incorporate a regularization term to guarantee the stability of the solution. Therefore, we will minimize a cost function that consists of two parts

$$\Phi = \|\mathbf{M}\,\mathbf{a} - \mathbf{d}\|_2^2 + \mu\|\mathbf{a}\|_2^2 . \tag{5.4}$$

The first term of the cost function is the misfit, the second is the regularization term that is given, in this case, by the squared l_2 norm of the filter. The trade-off parameter (often called pre-whitening) μ controls predictability and noise rejection. The trade-off parameter μ in conjunction with the length of the filter L need to be adjusted to properly model the signal and reject the noise. The minimization of the cost function Φ leads to the damped least-squares solution

$$\hat{\mathbf{a}} = (\mathbf{M}^H\mathbf{M} + \mu\mathbf{I})^{-1}\mathbf{M}^H\mathbf{d} . \tag{5.5}$$

Once the filter is compute via expression 5.5 one can predict the data as follows

$$\hat{\mathbf{d}} = \mathbf{M}\hat{\mathbf{a}} \tag{5.6}$$

where

$$\hat{\mathbf{d}} = (\hat{d}_3, \hat{d}_4, \hat{d}_5, \hat{d}_6, \hat{d}_7)^T \tag{5.7}$$

clearly, the first two points $\hat{d}_1, \hat{d}_2$ were not predicted by our model. However, one can easily use backwards prediction to predict these two points as well. Let us turn our attention into the signal and predicted error

$$\begin{aligned}\hat{\mathbf{d}} &= \mathbf{M}\hat{\mathbf{a}} \\ &= \mathbf{M}(\mathbf{M}^H\mathbf{M}+\mu\mathbf{I})^{-1}\mathbf{M}^H\mathbf{d} \\ &= \mathbf{A}\mathbf{d}\,.\end{aligned} \tag{5.8}$$

where the data resolution matrix $\mathbf{A} = \mathbf{M}(\mathbf{M}^H\mathbf{M}+\mu\mathbf{I})^{-1}\mathbf{M}^H$. Similarly, one can compute the estimator of the noise

$$\begin{aligned}\hat{\mathbf{e}} &= \mathbf{d}-\hat{\mathbf{d}} \\ &= (\mathbf{I}-\mathbf{A})\,\mathbf{d}\,.\end{aligned} \tag{5.9}$$

The tradeoff parameter μ and the filter length L control the ability of the prediction filter predict the signal versus noise rejection. In general, filters that are too aggressive in terms of random noise removal will lead to excessive signal leakage in the error panel. Conversely, filters that preserve the signal cannot remove random noise (Gulunay, 2017).

5.2.2 The coherent portion of the signal in the estimated noise

We will assume that the error $\mathbf{e}$ contains coherent signal. Chiu et al. (2017) proposes to apply the filter $\mathbf{A}$ to the error to estimate its coherent part

$$\begin{aligned}\mathbf{r} &= \mathbf{A}\hat{\mathbf{e}} \\ &= \mathbf{A}(\mathbf{I}-\mathbf{A})\mathbf{d}\end{aligned} \tag{5.10}$$

where $\mathbf{r}$ is the coherent portion of the signal extracted from the estimated error $\hat{\mathbf{e}}$. The final estimator of the "improved" signal is given by

$$\begin{aligned}\hat{\hat{\mathbf{d}}} &= \hat{\mathbf{d}}+\mathbf{r} \\ &= \mathbf{A}\mathbf{d}+\mathbf{A}(\mathbf{I}-\mathbf{A})\mathbf{d} \\ &= (2\mathbf{A}-\mathbf{A}\mathbf{A})\mathbf{d}\end{aligned} \tag{5.11}$$

now we anlysize the form $\mathbf{AA}$

$$\mathbf{A}\,\mathbf{A} = \mathbf{M}(\mathbf{M}^H\mathbf{M} + \mu\mathbf{I})^{-1}\mathbf{M}^H\ \mathbf{M}(\mathbf{M}^H\mathbf{M} + \mu\mathbf{I})^{-1}\mathbf{M}^H \tag{5.12}$$

From where one can see that when $\mu = 0$ $\mathbf{A}$ is an idempotent matrix: $\mathbf{A}\,\mathbf{A} = \mathbf{A}$. Therefore when $\mu = 0$

$$\hat{\hat{\mathbf{d}}} = \mathbf{A}\mathbf{d} = \hat{\mathbf{d}}\,, \tag{5.13}$$

In other words, if $\mu = 0$ there is no advantage in reapplying the prediction filter to the error. Interestingly, the flow proposed by Chiu et al. (2017) suggests that one might prefer to use a significant trade-off parameter to attenuate the noise and then estimate the signal leaked into the error panel.

5.3 Example

Figures 5.2 and 5.2 portray an example where I tried to use the flow proposed by Chiu et al. (2017). In this case, I adopted a filter of length $L = 10$ points, and $\mu = 1\%$. The trade-off parameter is percentual of the zero-lag autocorrelation coefficient of the input data. In other words $\mu = 1\%$ corresponds to $\mu = 0.01 \times r_0$ where r_0 is the zero-lag autocorrelation coefficient of the input signal. It is evident that the signal that has leaked in the first error panel was correctly modeled and incorporated back in the predicted signal. However, it is also evident that, as always, there is no free lunch. The final estimation of the signal (panel e) in Figures 5.2 and 5.2) contain more noise than the initial estimator of the signal in panel b).

I have repeated the test in Figures 5.3 and 5.4. However, in this case I have adopted a filter of length $L = 10$ points, and $\mu = 20\%$. The large trade-off parameter makes the initial filter more aggressive. The latter can be seen in panel b) of Figures 5.3 and 5.4. The final result shows that we have been able to partically eliminate signal leakege at the cost of adding noise to the final estimator of the signal in panel e).

5.4 Conclusion

The flow proposed by Chiu et al. (2017) can be used to attenuate signal leakage in FX deconvolution. However, it is not clear the real benefit of it at this point. In general, my main observation is that

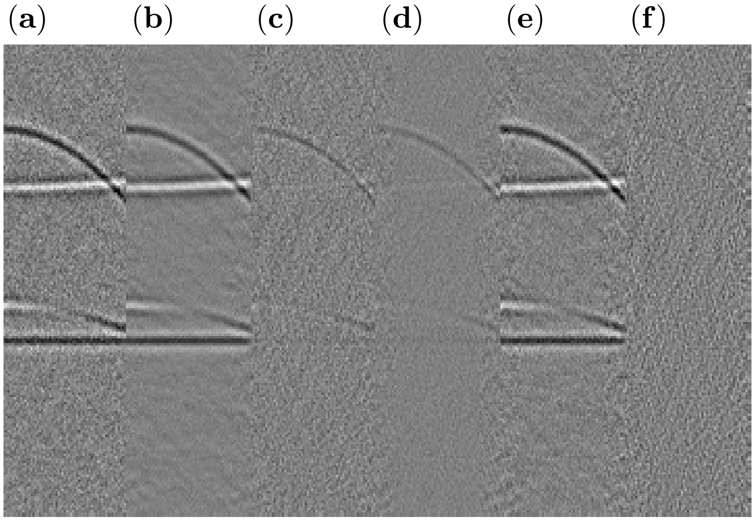


Figure 5.1: Results obtained with $\mu = 1\%$, $L = 10$. a) Data $\mathbf{d}$. b) Predicted data $\hat{\mathbf{d}}$ by FX deconvolution. c) Error $\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}}$. d) Predicted coherent part of the error $\mathbf{r}$. e) New predicted data $\hat{\mathbf{d}} + \mathbf{r}$. f) New error $\mathbf{d} - (\hat{\mathbf{d}} + \mathbf{r})$

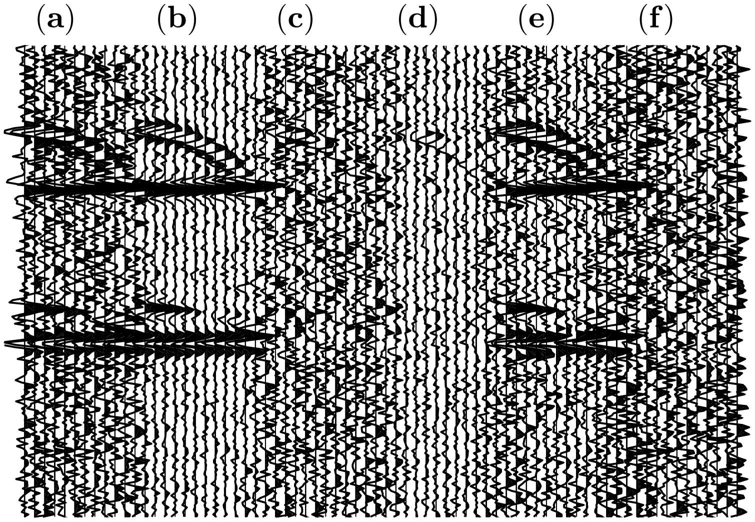


Figure 5.2: Results obtained with $\mu = 1\%$, $L = 10.$, only 1:5 traces plotted in each panel. a) Data $\mathbf{d}$. b) Predicted data $\hat{\mathbf{d}}$ by FX deconvolution. c) Error $\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}}$. d) Predicted coherent part of the error $\mathbf{r}$. e) New predicted data $\hat{\mathbf{d}} + \mathbf{r}$. f) New error $\mathbf{d} - (\hat{\mathbf{d}} + \mathbf{r})$

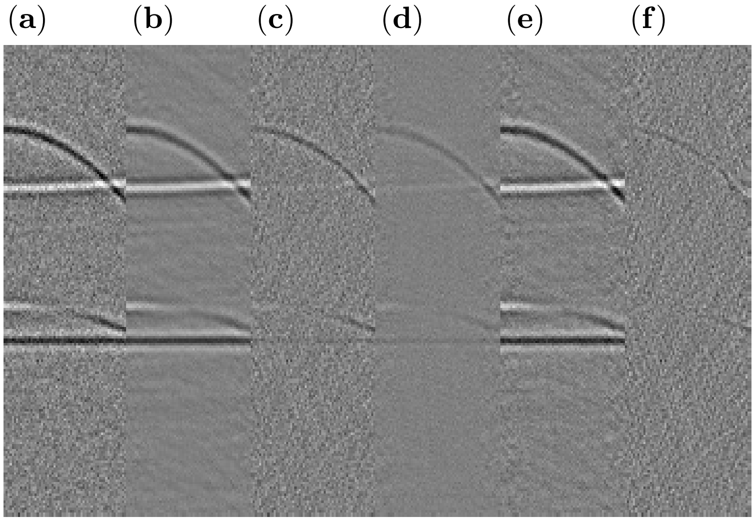


Figure 5.3: Results obtained with $\mu = 20\%$, $L = 10$. a) Data $\mathbf{d}$. b) Predicted data $\hat{\mathbf{d}}$ by FX deconvolution. c) Error $\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}}$. d) Predicted coherent part of the error $\mathbf{r}$. e) New predicted data $\hat{\mathbf{d}} + \mathbf{r}$. f) New error $\mathbf{d} - (\hat{\mathbf{d}} + \mathbf{r})$

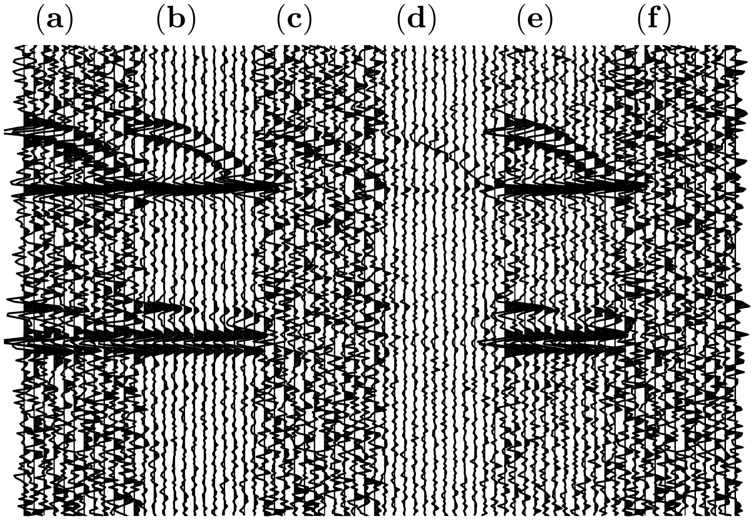


Figure 5.4: Results obtained with $\mu = 20\%$, $L = 10.$, only 1:5 traces plotted in each panel. a) Data $\mathbf{d}$. b) Predicted data $\hat{\mathbf{d}}$ by FX deconvolution. c) Error $\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}}$. d) Predicted coherent part of the error $\mathbf{r}$. e) New predicted data $\hat{\mathbf{d}} + \mathbf{r}$. f) New error $\mathbf{d} - (\hat{\mathbf{d}} + \mathbf{r})$

one can minimize signal leakage at the cost of bringing back noise to the final estimator of the clean data.

A few intriguing questions remain unanswered. For instance, what is the connection, if any, of the flow proposed by Chiu et al. (2017) and the projection filter (Soubaras, 1995).

References

Canales, L. L., 1984, Random noise reduction: SEG Technical Program Expanded Abstracts 1984, 525–527.

Chiu, S., B. Gong, and Z. Meng, 2017, Residual-signal recovery using a pattern-based method: SEG Technical Program Expanded Abstracts 2017, 5069–5073.

Gulunay, N., 1986, Fxdecon and complex wiener prediction filter: SEG Technical Program Expanded Abstracts 1986, 279–281.

Gulunay, N., 2017, Signal leakage in f-x deconvolution algorithms: Geophysics, **82**, W31–W45.

Soubaras, R., 1995, Prestack random and impulsive noise attenuation by *fx* projection filtering: SEG Technical Program Expanded Abstracts 1995, 711–714.

Monogenic signal for features and pattern detection

B. Manzanilla [1]

Abstract

Over the last decade computer vision and pattern recognition have proliferated across many disciplines. Seismic data are more than ever a perfect candidate for the extraction of features and recognition of patterns, specially in large data sets. In times of low oil prices this is a valuable tool to leverage the value of already acquired data given the increase in the acquisition cost to revenue ratio. The basis of any feature extraction and pattern recognition algorithm is the ability to convey to a computer whatever features are of interest to the human interpreter. This task entails the identification of features that are constantly present across data sets -invariants- and the design of tools to measure them. This forms the basis for any other more elaborate pattern recognition and data analysis task. Among those invariants, local amplitude and phase in signals are one of the best descriptors. In the present article, some ideas on how to make use of the Multidimensional Analytic or Monogenic Signal as a tool to identify patterns are set forth. Local orientation, phase and amplitude, are explained, and how to interpret the results. Finally some potential paths for further development are suggested. This report, ponders on the value of the local phase and amplitude as potential starting points in the generation of more complex pattern recognition algorithms, valuable to the interpretation and the processing community.

[1] Email: `bmanzani@ualberta.ca`

6.1 Introduction: The Analytic representation of signals

In the environment of pattern recognition and feature extraction there are many tools used to define an image according to its characteristics. Any digital image, either optical, acoustic or seismic is in the end a two dimensional signal. The two fundamental properties of one dimensional signals are its phase, a scalar from zero to 2π and amplitude (Oppenheim (1999)). For any signal expressed in more than one dimension, the phase is represented by a vector (Zang and Sommer (2007)). This motivated on which this report is based. In order to define the tools for the analysis of two dimensional signals, a first attempt is made to explain the foundations for a one dimensional signal. In the case of one dimensional signals, the split into even and odd parts is of essential importance. Let *f(t)* be a time signal that , after discretization is represented as a time series:

$$f\left[t\right] = a \in \mathbb{Q}, \quad \text{and} \quad \forall t \in \mathbb{Q}^{+} \tag{6.1}$$

This signal has a unique Fourier representation,. Its *n-th* term is:

$$F\left[\omega_n - \phi_n\right] = a_n(cos\left[\omega_n - \phi_n\right]) + (i)b_n(sin\left[\omega_n - \phi_n\right]) \quad | \quad a_n, b_n, \omega_n \text{ and } \phi_n \in \mathbb{Q}, \quad \text{and} \quad \forall n \in \mathbb{N} \tag{6.2}$$

The real part of this signal is said to be *even* in nature, since for any positive frequency there is always an identical counterpart on the negative half of the spectrum, as an innate characteristic of the *cosine* in its Fourier series expression. However, any signal whose spectrum is *odd*, will be purely imaginary, which accounts for the *sine* part. Therefore every negative frequency component is identical to its positive counterpart with a sign reversal or in terms of phase the same signal with a 90 degrees shift.

This accounts for a multiplication of the spectrum by the imaginary unit i(Taner, 1979). A signal whose odd part is obtained directly rotating the phase of its even part (conjugate harmonic) is called *analytic*. The analytic signal is of great value because it fulfills properties valid for real life signal processing problems across many different disciplines. The first one is that it satisfies the Laplace equation. The second one is that its derivatives are smooth and bounded at each point. Besides, this analytic properties are translated in our case into an identity split. This means that the analytic signal is separable into two fundamental parts: instantaneous amplitude (envelope) and instantaneous phase (Felsberg and Sommer (2001)). For a signal in the frequency domain represented as a complex function as F_H:

$$F_H = F(\omega) = A\left[\omega\right] \exp i\left[\pi(\omega_0 - \phi_n)n\right] \tag{6.3}$$

A simple way to obtain the conjugate of a signal is by flipping the the sign of the negative part of its spectrum (Bridge (2017)) and adding them together. The sum of the original signal and its conjugate is:

$$f_H(t) = \quad \mathfrak{F}^{-1}\{F(\omega) + i(sign(\omega))F(\omega)\} \tag{6.4}$$

There are two fundamental observations. Firstly, the odd part of the signal is roughly equal to the derivative of the even part, normalized by the magnitude of the frequency. In frequency domain:

$$i(sign(\omega))F(\omega) = \frac{i\omega}{|\omega|}F(\omega) = \frac{1}{|\omega|}\mathfrak{F}\{\frac{d}{dt}f(\omega)\} \tag{6.5}$$

$$f_H(t) = \quad \mathfrak{F}^{-1}\{F(\omega) + \frac{-i\omega}{|\omega|}F(\omega)\} \tag{6.6}$$

With time representations:

$$f_H(t) = \quad f(t) + f^*(t) \tag{6.7}$$

This fact provides a "differential" like operator without the high pass effect. Secondly, we have a complex number representation, which is in itself a split of identity, i.e. the signal may be expressed as a real instantaneous amplitude (or envelope) times a complex exponential whose argument is the frequency and the instantaneous phase (Felsberg & Sommer, 2001):

$$A(t) = \sqrt[2]{f^2 + (f^*)^2} \quad \phi(t) = atan(\frac{f^*}{f}) \tag{6.8}$$

These signal properties open the door for feature extraction tools like the instantaneous attributes (Chopra and Marfurt, 2005). Moreover, the instantaneous phase of the signal carries the majority of the structural information whereas the amplitude envelope can be seen as a measure of confidence

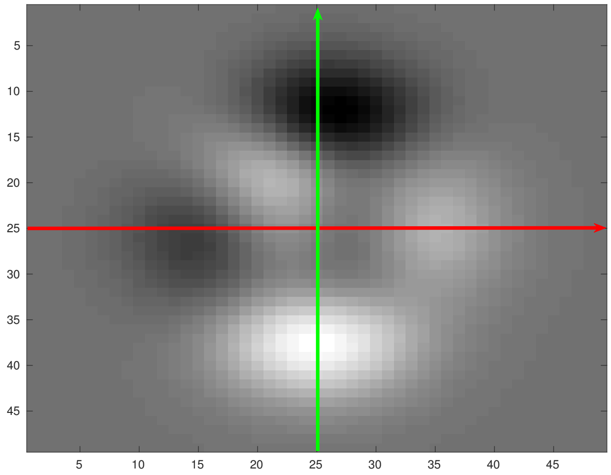


Figure 6.1: At every point there are two fundamental directions in which we can calculate the instantaneous phase, and this yields two different phase values at the same position, which is impossible.

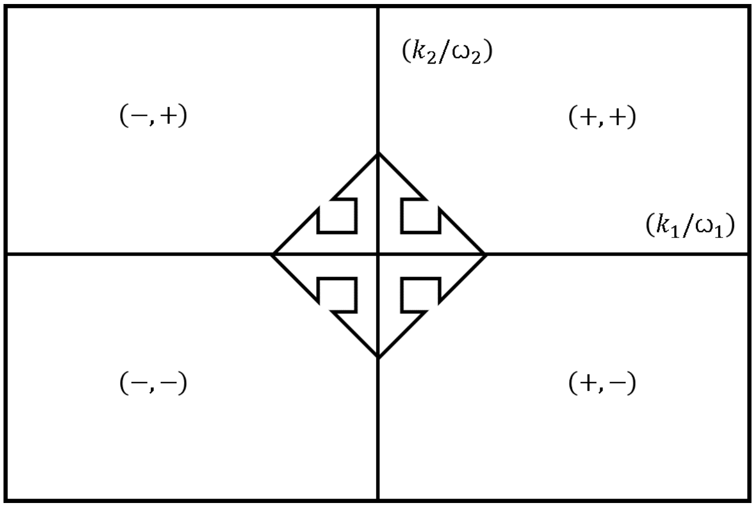


Figure 6.2: There are two directions in which we can analyze the symmetry in the Fourier domain: sign of component one changes as one traverses axis number two, and vice-versa.

on that given phase value at that given instant(Felsberg and Köthe (2005)). In fact, one can roughly reconstruct the structure of a signal, by using the phase alone (Oppenheim and Lim, 1981).

However, the Hilbert transform is a method only valid for one dimensional signals since the only difference introduced to produce the conjugate harmonic is a scalar phase rotation by one quarter of π radians. Now, let us imagine we can generate arbitrary one dimensional signals as slices in any possible propagation direction over an image or other signal in a higher dimensional space. Unfortunately, a major disadvantage in such a representation of an analytic signal (as 1D slices) is that the value of instantaneous phase and amplitude would be totally different just by changing the direction of the slice we take.

Hence, the possibility to use the Hilbert transform to obtain the instantaneous properties of the overall image by dividing it into one dimensional signals is eliminated, since any change in orientation would yield different results. In this article, I present a method to separate the *local* counterparts of the *instantaneous* phase and amplitude of two dimensional vector valued scalar signals i.e. images. The theory is already established in the corpus of specialized literature (Oppenheim and Lim (1981), Taner et al. (1979), Felsberg and Sommer (2001), Bridge (2017)).

6.1.1 Scale Space and self similarity

So far we have seen that the Riesz transform is a convolutional operator with properties similar to those of the derivative. At the beginning of the 1980's in the last century, some formalism about the scale in images were set forth (Witkin (1984), Koenderink (1984), Lindeberg and ter Haar Romeny (1994)). A key discovery was that any natural image sampled with a finite bandwidth (finite support) follows a Gaussian distribution in its samples and behaves as a smaller scale copy of itself when convolved with a Gaussian filter (see Figure 6.5). Therefore the convolution of an image with a Gaussian kernel does not alter the phase nor the position of local maxima and minima. This is in fact a preponderant interpretation of how the human visual system works, and thus a basis to develop pattern detection algorithms, which is why it has remained a popular topic in the computer vision and pattern recognition literature. By looking at an image at progressively farther and farther distances our vision fundamentally sees a smaller (decimated) and smoother version of the original. Here the only limit is that, at a certain point, the neighboring objects merge, successively. This situation is depicted in the figure(6.4), where a set of smoothed one dimensional signals is shown (top of the image). The curves on the bottom trace the positions of all peaks and troughs in the family of smooth signals. The ”n” shape patterns, eventually merge on the smoothest signal, but

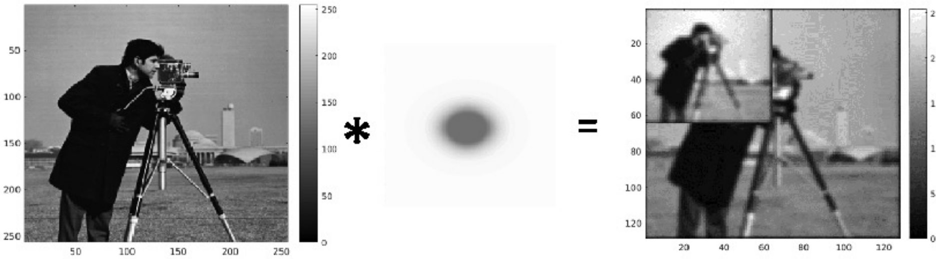


Figure 6.3: On the far left, a test image was convolved (symbolized by the star) with a Gaussian kernel(center). After convolution the local maxima and minima are still honored, as it is attested by the fact that the main features in the picture does not change even after decimation of the convolved image(far right).

one never sees the creation of any new peak nor trough. This is called a scaleogram (Lindeberg and ter Haar Romeny (1994)). This is remarkable since, in the case of higher dimensional signals the consistency in the location of local extrema is not trivial and depends highly on the kernel chosen (Lindeberg and ter Haar Romeny (1994)). An empirical explanation is that Gaussian kernels are zero phase, unitary filters (even functions). This is what the so called scale space concept entails, in which the iterative convolution of an image with a Gaussian kernel yields a set of images which although progressively sparser, still agree in the fundamental positions of local extrema. This images are called Pyramids since every new image is smoothed and decimated producing a set of images that mimics the levels of a "Pyramid" (Freeman et al. (1991)). Each level will have half of the number of columns and rows of the preceding one.

Borders, Edges and Derivatives: Steerable Filters

The following points should be considered the bottom line of the interpreter just as any other user of computer systems in which the user relies in his eyes to evaluate results of any kind (Marr and Vision (1982)). One of the fundamental operations in any pattern recognition technique is to be able to segment images, in order to separate features of interest. Our visual system detects only surfaces. Thus for visualization purposes, the only fundamental segmentation pattern we are able to recognize are either curves contained in surfaces or changes in light intensity (Marr and Vision (1982)). Any border may be perceived as a sudden change in amplitude with congruent phase for all components. If it is a line the local phase will be either $\pm\pi$ or $\pm\frac{\pi}{4}$ if it is the boundary of two surfaces (Kovesi (2003)). Hence, the first choice of an operator for edge and border detection is the gradient (Marr and Vision (1982)) under noise-free conditions . However we know that the gradient

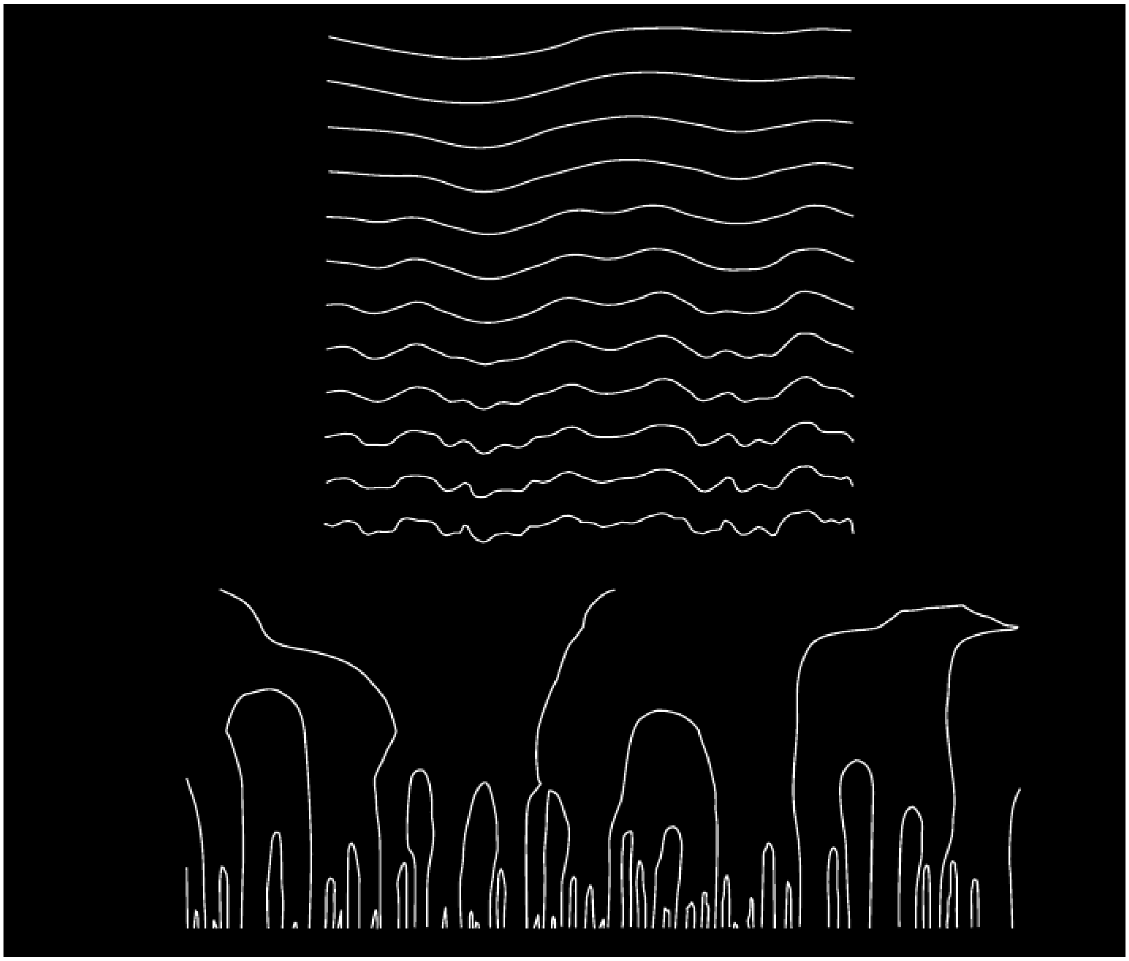


Figure 6.4: Top: A set of signals is generated. The recursive convolution of the Gaussian kernel with the image yields a smoother result every time. Bottom: The scaleogram. By tracking the positions of the peaks and troughs in the family of signals we can generate the "n" shaped plots on the bottom. This image was modified after Lindeberg and ter Haar Romeny (1994) and was originally published by Witkin (1984).

has a high-pass effect, that increases the amplitude of random noise, which is present in all real signals. This has rendered its direct application obsolete. However, we know that the convolution over complex valued and real functions is commutative, and that is equivalent to multiplication in frequency domain. Functions like the Gaussian kernel, Poisson (Felsberg and Köthe (2005)), Gabor filters and their logarithms (Bridge (2017)), among others, are convolutional filters with zero phase and unitary amplitude. Similarly, differential operators such as finite differences may be applied as convolutional masks. Hence, convolving the signal with the directional derivative of a smoothing kernel in time, or multiplying them in frequency domain is equivalent to obtaining the directional derivatives of a signal and then smoothing them. This methodology is similar to the structure tensor (Zang and Sommer (2007)). Moreover, these operators are suitable for the construction of pyramids of directional derivatives, provided that local maxima and minima in the derivative are congruent for each smoothing level.

The aforementioned strategy, has given place to extremely fast algorithms for feature and pattern recognition with convolutional kernels, with several applications in diverse fields including seismic interpretation and pattern recognition (Simoncelli and Freeman (1995)). In the Figure (6.5), I show a smoothed directional derivative image produced with the method of Gaussian pyramids produced with the Gaussian directional derivative operator.

A second application of the pyramid methodology to detect edges in images is the so-called Laplacian pyramid. To construct it, we generate a set of images product of recursive smoothing. However, in this case, unlike in the Gaussian pyramid, each smoothed image is first subtracted from its parent image, prior to decimation. This to decrease amplitude variations within the borders while at the same time emphasizing the borders themselves. Nevertheless, a draw back in this methods is that the integrity of the phase is not guaranteed. Given that we are subtracting two similar images but with different levels of details, the smallest wavelengths are smeared if the smoothing difference is too large. In Figure 6.6. I present an example where I produce a two level Laplacian pyramid. If the level of smoothing is not chosen close enough to the parent image the non stationarity of the wavelet is apparent given that by following the contour of the cameraman in the image, one can see a gentle variation in gray tone parallel to the coat, tripod and legs. A second challenge in the use of the Laplacian pyramid as a tool for edge and border detection is that of noise levels and level of detail in the identification process. For instance a tool to pick the borders once emphasized with in the Laplacian pyramid levels could the second central differences applied as a convolutional mask (6.7). If we again resort to direct differentiation techniques, the signal to noise ratio falls significantly.

To summarize this section, there are two main challenges in the use of pyramids as steerable filters. Firstly, the filters are not perfectly orthogonal, in other words you have to use more than the bare

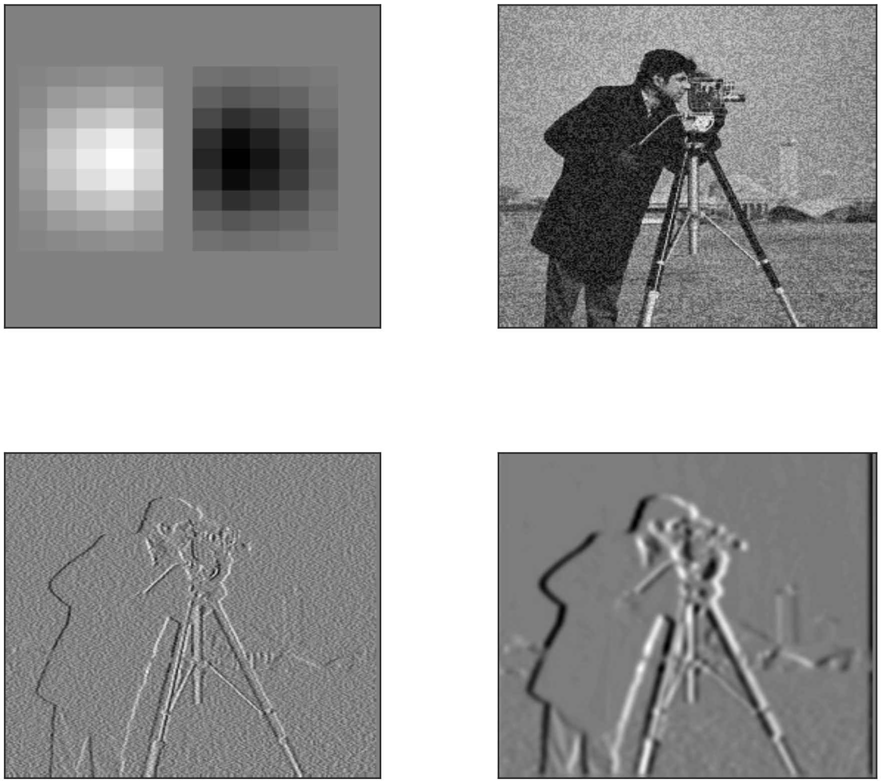


Figure 6.5: Top left: the derivative of a Gaussian kernel is equivalent to smoothing plus derivation. Top right: The target image contains random noise. Bottom left: due to its high pass effect, after performing conventional finite differences on the image the noise is increased. Bottom right: the direct convolution of the derivative of the Gaussian kernel with the image yields a smoother result, suitable for edge detection.

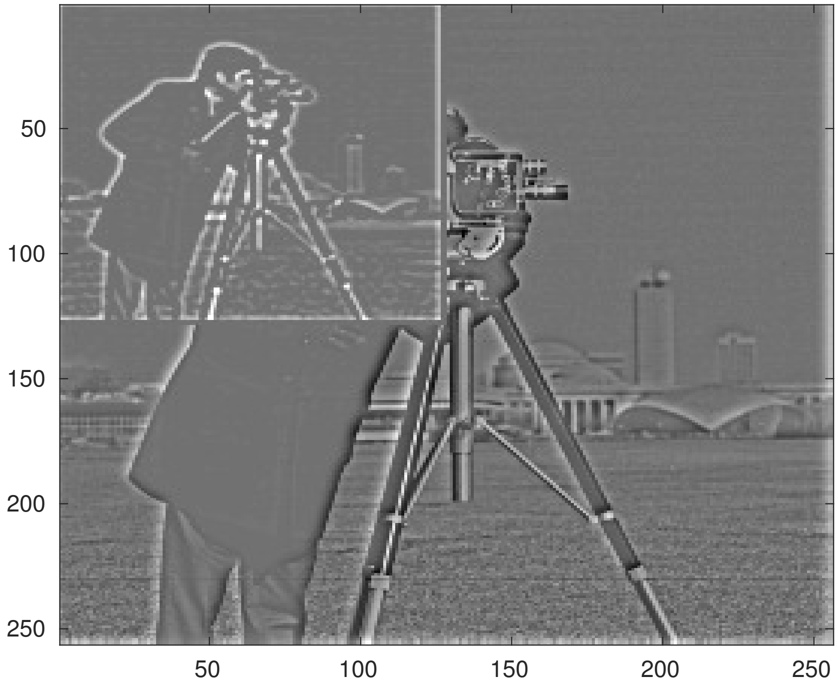


Figure 6.6: To construct a Laplacian pyramid after convolution with a Gaussian kernel each image is subtracted from its parent, this removes the smoothest trends inside the areas bounded by all edges.The main features of the image are again honored, but a phase change is introduced.

minimum number of filters. Thus, pyramids are said to overcomplete (Freeman and Adelason, 1995). This poses a problem as well for the generation of equivalent analytic signals since the instantaneous attributes are a result of a basis of elementary filters, that are independent from each other. Secondly as they are not an analytic representation, they are not suitable to the generation of an analytic signal and all the attributes of interest that this kind of representation would possess.

6.2 Monogenic Signal

6.2.1 Multidimensional analytic signal

In this section I introduce the extension of the Analytic signal concepts for images, and propose the feasibility of an extension to volumes and other dimensions. As starting point, it is important to

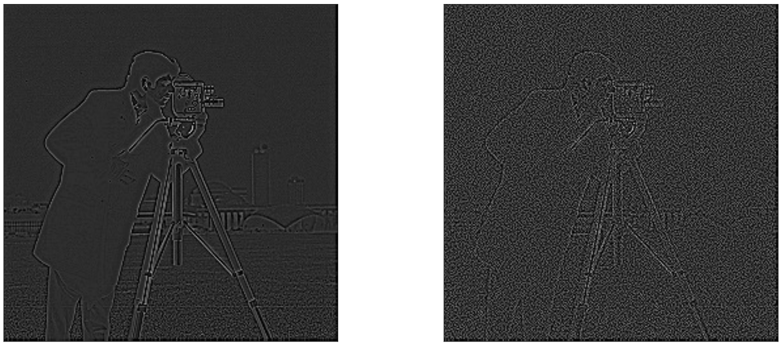


Figure 6.7: One of the challenges in feature extraction images is the noise. In this example, the second difference was calculated on the same image without noise (left) and with 10% of white noise.

stress the profound difference between local and instantaneous properties. The word instantaneous entails a temporal reference or, relaxing the term, a point in a line -e.g. the line of time. For two dimensional signals phase and amplitude are considered local because they hold a constant value for a given point, regardless of the direction in which they are calculated -i.e. rotation invariance. Moreover, since the phase is roughly a measure of structure, or changes, it is invariant to changes in the overall value of amplitude. We can think of the image as a set of points. In turn, every point in the image is the intersection between any given number of lines in any possible direction projected on the image to analyze. In order for a signal to be a Monogenic function (and thus to be able to extract its local properties) it has to satisfy that the total derivative (i.e. the sum of its directional derivatives) is constant upon any possible translation:

$$\Phi_H(x_1,y_1) = \Phi_H(x_2,y_2), \quad A_H(x_1,y_1) = A_H(x_2,y_2) \tag{6.9}$$

or rotation and scaling

$$F_H(\rho_1,\theta_1) = F_H(\rho_2,\theta_2), \quad \forall\rho_1 = s(\rho_2) \quad \text{and} \quad \theta_1 = \theta_2 + \alpha \tag{6.10}$$

In a similar manner, scaling of the amplitude by a constant factor should never affect the phase of the signal.

6.2.2 The Riesz Transform

For an image in the Fourier domain, there are two frequency or wavenumber components (κ/ω) and therefore there are two directions in which sign can change (see Figure 6.8). Therefore we can think that an analytic signal taking arguments in $\mathbb{R}^n$ is equivalent to finding n conjugate harmonics by applying the Riesz transform to the image n times in n directions (Zang and Sommer (2007), Bridge (2017)). The Riesz transform is an analogous to the one dimensional Hilbert transform, as a

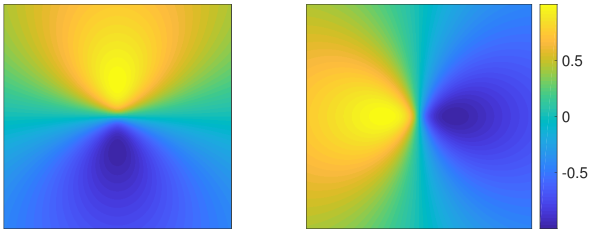


Figure 6.8: Riesz operators in Frequency domain. Note the polarity reversal across the horizontal (left) and vertical axis (right). Even though the $\frac{1}{|\vec{\omega}|}$ term damps the amplitude as we move away form the origin the filters are still different from zero at the boundary.

convolutional kernel transform extended to $\mathbb{R}^n$. At this point we must recall that the conjugate harmonics of a signal have roughly the shape of a differential operator. Thus we may loosely express the analytic signal as:

$$\mathfrak{F}(\mathfrak{R}_n f)(x) = -i\frac{\omega_j}{|\vec{\omega}|}(\mathfrak{F}f)(x)\frac{1}{|\vec{\omega}|}\mathfrak{F}\{\frac{\partial}{\partial x_j}f(t)\} \tag{6.11}$$

where the frequency (or wavenumber) vector in $\mathbb{R}^n$ is:

$$\vec{\omega} = \{\omega_j | j \in (1, n); n \in \mathbb{N}\} \tag{6.12}$$

However although noise is partially controlled by the $\frac{1}{|\vec{\omega}|}$ term, we can see in Figure 6.9 that the Riesz operator in Frequency domain is not zero at the boundary (it is in fact asymptotic), and thus in noisy data this would introduce artifacts, upon the application of the filter masks. Thus, a set of filters for smoothing and derivatives is desirable to tapper off the highest frequencies at the boundaries. For our application we chose log-Gabor filters (Fischer et al. (2009)). These filters are defined as:

$$G(\omega) = \frac{\omega_0}{\omega}\exp\left(\frac{-\left(\log(\omega/\omega_0)\right)^2}{2\left(\log(\sigma/\omega_0)\right)^2}\right) \tag{6.13}$$

Where ω is a circular adimensional frequency that takes values from 0 to π. The scale or bandwidth of the filter band is measured with the variance of the function normalized by the frequencyσ.

Figure 6.9: Riesz operators in Frequency domain. Note the polarity reversal across the horizontal (left) and vertical axis (right). Even though the $\frac{1}{|\vec{\omega}|}$ term damps the amplitude as we move away form the origin the filters are still different from zero at the boundary.

6.3 Monogenic Scale Space and curvature Tensor

By combining the concepts of Scale-Space and Riesz Transform one can construct "Pyramids" of analytic signals and their attributes. This opens the door to a tremendous number of useful applications in which there is not just a means to approximately describe borders and edges, but to actually obtain curvature and more detailed geometry parameters (Felsberg and Sommer (2004)). This concept was coined in Felsberg (2004), and named as *Monogenic Scale-Space*. Figure (6.10) is again used and the border detected with the local amplitude of the curvature tensor.

6.3.1 Application of the Monogenic signal to border and edge detection

In this section some results in the application of the local amplitude, phase, orientation to detect pattern are shown. The first set of images are a similar test image (figure 6.10) as the one presented by Zang and Sommer (2007). More examples pertinent to seismic interpretation will be presented below. The current test image is just a *portable network graphics* file (*png*), with binary dimensions and amplitude (eight bits). The first approach was to produce the local amplitude (6.11), local

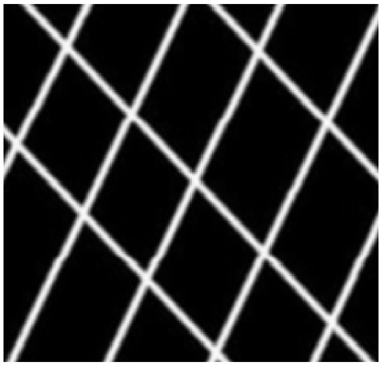


Figure 6.10: Raw test image. The test image is a grid like pattern with an amplitude of 0.5 (127) relative to the maximum possible for 8 bit images (255) against a black background (zero amplitude). Modified after Di-Zang, Sommer, 2007.

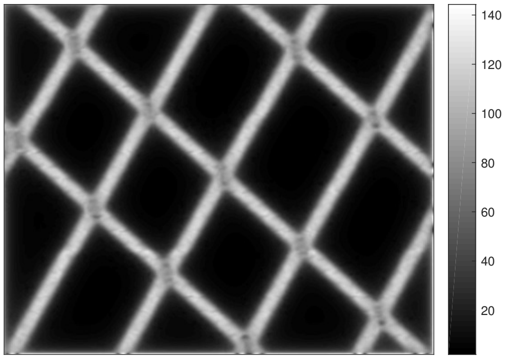


Figure 6.11: Local Amplitude of the test image. Modified after Di-Zang, Sommer, 2007.

phase (6.12), and an orientation filter (6.13). The local orientation is a key advantage of the multidimensional analytic signal over its standard one dimensional counterpart. Given the fact that we have now two directions to calculate the odd parts of the signal, we can estimate the ratio of change in horizontal and vertical directions (horizontal and vertical Riesz operators), to estimate the orientation or dip angle of borders and edges.

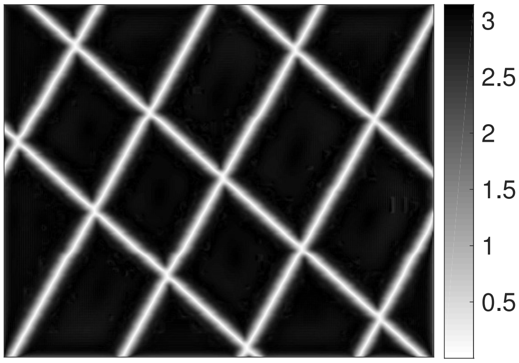


Figure 6.12: Local phase of the test image. Angles go from 0 to π.Modified after Di-Zang, Sommer, 2007.

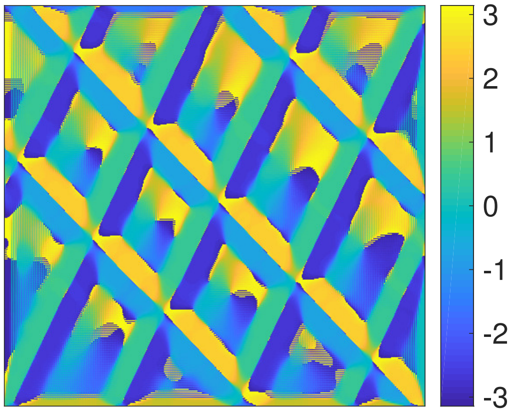


Figure 6.13: Local orientation of the borders of the test image. Angles go from $-\pi$ to π. Modified after Di-Zang, Sommer, 2007.

Curvature: calculation of the curvature tensor components

From the images we can see that amplitude responds quite well with the pattern however as it is pointed out in Zang and Sommer (2007), the detection of corners is not identifiable by means of a two dimensional analytic signal. Thus we now need to extend the method to be able to detect curvature. For our purposes, curvature in an image means either a ”lump” or ”depression” in the local amplitude, in the case of pictures this is light intensity and for seismic images, this may be reflectivity (if calculated), or local amplitude of the seismic signal. A second type of curvature is when there is a constant amplitude but a sharp turn in a border. Curvature is thus, the fundamental the purpose to test these filters in the grid pattern shown. This is key in the identification of seismic targets as these are rarely bounded by straight lines. In order to achieve this a new tool is needed. This tool is called the curvature tensor, which is the analytic signal counterpart of the structure tensor (Zang and Sommer (2007)). The Curvature Tensor is the result of the application of the Riesz transform to a set of angular filters similar to those used in the steerable pyramids (see Simoncelli and Freeman (1995)). The filters are called spherical harmonics and they are basically angular weighting coefficients, thus adding more possible combinations to calculate angles in the test images. These set of filters have the core characteristic of reacting to corners (Zang and Sommer (2007)). Thus we can now calculate, the so called two dimensional local amplitude (figure 6.14), phase (figure 6.15) and the mean curvature response (corner detector). The amplitude maxima and minima are presented in figure (6.16) and the corner detector in figure (6.17)

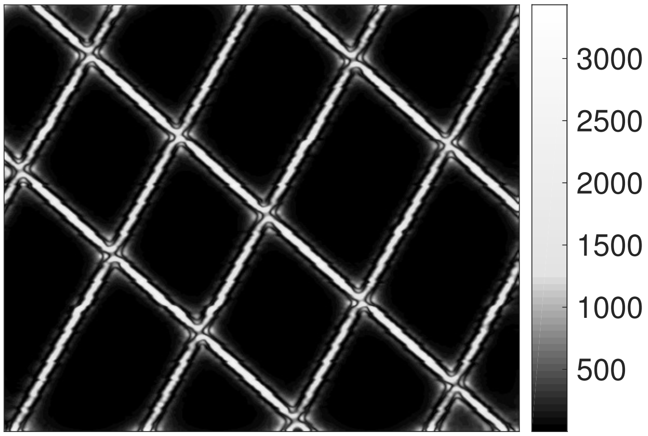


Figure 6.14: Two dimensional local Amplitude of the test image. Notice the reaction to the intersections in the pattern. Unlike the one dimensional local amplitude, the two dimensional local amplitude responds more accurately in the corners. Input test image modified after Di-Zang, Sommer, 2007.

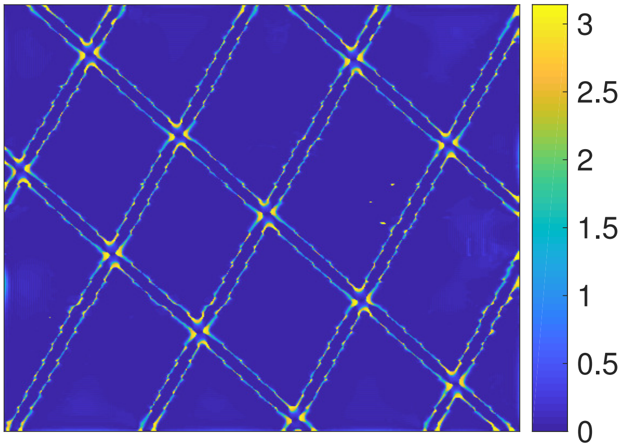


Figure 6.15: Two dimensional local phase of the test image. Notice the reaction to the intersections in the pattern. Angles go from 0 to π. Input test image modified after Di-Zang, Sommer, 2007.

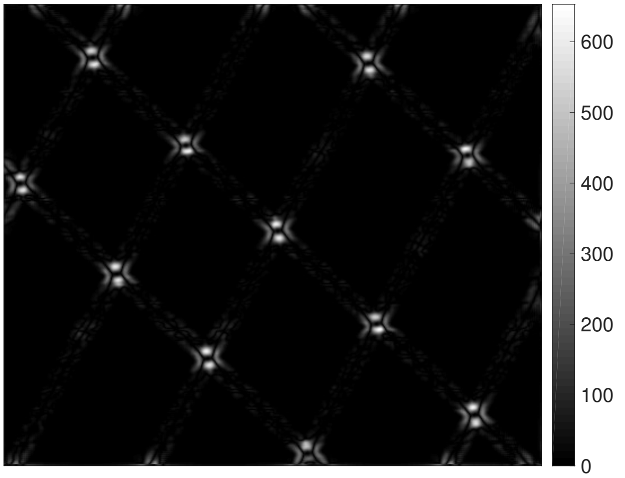


Figure 6.16: Two dimensional amplitude maxima mapped by the Gaussian curvature -which is the second total derivative. The image has local maxima at the intersections. Input test image modified after Di-Zang, Sommer, 2007.

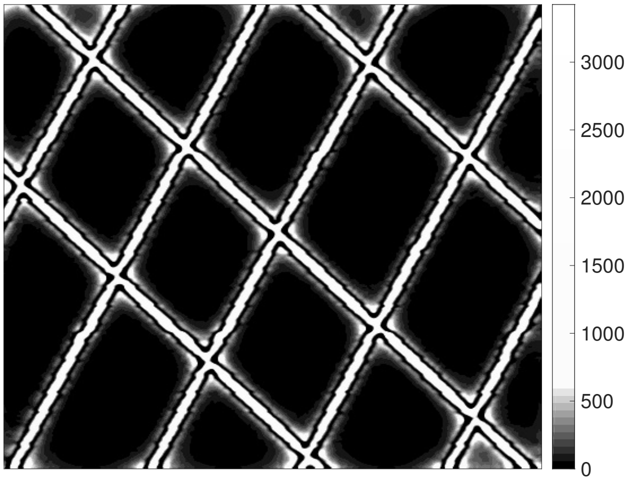


Figure 6.17: Local response of corners (the pixel size in the image). Input test image modified after Di-Zang, Sommer, 2007.

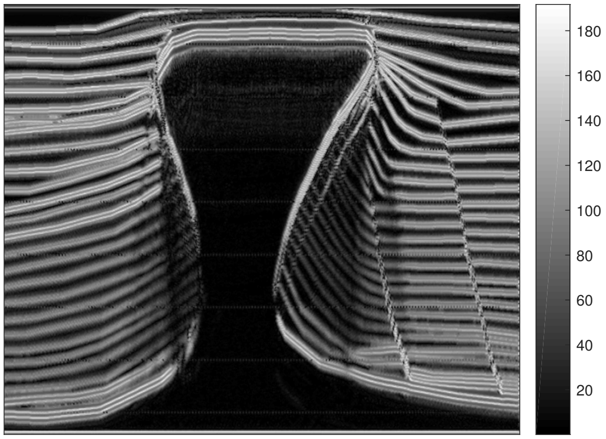


Figure 6.18: Two dimensional local Amplitude of a png image of a salt dome model synthetic seismic data.

6.3.2 Application of the multidimensional analytic signal to synthetic data

The following are examples of the same attribute presented before, applied to a *png* version image of a synthetic salt model. This is doable since the analytic signal takes care of the pixel amplitude distribution and it is insensitive to the physical phenomena that may produce such amplitudes. The image preset two characteristics that important to mention. First, since the format works with eight bits amplitude there 2^8 possible values as compared to the $\pm 2^{3}2$ in a four byte conventional trace. This means that we loose details in the image. However given the fact that we low pass first the images to generate smooth versions, we prevent the formation of aliasing Moire patterns. The image was taken from: http://www.seismiccity.com/Services.html.

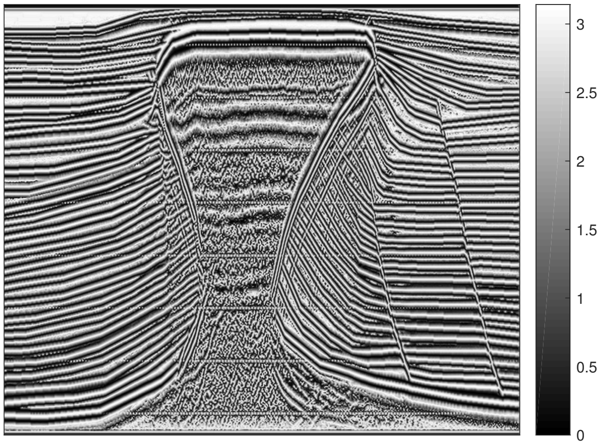


Figure 6.19: Two dimensional local phase of a png image of a salt dome model synthetic seismic data. Angles go from 0 to π.

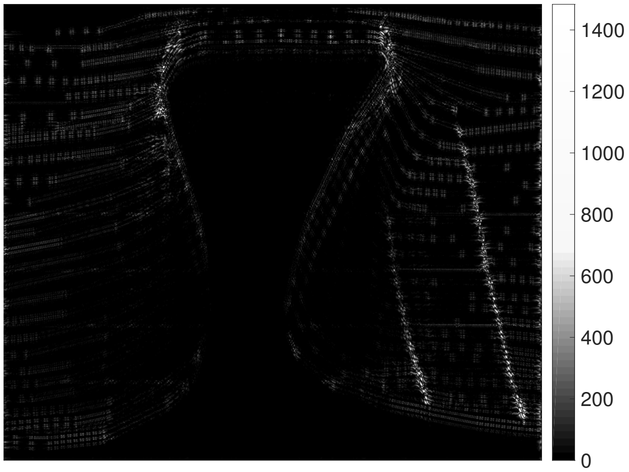


Figure 6.20: Local response of mean curvature or corners.

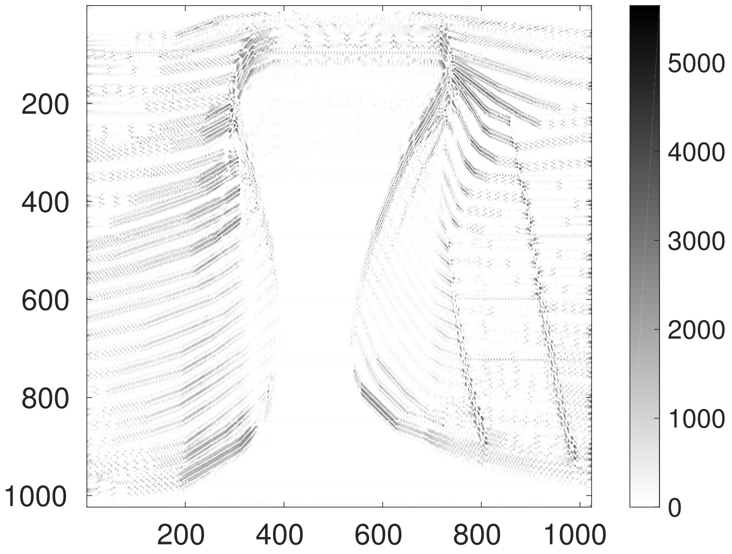


Figure 6.21: Two dimensional amplitude minima mapped by the Gaussian curvature (values > 0).

6.4 Application of the multidimensional analytic signal to real Seismic Data: John Lake.

The following examples are a brief exercise to evaluate how suitable the multidimensional analytic signal is to identify channels. There are two sets of images: time slices for which I present one dimensional local attributes and curvature tensor attributes. The channel is already quite visible in the raw data (6.24). However the response of the curvature tensor attributes, and in particular, the one dimensional phase, allows to convey information about the channel that can be, in principle measurable. The phase is invariant to amplitude changes in the time slice, condition that makes it an ideal detector for structures in data where amplitude change are abrupt. The central frequency chosen was $\pi/2$ and the standard deviation was $\sigma = .2$.

The second most efficient attributes in picking the channel features are the corner detector (6.27) and the Gaussian curvature or local maxima and minima detector (6.28).

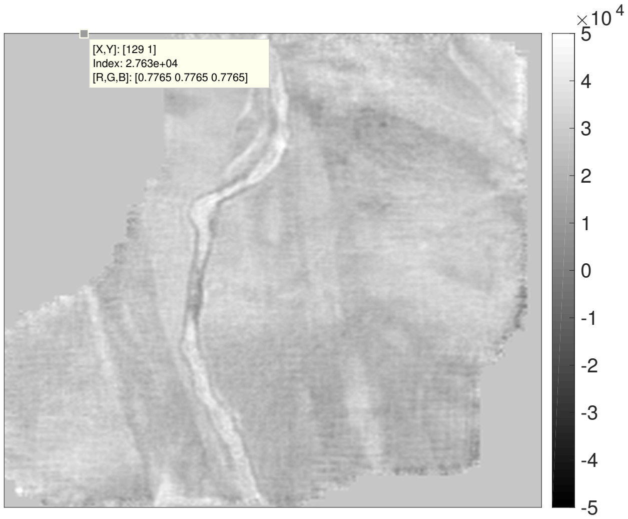


Figure 6.22: Time slice of John Lake dataset showing a channel. The presence of the channel is evident in the data, however at this stage we show that, in principle it is possible to detect features of interest in quantifiable fashion that can be used by machine learning algorithms.

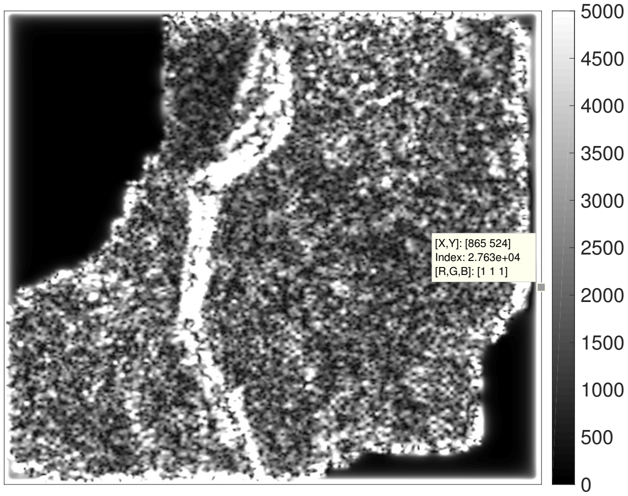


Figure 6.23: One dimensional local Amplitude of a time slice showing a channel.

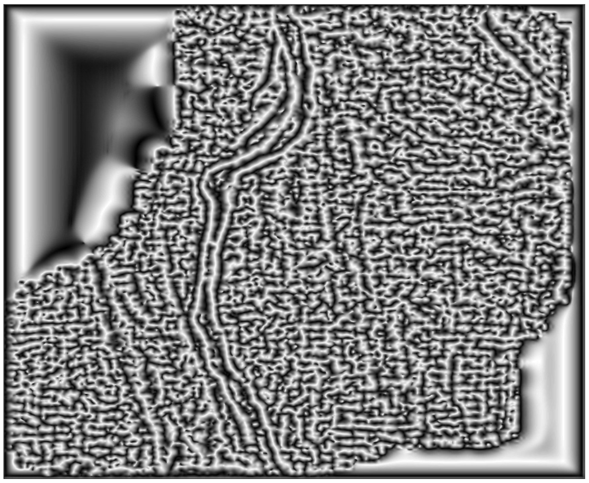


Figure 6.24: One dimensional local phase. Angles go from 0 to π.

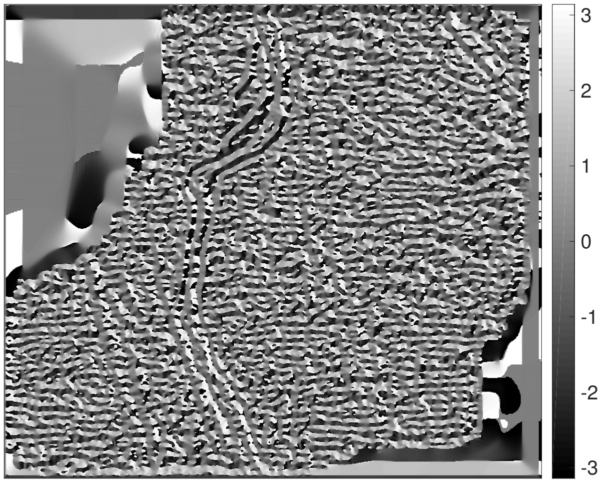


Figure 6.25: One dimensional local orientation. Angles go from $-\pi$ to π.

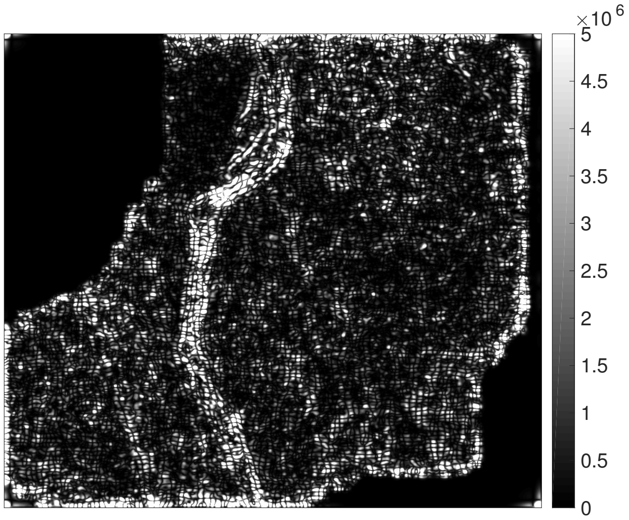


Figure 6.26: Two dimensional amplitude.

6.5 Conclusions and the road ahead

The multidimensional analytic signal is a useful tool to detect features in images, however its use needs accurate determination of parameters in order to produce results that are useful for the detection of features of interest to the interpreter. It is shown that, in principle it is possible to detect features of interest in a quantifiable fashion that can be used by machine learning algorithms (angles, envelope amplitudes, curvature). Since the local attributes are insensitive to the phenomena that produced the images, this makes them ideal for robust and fast applications, but its use requires a good working knowledge of parameter testing in order to avoid biasing the interpretation. The curvature tensor provides better resolution to non-straight boundaries and amplitudes maxima or minima. it allows to detect patterns that might of interest to map reservoir and delimit potential drilling targets. The response of the curvature attributes to faults and contacts seems to be promising in the case of the synthetic salt dome model. However this model is just an 8 bit graphic file with very low noise content. The current state of development of this work requires further research in parameter selection, particularly to interpret the response of the phase and orientation of the curvature tensor. This point will require the selection of a methodology for unwrapping the phase, as in the current state of development this is still represented with wrapped color scales. From the point of view of pattern detection this work seems a good starting point to develop a deeper understanding and potentially connect this methodology with other like the steerable pyramids.

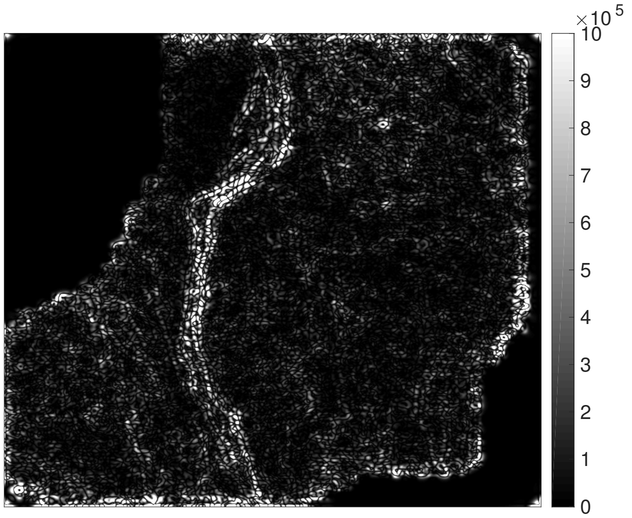


Figure 6.27: Local response of corners (the pixel size in the image). Angles go from $-\pi$ to π.

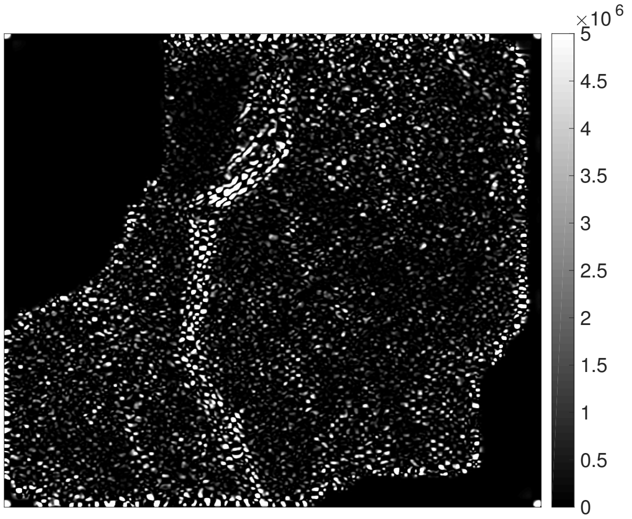


Figure 6.28: Two dimensional amplitude minima mapped by the positive part of the Gaussian curvature -which is the second total derivative. The image has local maxima at the intersections.

6.6 Appendix: attributes table

Local Attributes Table		
Attribute Name	Values	Responds to
Amplitude	$[0, \infty)$	Total power of original signal and its conjugate harmonics
Phase	$[0, \pi]$	Responds to structure in image
Orientation	$[\pi, \pi]$	Indicates the angular orientation of the structure (in this work)
2D Amplitude	$[0, \infty)$	Responds both to borders and corners
2D Phase	$[0, \pi]$	Gives the angle of the normal vector to any border requires unwrapping
2D Orientation	$[\pi, \pi]$	Indicates the angle of a vector parallel to the borders.
Mean curvature or corner detector	$(-\infty, \infty)$	Responds both corners i.e. sudden orientation changes in edges.
Gaussian curvature or local extrema detector	$(-\infty, \infty)$	Responds to local maxima/minima in amplitude.

References

Bridge, C. P., 2017, Introduction to the monogenic signal: arXiv preprint arXiv:1703.09199.

Felsberg, M., and U. Köthe, 2005, Get: The connection between monogenic scale-space and gaussian derivatives.: Scale-Space, Springer, 192–203.

Felsberg, M., and G. Sommer, 2001, The monogenic signal: IEEE Transactions on Signal Processing, **49**, 3136–3144.

———, 2004, The monogenic scale-space: A unifying approach to phase-based image processing in scale-space: Journal of Mathematical Imaging and vision, **21**, 5–26.

Fischer, S., R. Redondo, and G. Cristobal, 2009, How to construct log-gabor filters: Open Access Digital CSIC Document, **21**.

Freeman, W. T., E. H. Adelson, et al., 1991, The design and use of steerable filters: IEEE Transactions on Pattern analysis and machine intelligence, **13**, 891–906.

Koenderink, J. J., 1984, The structure of images: Biological cybernetics, **50**, 363–370.

Kovesi, P., 2003, Phase congruency detects corners and edges: Presented at the The australian pattern recognition society conference: DICTA 2003.

Lindeberg, T., and B. M. ter Haar Romeny, 1994, Linear scale-space i: Basic theory, *in* Geometry-Driven Diffusion in Computer Vision: Springer, 1–38.

Marr, D., and A. Vision, 1982, A computational investigation into the human representation and processing of visual information: WH San Francisco: Freeman and Company, **1**.

Oppenheim, A. V., 1999, Discrete-time signal processing: Pearson Education India.

Oppenheim, A. V., and J. S. Lim, 1981, The importance of phase in signals: Proceedings of the IEEE, **69**, 529–541.

Simoncelli, E. P., and W. T. Freeman, 1995, The steerable pyramid: A flexible architecture for multi-scale derivative computation: Image Processing, 1995. Proceedings., International Conference on, IEEE, 444–447.

Taner, M. T., F. Koehler, and R. Sheriff, 1979, Complex seismic trace analysis: Geophysics, **44**, 1041–1063.

Witkin, A., 1984, Scale-space filtering: A new approach to multi-scale description: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84., IEEE, 150–153.

Zang, D., and G. Sommer, 2007, Signal modeling for two-dimensional image structures: Journal of Visual Communication and Image Representation, **18**, 81–99.

Computational efficient multi-dimensional Singular Spectrum Analysis

J. Cheng[1] and M. D. Sacchi

Abstract

We present a computational efficient multi-dimensional Singular Spectrum Analysis method for the recovery and de-noising of multi-dimensional seismic data. Compared to the other implementations of Singular Spectrum Analysis method, the proposed algorithm does not require building multi-level block Hankel trajectory matrices. The key is to replace the singular value decomposition of a multi-level block Hankel matrix by the randomized QR decomposition. We also present a new strategy in which anti-diagonal averaging of the multi-level block Hankel matrix is efficiently computed via convolution. The new algorithm significantly decreases the computational cost and memory requirement of Singular Spectrum Analysis data recovery. We test the effectiveness of the method through synthetic and real data examples.

7.1 Introduction

The enhancement of signal-to-noise ratio of seismic records and the interpolation of seismic data are important subjects in the field of seismic data processing. Different techniques have been utilized to de-noise seismic traces and to recover seismic traces at unsampled spatial positions. Many of

[1] Email: `jinkun@ualberta.ca`

these techniques rely on mathematical transforms, such as the Fourier transform (Duijndam et al., 1999; Liu and Sacchi, 2004; Xu et al., 2005), Radon transform (Kabir and Verschuur, 1995; Trad et al., 2002) and Curvelet transform (Herrmann and Hennenfent, 2008; Hennenfent et al., 2010), that convert the data to a different domain. Noise attenuation and interpolation can also be carried out via the spatial prediction filters (Canales, 1984; Spitz, 1991) and the projection filters (Soubaras, 1994) in the frequency-space domain.

In recent years, a new family of de-noising and interpolation methods based on rank reduction has been proposed. The rank reduction methods assume that the ideal well-sampled, noise-free data can be represented by a low-rank matrix/tensor (Trickett, 2003; Kreimer and Sacchi, 2012). De-noising and reconstruction can be implemented by finding a low-rank structured data that honors the noisy and incomplete observation. This article interests in rank-reduction techniques based on Cadzow filtering, also called Singular Spectrum Analysis (Sacchi, 2009; Trickett and Burroughs, 2009; Trickett et al., 2010). Cadzow filtering and Singular Spectrum Analysis denote equivalent methodologies that arise from different research areas. Researchers usually adopt Cadzow filtering as a tool for de-noising in signal processing (Cadzow, 1988) whereas Singular Spectrum Analysis has been proposed for analyses of time series and dynamic systems (Broomhead and King, 1986). We will adopt the name Singular Spectrum Analysis (SSA) in this article following (Sacchi, 2009). It is important to note that SSA has been extended to Multi-dimensional Singular Spectrum Analysis (MSSA) for multi-dimensional seismic data de-noising and reconstruction (Oropeza and Sacchi, 2011; Gao et al., 2013). MSSA for seismic data de-noising and reconstruction is based on the assumption that in frequency-space domain, each constant frequency slice of seismic data can be embedded into a low-rank Hankel matrix. If the data consists K dipping events, the rank of the associated Hankel matrix equals K. Incoherent noise and missing observations will increase the rank and therefore, rank reduction is an effective tool for de-noising and interpolation (Oropeza and Sacchi, 2011). Rank reduction is commonly applied via the Singular Value Decomposition (SVD) (Golub and van Loan, 1996) and the filtered data are acquired by averaging the anti-diagonal elements of the reduced-rank Hankel matrix.

Despite the efficacy of MSSA, its computational cost has always been a concern because rank reduction is usually implemented via the classic truncated SVD. Efforts have been made by replacing the SVD by more efficient algorithms, such as randomized singular value decomposition (Oropeza and Sacchi, 2011) and Lanczos Bidiagonalization(Gao et al., 2013). It is important to note that Gao et al. (2013) utilized a fast Hankel matrix-vector product in a Lanczos Bidiagonalization algorithm to avoid building Hankel matrices. The method is very efficient in calculating the low-rank approximation of the Hankel matrices. However, large Hankel matrices are formed in the final anti-diagonal averaging stage in order to recover the filtered data.

We present a computational efficient implementation for MSSA that does not require building Hankel matrices. To improve the computational performance, firstly, we adopt the randomized QR decomposition (RQRD) (Chiron et al., 2014; Cheng and Sacchi, 2016) for acquiring the low-rank approximation of the Hankel Matrices. Secondly, following Gao et al. (2013), we embed the Hankel matrices into circulant matrices and compute the Hankel matrix-vector multiplication via Fourier transform. Finally, we propose to compute the anti-diagonal averaging of Hankel matrices via convolution (Korobeynikov, 2010). The latter can also be computed efficiently via Fourier transform. As a result, the construction of the Hankel matrices is avoided in MSSA and therefore the proposed method is computationally efficient. We adopt synthetic and real data examples to test the performance of the proposed algorithm.

7.2 Computational efficient SSA

7.2.1 The complexity of SSA

We first provide a brief review of the SSA methodology and analyze the computational complexity for the method. We begin our discussion by examining 2D seismic data $D(\omega, x)$ in the frequency-space domain. We assume the data are regularly sampled along the spatial dimension. At a given monochromatic frequency ω_0, the frequency slice can be denoted as $D(\omega_0, x) = [D_1, D_2, \cdots, D_N]^T$, where N is the total number of traces. SSA of seismic data entails the following three steps:

- We first embed $D(\omega_0, x)$ into a Hankel structured trajectory matrix $\mathbf{H}$ as follows

$$\mathbf{H} = \begin{bmatrix} D_1 & D_2 & \cdots & D_M \\ D_2 & D_3 & \cdots & D_{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ D_L & D_{L+1} & \cdots & D_N \end{bmatrix}, \tag{7.1}$$

  where $L + M - 1 = N$ and $L \leq M$. A good strategy is to choose L such that the Toeplitz matrix is approximately square. In other words, we let $L = \left[\frac{N}{2}\right] + 1$. Forming the Hankel structured matrix is computational efficient but this step requires the storage of an $M \times L$ matrix.

- The second step of SSA entails finding a low-rank approximation of the Hankel trajectory matrix. This is usually done by the truncated SVD as follows

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathrm{SVD}[\mathbf{H}]\,, \tag{7.2}$$

  where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{S}$ is a diagonal matrix. Note that this step can be extremely expensive for large matrices. The computational complexity of the singular value decomposition is approximately $\mathscr{O}(L^2M + LM + M^3)$ (Golub and van Loan, 1996). A new set of singular values $\hat{\mathbf{S}}$ are computed via

$$\begin{aligned} \hat{\mathbf{S}}_{l,l} &= \mathbf{S}_{l,l} \quad & l \leq K \\ \hat{\mathbf{S}}_{l,l} &= 0 \quad & l > K \end{aligned} \tag{7.3}$$

  The low-rank approximation of the Hankel matrix is then computed via

$$\hat{\mathbf{H}} = \mathbf{U}\hat{\mathbf{S}}\mathbf{V}. \tag{7.4}$$

  The regrouping of the Hankel matrix also yield a complexity $\mathscr{O}(L^2K)$.

- In the last step, anti-diagonal averaging of the reduced-rank matrix is applied to recover the filtered signal. In other words, filtered data can be recovered via

$$\hat{D}_i = \begin{cases} \frac{1}{i}\sum_{j=1}^{i} \hat{H}_{j,\,i-j+1} & 1 \leq i \leq M\,, \\ \frac{1}{M}\sum_{j=1}^{M} \hat{H}_{j,\,i-j+1}, & M \leq i \leq L\,, \\ \frac{1}{N-i+1}\sum_{j=i-L+1}^{M} \hat{H}_{j,\,i-j+1}, & L \leq i \leq N\,, \end{cases} \tag{7.5}$$

  where i denotes the trace indices. Anti-diagonal averaging requires $\mathscr{O}(N)$ multiplications and $\mathscr{O}(ML)$ sums (Korobeynikov, 2010).

Randomized QR decomposition

We propose to use a randomized QR decomposition as an alternative to the SVD. Instead of applying SVD to the Hankel matrix, a random projection is first performed

$$\mathbf{M} = \mathbf{H}\Omega, \tag{7.6}$$

where Ω denotes a random set that is composed by k independent vectors ($k \ll M$). The random projection shrinks the size of Hankel matrix ($L \times M$) to a much smaller matrix ($L \times k$) while keeping as much variability as possible.

The random projection can be computed using a fast Hankel matrix-vector product. Therefore building Hankel trajectory matrix can be avoided when performing matrix rank reduction. The idea is to embed the Hankel matrix into a circulant matrix and then use fast Fourier transform to compute matrix times vector multiplications (O'Leary and Simmons, 1981). We refer the readers to Gao et al. (2013), where the authors discussed in details a fast Toeplitz matrix-vector product. A Hankel matrix can be easily converted to a Toeplitz matrix by reversing the columns. We conveniently adopt the algorithm in Gao et al. (2013) given the following relation

$$\mathbf{y} = \mathbf{H}\mathbf{x} = \mathbf{T}\mathbf{z}, \tag{7.7}$$

where $\mathbf{T}$ denotes a Toeplitz matrix and $\mathbf{x}$ is a vector (Korobeynikov, 2010). $\mathbf{z}$ is acquired by reversing the order of the entries of $\mathbf{x}$. $\mathbf{y}$ is the resulting vector of the multiplication. As is discussed in Gao et al. (2013), no Hankel or Toeplitz matrix is required. Therefore, fast matrix-vector multiplication, is a key step to avoid building the Hankel trajectory matrix in the SSA algorithm. Fast Hankel matrix-vector product is summarized in Algorithm 7.3, where $\circ$ denotes the Hadamard (element-wise) product.

Algorithm 7.1 Fast Hankel matrix-vector product

```
Inputs:
    Seismic data at a given frequency: $\mathbf{D}$; Vector $\mathbf{x}$; Size of Hankel matrix: $L$ and
    $M$
Output:
    Vector after multiplication: $\mathbf{y}$
$\mathbf{c} = [D_M, D_{M+1}, \cdots, D_N, D_1, D_2, \cdots, D_{M-1}]^T$   (first row of circulant matrix)
$\mathbf{z} = [x_M, x_{M-1}, \cdots, x_1, 0, 0, \cdots, 0]^T$   (reverse order and padding zeros to length N)
$\hat{\mathbf{y}} = \text{ifft}(\text{fft}(\mathbf{c}) \circ \text{fft}(\mathbf{w}))$
$\mathbf{y} = \hat{\mathbf{y}}(1 : L)$
```

Then an economic-size QR decomposition is applied to matrix $\mathbf{M}$

$$[\mathbf{Q}, \mathbf{R}] = \mathrm{qr}(\mathbf{M}). \tag{7.8}$$

Finally the low-rank approximation can be computed y projecting $\mathbf{H}$ onto the orthonormal basis $\mathbf{Q}$

$$\hat{\mathbf{H}} = \mathbf{Q}(\mathbf{Q}^{*}\mathbf{H})\,. \tag{7.9}$$

As is discussed in Cheng and Sacchi (2016), the parameter k in the randomized QR decomposition is a relaxation of the desired rank K. In other words, randomized QR decomposition is less stringent on the choice of rank (number of dips). The latter is very important since we usually do not have prior information about the rank of seismic data. We also point out that the QR decomposition is a very stable algorithm to provide the orthonormal basis $\mathbf{Q}$. Conversely, Lanczos bidiagonalization (Gao et al., 2013) tends to be unstable and an expensive re-orthogonalization on Lanczos vectors is often required.

7.2.2 Fast anti-diagonal averaging

We show that the anti-diagonal averaging of Hankel matrix can be efficiently computed via convolution. To clearly demonstrate the method, the singular value decomposition is adopted and we assume that the desired rank of the Hankel trajectory matrix equals 1. We can rewrite Equation 7.4 as follows

$$\hat{\mathbf{H}} = s_1\mathbf{u}_1\mathbf{v}_1. \tag{7.10}$$

Since we assume $\hat{\mathbf{H}}$ is a rank-1 matrix, s_1 is a constant and denotes the largest singular value and $\mathbf{u}_1$ and $\mathbf{v}_1$ denotes the first row and the first column of matrix $\mathbf{U}$ and $\mathbf{V}$, respectively. $s_1\mathbf{u}_1\mathbf{v}_1$ is also named an eigenimage of the matrix $\hat{\mathbf{H}}$. Combining Equation 7.5 and Equation 7.10 yields the following expression

$$\hat{D}_i = \begin{cases} \frac{s_1}{i}\sum_{j=1}^{i} u_{1_j} v_{1_{i-j+1}}, & 1 \le i \le M\,, \\ \frac{s_1}{M}\sum_{j=1}^{M} u_{1_j} v_{1_{i-j+1}}, & M \le i \le L\,, \\ \frac{s_1}{N-i+1}\sum_{j=i-L+1}^{M} u_{1_j} v_{1_{i-j+1}}, & L \le i \le N\,. \end{cases} \tag{7.11}$$

The filtered spatial data can be computed via the expression

$$\hat{D}_i \; = \; s_1 w_i \sum_{j=1}^{N} u_{1_j} v_{1_{i-j+1}}, \tag{7.12}$$

where w_i denotes constants that are determined by the size of Hankel matrix and are computed in advance. We can rewrite Equation 7.12 in its vector form

$$\hat{\mathbf{D}} \; = \; \mathbf{w} \circ [s_1(\mathbf{u}_1 * \mathbf{v}_1)], \tag{7.13}$$

where $\mathbf{u}_1 * \mathbf{v}_1$ denotes the convolution that can be efficiently computed using the Fast Fourier Transform. We can repeat the process to compute each eigenimage corresponding to each desired singular value of the rank-K approximation. The filtered data equals the summation of the K eigenimages

$$\hat{\mathbf{D}} \; = \; \mathbf{w} \circ [s_1(\mathbf{u}_1 * \mathbf{v}_1) + s_2(\mathbf{u}_2 * \mathbf{v}_2) + \cdots + s_K(\mathbf{u}_K * \mathbf{v}_K)]. \tag{7.14}$$

The computational complexity reduces to $\mathscr{O}(N \, \log(k))$.

The strategy can also be adopted for the randomized QR decomposition. Again if we assume $\hat{\mathbf{H}}$ is a rank-1 matrix, Equation 7.9 reduces to

$$\hat{\mathbf{H}} = \mathbf{q}_1 \mathbf{t}_1 \, , \tag{7.15}$$

where $\mathbf{t}_1 \; = \; \mathbf{q}_1^* \mathbf{H}$ and $\mathbf{q}_1$ denotes the first row of the matrix $\mathbf{Q}$ from the QR decomposition.

Apparently $\mathbf{t}_1$ can be computed via the fast Hankel matrix-vector product. In other words, the computational of the explicit Hankel matrix can be avoided in the anti-diagonal averaging, and thus, in the full SSA algorithm. Equation 7.15 then becomes comparable with Equation 7.10 with the filtered data given by

$$\hat{\mathbf{D}} \; = \; \mathbf{w} \circ (\mathbf{q}_1 * \mathbf{t}_1). \tag{7.16}$$

We can also extend the solution to find the rank-k approximation

$$\hat{\mathbf{D}} \; = \; \mathbf{w} \circ [(\mathbf{q}_1 * \mathbf{t}_1) + (\mathbf{q}_2 * \mathbf{t}_2) + \cdots + (\mathbf{q}_k * \mathbf{t}_k). \tag{7.17}$$

The resulting computational efficient SSA is summarized in algorithm 7.2.

Algorithm 7.2 Computational efficient SSA

```
Inputs:
    Seismic data: $\mathbf{D}$; rank: $k$
Output:
    SSA filtered data: $\hat{\mathbf{D}}$
for $\omega = \omega_{min} : \omega_{max}$ do
    $\mathbf{d} = \mathbf{D}(\omega, :)$
    $\Omega = \text{rand}(M, k)$    (generate random vectors)
    $\hat{\mathbf{d}} = 0$
    for $i = 1 : k$ do
        $\mathbf{M}(:, i) = \text{fast\_multiply}(\mathbf{d}, \Omega(:, i))$    (algorithm 1)
    end for
    $[\mathbf{Q}, \mathbf{R}] = \text{qr}[\mathbf{M}]$
    for $i = 1 : k$ do
        $\mathbf{q} = \mathbf{Q}(:, i)$
        $\mathbf{z} = \text{fast\_multiply}(\mathbf{d}, \mathbf{q})$
        $\hat{\mathbf{d}} = \hat{\mathbf{d}} + \text{ifft}(\text{fft}(\mathbf{q}) \circ \text{fft}(\mathbf{z}))$
    end for
    $\hat{\mathbf{D}}(\omega, :) = \hat{\mathbf{d}}$
end for
```

7.3 Computational efficient MSSA

We now turn our attention to the case where seismic data depends on more spatial dimensions. In other words, we extend the method of SSA to MSSA. In our analysis, we will use 3D seismic data knowing that he method can be extended into 5D without difficulties. For 3D seismic data, at a given monochromatic frequency ω_0, the single frequency slice can be denoted as follows,

$$D(\omega_0, x, y) = D_{j_1, j_2}\,, \tag{7.18}$$

where ω_0 represents the temporal frequency. $j_1 = 1, \cdots, N_1$ and $j_2 = 1, \cdots, N_2$ denote the trace index corresponding to each spatial dimension. N_1 and N_2 are the total number of traces in each dimension. For each frequency slice, we embed the multi-dimensional spatial data into a multi-level block Hankel (MBH) matrix. We first construct each row of the spatial data matrix D_{j_1, j_2} into a level-1 Hankel matrix as follows

$$\mathbf{H}^{(1)}j_2 = \begin{bmatrix} D_{1,j_2} & D_{2,j_2} & \cdots & D_{M_1,j_2} \\ D_{2,j_2} & D_{3,j_2} & \cdots & D_{M_1+1,j_2} \\ \vdots & \vdots & \ddots & \vdots \\ D_{L_1,j_2} & D_{L_1+1,j_2} & \cdots & D_{N_1,j_2} \end{bmatrix} . \tag{7.19}$$

Then we organize the second dimension into the Hankel structure by constructing a block Hankel matrix

$$\mathbf{H}^{(2)} = \begin{bmatrix} \mathbf{H}^{(1)}_1 & \mathbf{H}^{(1)}_2 & \cdots & \mathbf{H}^{(1)}_{M_2} \\ \mathbf{H}^{(1)}_2 & \mathbf{H}^{(1)}_3 & \cdots & \mathbf{H}^{(1)}_{M_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}^{(1)}_{L_2} & \mathbf{H}^{(1)}_{L_2+1} & \cdots & \mathbf{H}^{(1)}_{N_2} \end{bmatrix} , \tag{7.20}$$

Consequently, the size of the level-2 block Hankel matrix $\mathbf{H}^{(2)}$ is $L_1L_2 \times M_1M_2$, where we let $L_i = [\frac{N_i}{2}]$ and $M_i = N_i + 1 - L_i$, $i = 1, 2$.

Similar to the computational efficient SSA method discussed in the previous section, we adopt RQRD as an alternative to the SVD for computing the low-rank approximation of a given matrix as follows

$$\begin{aligned} \mathbf{M} &= \mathbf{H}^{(2)}\Omega, \\ [\mathbf{Q}, \mathbf{R}] &= \mathrm{qr}(\mathbf{M}), \\ \hat{\mathbf{H}}^{(2)} &= \mathbf{Q}(\mathbf{Q}^*\mathbf{H}^{(2)}) , \end{aligned} \tag{7.21}$$

where the random projection is computed using a fast MBH matrix-vector product. We refer the readers to Gao et al. (2013), where the authors discussed in details a fast multi-level Hankel matrix-vector product. Fast MBH matrix-vector product is also summarized in Algorithm 7.3.

The anti-diagonal averaging can be also efficiently computed via convolution. To clearly illustrate the fast anti-diagonal averaging algorithm, we assume the desired rank of the Hankel matrix is 1. The $\mathbf{Q}$ basis from the QR decomposition shrinks to a vector $\mathbf{q}$. Equation 7.22 can be rewritten as

$$\hat{\mathbf{H}}^{(2)} = \mathbf{q}\mathbf{t} , \tag{7.22}$$

where $\mathbf{t} = \mathbf{q}^*\mathbf{H}^{(2)}$. Conveniently, $\mathbf{t}$ can be computed via the aforedescribed fast MBH matrix-vector multiplication. The anti-diagonal averaging of a level-2 block Hankel matrix can be expressed as follows

$$\hat{D}_{j_1,j_2} = \frac{1}{W_1}\frac{1}{W_2}\sum_{i_1=1}^{j_1}\sum_{i_2=1}^{j_2}\mathbf{U}_{j_1,\,j_2}\mathbf{V}_{i_1-j_1+1,\,i_2-j_2+1} = \frac{1}{W_1}\frac{1}{W_2}\mathbf{U} * \mathbf{V} , \tag{7.23}$$

where $\mathbf{U}_{j_1,j_2} = \mathbf{q}_{(j_1 \times M_2)+j_2)}$ and $\mathbf{V}_{i_1,i_2} = \mathbf{t}_{(i_1 \times L_2)+i_2)}$ and $*$ denotes the matrix convolution in this case. In other words, $\mathbf{q}$ and $\mathbf{t}$ are reshaped into matrices based on the size of the Hankel matrices at each level. W_1 and W_2 count the number of summations along each anti-diagonal direction at each level. As a result, the anti-diagonal averaging of level-2 MBH matrix can be computed via matrix convolution. The latter can be computed efficiently via 2D Fast Fourier Transform (FFT). k matrix convolutions are required if the desired rank equals k. The resulting computational efficient MSSA is summarized in algorithm 7.4. The method can be extended to higher dimensional situations. For instance, both MBH matrix-vector multiplication and anti-diagonal averaging can be computed by 4D FFT for 5D seismic volumes.

Algorithm 7.3 Fast MBH matrix-vector product (level-2)

```
Inputs:
    Seismic data at a given frequency: $\mathbf{D}$; Vector $\mathbf{x}$;
    Size of Hankel matrix: $L_1$, $L_2$, $M_1$ and $M_2$
Output:
    Vector after multiplication: $\mathbf{y}$
$\underset{M_1 \times M_2}{\mathbf{X}} = \text{reshape}(\underset{M_1 M_2 \times 1}{\mathbf{x}})$
(1)  Rearrange the entries of spatial data $\mathbf{D}$ and the the entries of $\mathbf{X}$:
     $\underset{N_1 \times N_2}{\mathbf{C}} = \begin{bmatrix} \mathbf{D}_{M_1,M_2} & \cdots & \mathbf{D}_{N_1,M_2} & \mathbf{D}_{1,M_2} & \cdots & \mathbf{D}_{M_1-1,M_2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{M_1,N_2} & \cdots & \mathbf{D}_{N_1,N_2} & \mathbf{D}_{1,N_2} & \cdots & \mathbf{D}_{M_1-1,N_2} \\ \mathbf{D}_{M_1,1} & \cdots & \mathbf{D}_{N_1,1} & \mathbf{D}_{1,1} & \cdots & \mathbf{D}_{M_1-1,1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{M_1,M_2-1} & \cdots & \mathbf{D}_{N_1,M_2-1} & \mathbf{D}_{1,M_2-1} & \cdots & \mathbf{D}_{M_1-1,M_2-1} \end{bmatrix},$
     $\mathbf{Z} = \begin{bmatrix} \mathbf{X}_{M_1,M_2} & \cdots & \mathbf{X}_{1,M_2} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{M_1,1} & \cdots & \mathbf{X}_{1,1} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}$
(2)  $\mathbf{Y} = \text{ifft}_{2D}(\text{fft}_{2D}(\mathbf{C}) \circ \text{fft}_{2D}(\mathbf{Z}))$
(3)  $\mathbf{W} = \mathbf{Y}(1 : L_1, 1 : L_2)$
(4)  $\underset{L_1 L_2 \times 1}{\mathbf{y}} = \text{reshape}(\underset{L_1 \times L_2}{\mathbf{W}})$
```

Algorithm 7.4 Computational efficient MSSA

```
**Inputs:**
    Seismic data in frequency-space domain: $D$; rank: $k$; Size of Hankel matrix:
    $M_1$ and $M_2$;
**Output:**
    SSA filtered data: $\hat{D}$
**for** $\omega = \omega_{min} : \omega_{max}$ **do**
    $\mathbf{D} = D(\omega, :, :)$        (take the spatial data)
    $\Omega = \text{rand}(M_1 M_2, k)$        (generate random vectors)
    **for** $i = 1 : k$ **do**
        $\mathbf{M}(:, i) = \text{MBH\_multiply}(\mathbf{D}, \Omega(:, i))$        (algorithm 1)
    **end for**
    $[\mathbf{Q}, \mathbf{R}] = \text{qr}[\mathbf{M}]$
    **for** $i = 1 : k$ **do**
        $\mathbf{q} = \mathbf{Q}(:, i), \quad \mathbf{U}_{j_1,j_2} = \mathbf{q}_{(j_1 \times M_2)+j_2)}; \quad \mathbf{t} = \text{MBH\_multiply}(\mathbf{D}, \mathbf{q}), \quad \mathbf{V}_{i_1,i_2} = \mathbf{t}_{(i_1 \times L_2)+i_2)}$ ;

        $\hat{\mathbf{D}} = \hat{\mathbf{D}} + \text{ifft}_{2D}(\text{fft}_{2D}(\mathbf{U}) \circ \text{fft}_{2D}(\mathbf{V}))$     (convolution)
    **end for**
    $\hat{D}(\omega, :) = \hat{\mathbf{D}}$
**end for**
```

7.4 Performance analysis

To test the efficiency of the proposed computational efficient MSSA algorithm, we compared the performance of different implementations of SSA for random noise attenuation of 3D seismic data patches of different size. SSA rank reduction filter is applied on 50 frequency slices are performed in each run. Table 7.1 shows the signal-to-noise ratio of the data after de-noising and the computational time for the conventional multi-dimensional SSA (MSSA) (Oropeza and Sacchi, 2011), Randomized MSSA (Oropeza and Sacchi, 2011), SSA via Lanczos bi-diagonalization (Gao et al., 2013) and the proposed computational efficient MSSA (FSSA), respectively. The four algorithms exhibit similar capabilities in the attenuation of random noise. However, the proposed computational efficient MSSA outperforms the other three methods in computational efficiency.

$(N_x \times N_y)$	MSSA		RMSSA		Lanczos		FSSA	
	Time(s)	S/N(dB)	Time(s)	S/N(dB)	Time(s)	S/N(dB)	Time(s)	S/N(dB)
20×20	2.033	5.021	0.667	5.052	0.411	5.001	0.237	5.302
40×40	12.76	9.667	7.669	10.01	1.124	9.801	0.678	10.22
60×60	58.85	12.01	41.12	11.77	2.485	12.56	1.325	11.96
80×80	152.7	14.05	142.5	14.28	6.933	14.77	2.998	14.68

Table 7.1: Comparison of computational accuracy and efficiency for different SSA methods: conventional 3D MSSA (MSSA), randomized MSSA (RMSSA), Lanzcos Bidagonalization (Lanzcos), and the proposed computational efficient MSSA (FSSA). In this experiment, the subset size for FSSA is three times of the desired rank.

7.5 Examples

7.5.1 Synthetic example - 3D reconstruction

To test the effectiveness of the proposed algorithm, we first adopt a synthetic data set which is simulated with three linear events in x-y-t. The data contain $N_x \times N_y$ traces, with $N_x = 20$ and $N_y = 25$. The temporal length of the window is 0.4s with a sampling interval 0.002s. This represents a typical size of data patches commonly used for seismic data reconstruction (Oropeza and Sacchi, 2011). The noise-free and ideally sampled data are shown in Figure 7.1 (a). To mimic the seismic data that are observed in the field, we randomly remove 50% of the traces. In addition, we assume data that are contaminated with random noise Figure 7.1 (b). We adopted the classical SSA recovery algorithm proposed by Oropeza and Sacchi (2011).

$$D^{\nu} = \alpha D_{obs} + (I - \alpha S) F_{FSSA}(D^{\nu-1}), \tag{7.24}$$

where ν is the number of iteration and S denotes the sampling operator. F_{FSSA} denotes the proposed fast SSA filtering algorithm. The scalar $\alpha \leq 1$ denotes the weight for averaging the observed and SSA filtered data. The de-noised and reconstructed result is shown in Figure 7.1 (c). The reconstructed data becomes comparable with the noise-free and ideally sampled data. Figure

7.2 exhibits a slice ($x = 5$) of the data cube. The algorithm effectively suppresses the noise while reconstructing the coherent signal.

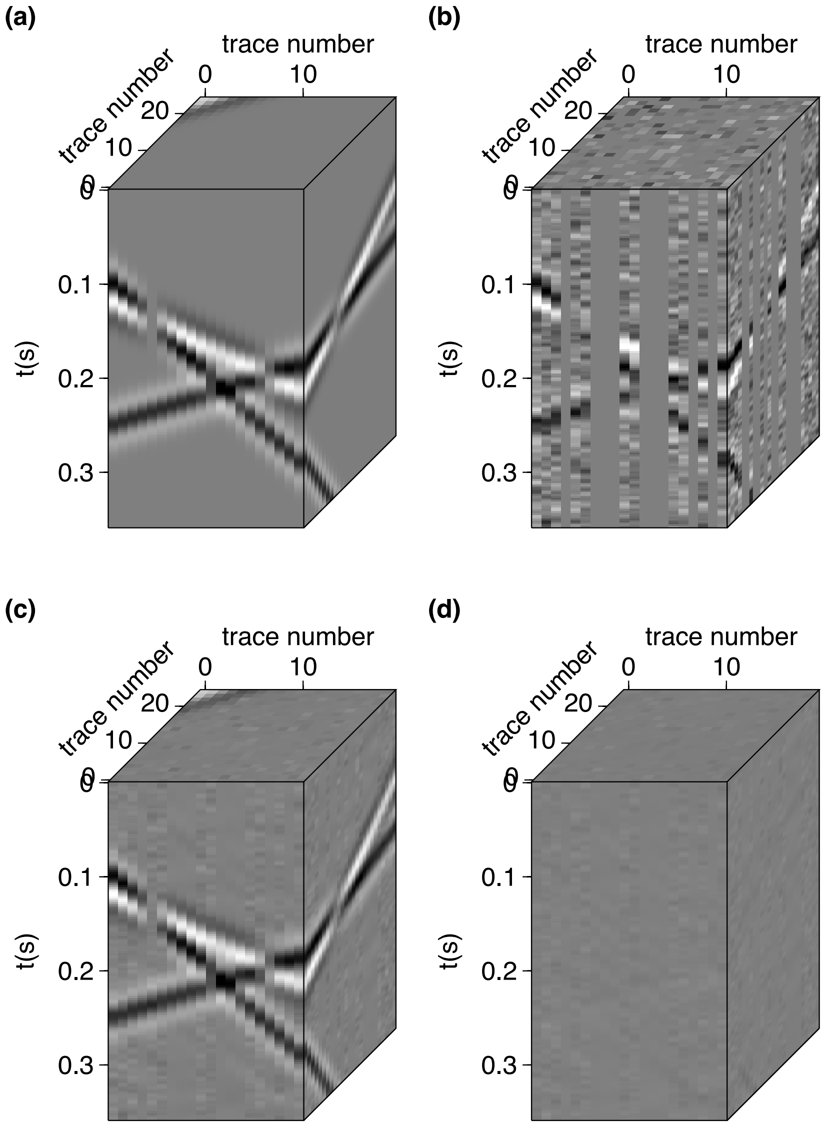


Figure 7.1: Results of SSA de-noising and reconstruction: synthetic example. (a) Ideal noise-free seismic data (b) Data with incoherent noise and 50% decimation (c) Data after fast SSA reconstruction (d) Differences between (a) and (c).

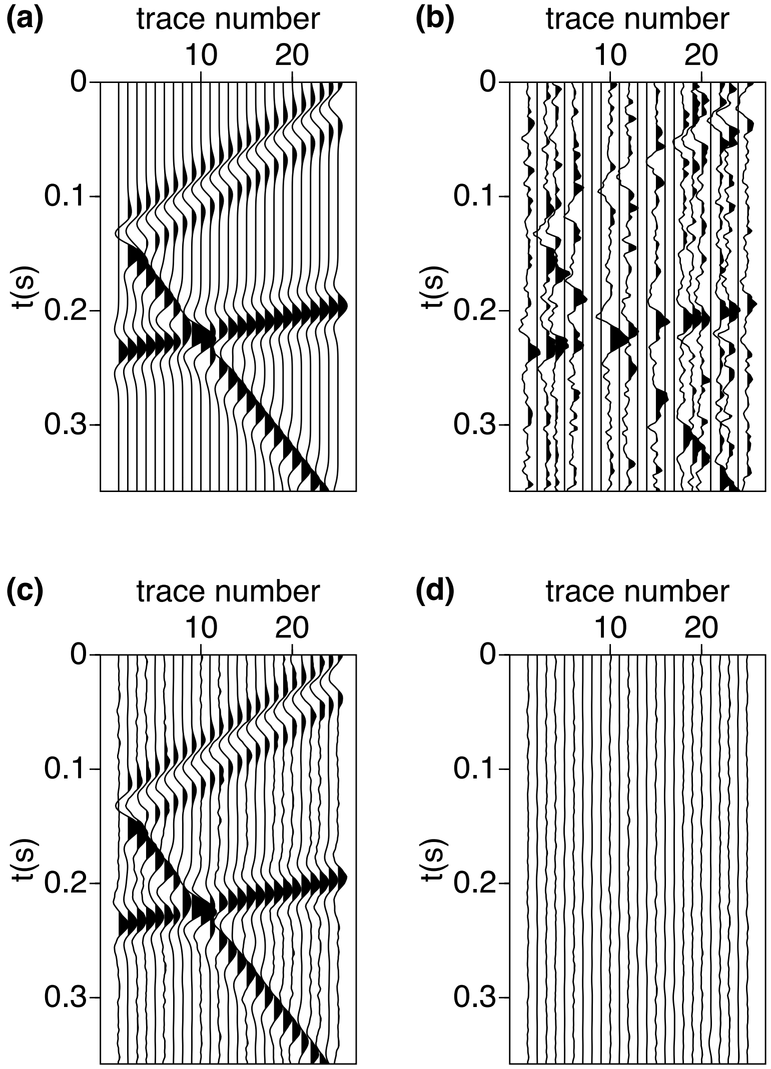


Figure 7.2: Results of SSA de-noising and reconstruction synthetic example (Slice at x=5). (a) Ideal noise-free seismic data (b) Data with incoherent noise and 50% decimation (c) Data after fast SSA reconstruction (d) Differences between (a) and (c).

7.5.2 Real data example - 3D reconstruction

We adopt the computational efficient MSSA algorithm to reconstruct a small patch of a prestack seismic data set. The data are rearranged into 15 CMP bins and 45 offset bins. NMO correction has been applied to each CMP gather before reconstruction. As is shown in Figure 7.3 (a), around 50% of traces are missing after binning. The result of computational efficient MSSA reconstruction is shown in Figure 7.3 (b). In this example, we choose $\alpha = 0.4$ and $k = 8$. The missing traces are successfully interpolated with signal that is coherent along the spatial direction. We also exhibits a single CMP gather before and after reconstruction in Figure 7.4.

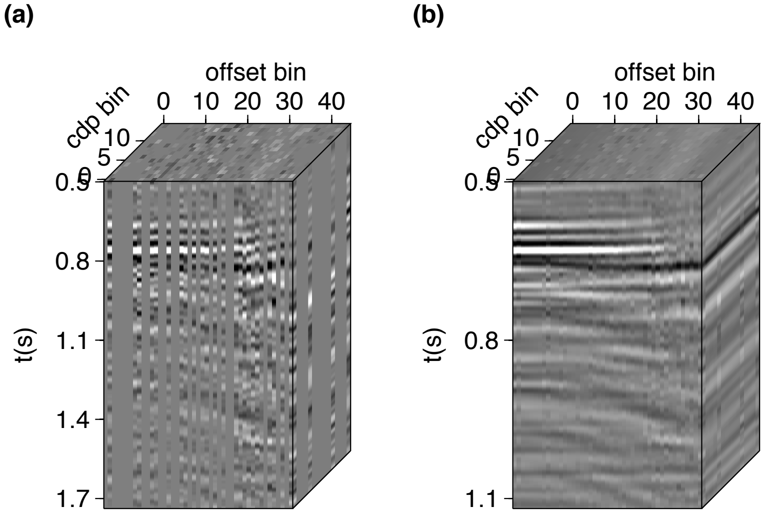


Figure 7.3: Results of SSA de-noising and reconstruction for 3D field data. (a) A small patch of field data before reconstruction (b) De-noised and reconstructed data via the fast SSA algorithm.

7.5.3 Real data example - 5D reconstruction(WCSB)

We also adopt an example involves the reconstruction of a real prestack 5D volume via the proposed computational efficient MSSA method. We extracted a small patch of the data set shown as the rectangular area in Figure 7.5 (Kreimer and Sacchi, 2012). After binning, the seismic traces were assigned to the midpoint-offset grid which has dimensions $16 \times 18 \times 12 \times 12$. Around 40% of the 4D grid do not contain any observation. We selected a time window in the interval $900 - 1250$ msec that corresponds to 351 samples. Figure 7.6 (a) shows a subset of the data prior to reconstruction. This subset is acquired by fixing CMP$_y$ and Offset$_y$. Figure 7.6 (b) illustrates the reconstructed CMP gathers via the proposed computational efficient MSSA method with $\alpha = 0.4$ and $k = 18$. Figure 7.7 shows a set of offset gathers with the same CMP$_y$ and Offset$_y$ fixed, where we display the original and reconstructed offset gathers.

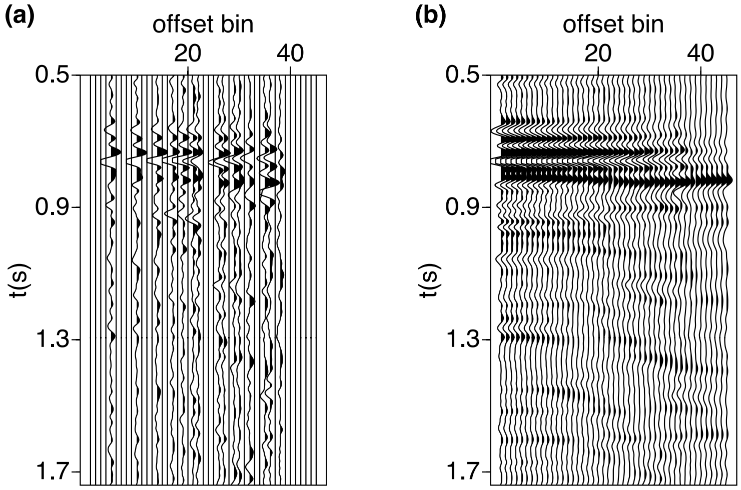


Figure 7.4: Results of SSA de-noising and reconstruction for 3D field data (Slice at cdp bin=8) (a) A small patch of field data before reconstruction (b) De-noised and reconstructed data via the fast SSA algorithm.

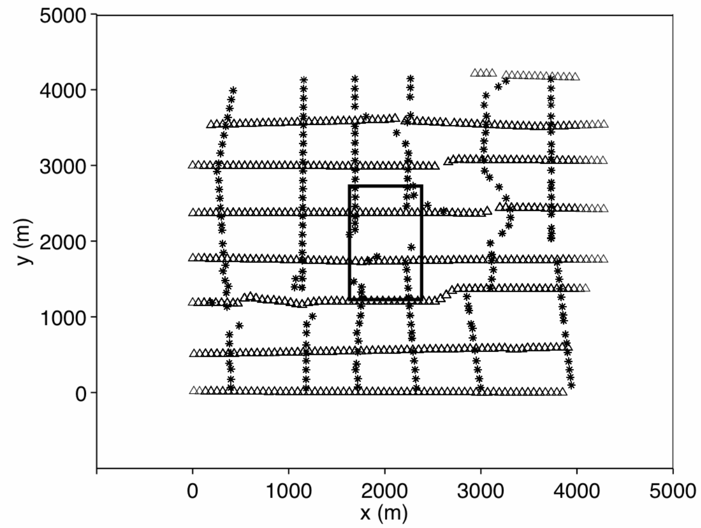


Figure 7.5: Results of SSA de-noising and reconstruction for 3D field data (Slice at cdp bin=8) (a) A small patch of field data before reconstruction (b) De-noised and reconstructed data via the fast SSA algorithm.

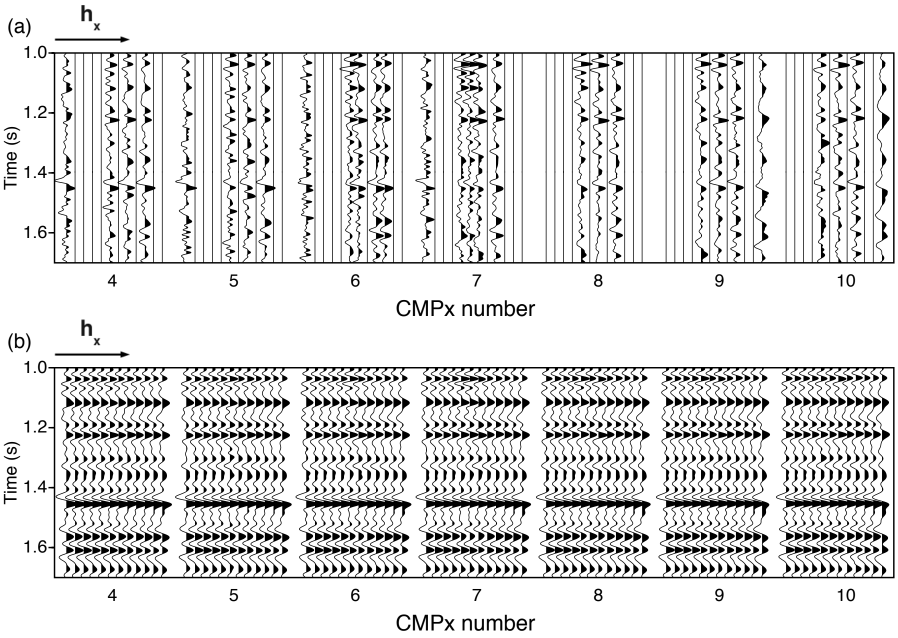


Figure 7.6: Results of SSA de-noising and reconstruction for 3D field data (Slice at cdp bin=8) (a) A small patch of field data before reconstruction (b) De-noised and reconstructed data via the fast SSA algorithm.

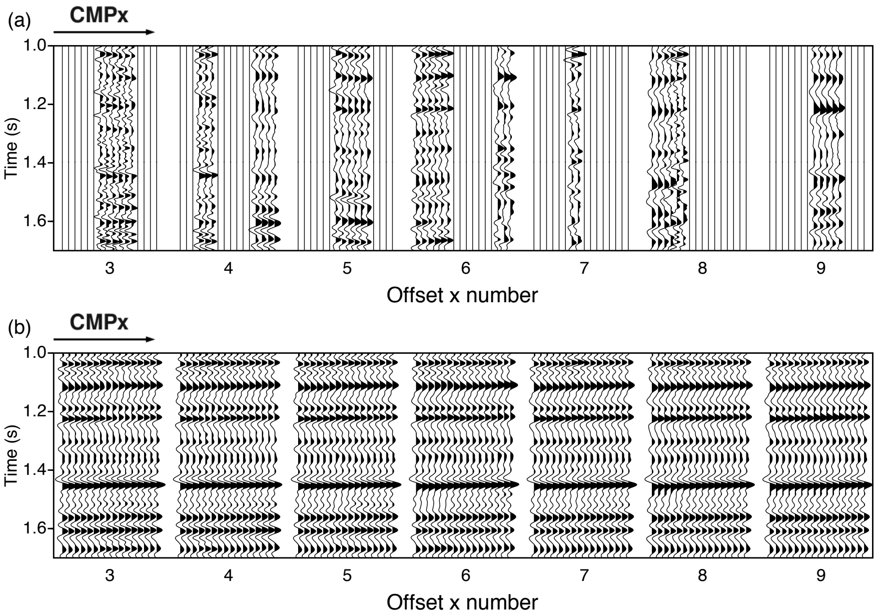


Figure 7.7: Results of SSA de-noising and reconstruction for 3D field data (Slice at cdp bin=8) (a) A small patch of field data before reconstruction (b) De-noised and reconstructed data via the fast SSA algorithm.

7.6 Conclusions

This article illustrates the computational efficient Multi-dimensional Singular Spectrum Analysis method for the reconstruction and de-noising multi-dimensional prestack seismic volumes. Important savings for large-scale dimensionality reduction problems are attainable via the proposed randomized QR decomposition. To avoid the construction of multi-level block Hankel matrices, we proposed fast multi-level block Hankel matrix-vector product and a fast convolution for the final anti-diagonal averaging. Both multi-level block Hankel matrix-vector product and the fast anti-diagonal averaging are computed via Fast Fourier Transform. The proposed method significantly improves the computational efficiency.

References

Broomhead, D., and G. King, 1986, Extracting qualitative dynamics from experimental data: Physica D, **20**, 217–236.

Cadzow, J., 1988, Signal enhancement-a composite property mapping algorithm: Acoustics, Speech and Signal Processing, IEEE Transactions on, **36**, 49 –62.

Canales, L. L., 1984, Random noise reduction: SEG Technical Program Expanded Abstracts, 525–527.

Cheng, J., and M. D. Sacchi, 2016, Fast dual-domain reduced-rank algorithm for 3d deblending via randomized qr decomposition: Geophysics, **81**, V89–V101.

Chiron, L., M. A. van Agthoven, B. Kieffer, C. Rolando, and M.-A. Delsuc, 2014, Efficient denoising algorithms for large experimental datasets and their applications in fourier transform ion cyclotron resonance mass spectrometry: Proceedings of the National Academy of Sciences, **111**, 1385–1390.

Duijndam, A., M. Schonewille, and C. Hindriks, 1999, Reconstruction of band-limited signals, irregularly sampled along one spatial direction: Geophysics, **64**, 524–538.

Gao, J., M. D. Sacchi, and X. Chen, 2013, A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions: Geophysics, **78**, V21–V30.

Golub, G., and C. van Loan, 1996, Matrix computations, third edition ed.: London: The Johns Hopkins University Press.

Hennenfent, G., L. Fenelon, and F. J. Herrmann, 2010, Nonequispaced curvelet transform for seismic data reconstruction: A sparsity-promoting approach: Geophysics, **75**, WB203–WB210.

Herrmann, F. J., and G. Hennenfent, 2008, Non-parametric seismic data recovery with curvelet frames: Geophysical Journal International, **173**, 233–248.

Kabir, M. N., and D. Verschuur, 1995, Restoration of missing offsets by parabolic radon transform: Geophysical Prospecting, **43**, 347–368.

Korobeynikov, A., 2010, Computation- and space-efficient implementation of ssa: Statistics and Its Interface, **3**, 357–368.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, V133–V122.

Liu, B., and M. Sacchi, 2004, Minimum weighted norm interpolation of seismic records: Geophysics, **69**, 1560–1568.

O'Leary, D., and J. Simmons, 1981, A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems: Siam J. Sci. Stat. Comput, **2**, 474–489.

Oropeza, V., and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Sacchi, M. D., 2009, F-X singular spectrum analysis: CSPG CSEG CWLS Convention, 392–395.

Soubaras, R., 1994, Signal-preserving random noise attenuation by the fx projection: 64th Annual International Meeting, SEG, Expanded Abstracts, 1576–1579.

Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, **56**, 785–794.

Trad, D. O., T. J. Ulrych, and M. D. Sacchi, 2002, Accurate interpolation with high-resolution time-variant radon transforms: Geophysics, **67**, 644–656.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation: 80th Annual International Meeting, SEG, Expanded Abstracts Annual International Meeting, SEG, Expanded Abstracts, 3829–3833.

Trickett, S. R., 2003, F-xy eigenimage noise supression: Geophysics, **68**, 751–759.

Trickett, S. R., and L. Burroughs, 2009, Prestack rank-reducing noise suppression: 79th Annual International Meeting, SEG, Expanded Abstracts.

Xu, S., Y. Zhang, D. Pham, and G. Lambare, 2005, Antileakage Fourier Transform for seismic data regularization: Geophysics, **79**, V87–V95.

Edge-preserving smoothing for simultaneous-source FWI model updates in high-contrast velocity models

Amsalu Y. Anagaw[1] and Mauricio D. Sacchi

Abstract

Full waveform inversion (FWI) can provide accurate estimates of subsurface model parameters. In spite of its success, the application of FWI in areas with high-velocity contrast remains a challenging problem. Quadratic regularization methods are often adopted to stabilize inverse problems. Unfortunately, edges and sharp discontinuities are not adequately preserved by quadratic regularization techniques. Throughout the iterative FWI method, an edge-preserving filter, on the other hand, can gently incorporate sharpness into seismic images. For every point in the seismic image, edge-preserving smoothing assigns the average value of the most uniform window neighboring the point. Edge-preserving smoothing generates piecewise-homogeneous images with enhanced contrast at boundaries. We adopt a simultaneous-source frequency-domain FWI, based on quasi-Newton optimization, in conjunction with an edge-preserving smoothing filter to retrieve high-contrast velocity models. The edge-preserving smoothing filter gradually removes the artifacts created by simultaneous-source encoding. We also propose a simple model update to prevents disrupting the convergence of the optimization algorithm. Finally, we provide tests to examine the proposed algorithm.

[1] Email: aanagaw@ualberta.ca

8.1 Introduction

The primary purpose of Full Waveform inversion (FWI) is to estimate high-resolution physical properties of geological structures by minimizing the data misfit between observed and modeled seismograms through local iterative optimization techniques (Lailly, 1983; Tarantola, 1987; Stekl and Pratt, 1998). FWI is an ill-posed inverse problem (Virieux and Operto, 2009) and, therefore, regularization and preconditioning strategies are often required to obtain stable estimates of velocity models. Regularization methods also serve to impose desired features on subsurface images (Zhdanov, 2002). Quadratic regularization methods are often adopted to deliver smooth models (Constable et al., 1987). In an attempt to preserve discontinuities in images, non-quadratic regularization techniques have become popular in the field of image processing (Künsch, 1994; Geman and Reynolds, 1992; Charbonnier et al., 1997). The basic idea involves adopting a regularization constraint that imposes sparsity on the spatial derivatives of the desired image. Non-quadratic regularization methods that safeguard high-velocity contrasts were adopted for linearized Born inversion (Youzwishen and Sacchi, 2006; Anagaw and Sacchi, 2012) and FWI (Akcelik et al., 2002; Anagaw and Sacchi, 2011; Guitton, 2012; Smithyman et al., 2015; Esser et al., 2016; Brandsberg-Dahl et al., 2017; Peters and Herrmann, 2017). Farquharson and Oldenburg (1998) investigated edge-preserving regularization techniques for the solution of electromagnetic inverse problems. Similarly, Valenciano et al. (2005) proposed a non-quadratic edge preserving regularization for interval-velocity inversion from RMS velocities.

Edge-preserving techniques for inverse problems require the resolution of a non-quadratic regularization problem that often contains more than one trade-off parameter (Anagaw and Sacchi, 2011; Valenciano et al., 2005; Farquharson and Oldenburg, 1998). Our experience indicates that estimating optimal trade-off parameters controlling goodness-of-fit and edge-enhancement is not an easy task when one tries to solve nonlinear inverse problems like FWI. An alternative solution to edge preservation via regularization involves adopting a model update followed by edge-preserving smoothing. The process, often called edge-preserving smoothing, corresponds to filtering techniques capable of denoising images with sharp discontinuities. The basic principle of edge-preserving smoothing is to apply smoothing (averaging) to uniform areas of the image and to switch off smoothing across edges. In this paper, we propose to use the edge-preserving smoothing algorithm, originally proposed by AlBinHassan et al. (2006), to preserve edges and discontinuities of velocity models recovered via FWI. In our approach, the edge-preserving smoothing filter is applied to each FWI model update via a scheme that gently sharpens the image without degrading data fitting. We also point out that Zhang and Zhang (2017) investigated a similar strategy for seismic traveltime tomography where a 1D edge-preserving smoothing operator is used in conjunction with a re-weighted least-squares technique to enhance edges in tomographic velocity inversion.

A 2D acoustic simultaneous-source frequency-domain FWI algorithm (Stekl and Pratt, 1998; Anagaw and Sacchi, 2014) is employed to generate model updates that undergo edge-preserving smoothing. We have chosen the quasi-Newton l-BFGS optimization engine (Nocedal, 1980) to estimate the model perturbations. However, it is clear that one could adopt a different optimization engine for computing the model perturbations. We have tested our algorithm with the BP/EAGE velocity model (Billette and Brandsberg-Dahl, 2005). The BP/EAGE model contains high-contrast velocity boundaries and complex salt bodies making it an excellent candidate for experimenting with the proposed algorithm.

8.2 Theory

8.2.1 FWI updates

Our work pertains to acoustic FWI with frequency-domain solvers and source encoding (Stekl and Pratt, 1998; Anagaw and Sacchi, 2014). Waveform inversion often uses the least-squares misfit defined as the l_2 norm of the residual between the observed data $\mathbf{d}^{obs}$ and synthetic data $\mathbf{d}^{cal}(\mathbf{m})$:

$$J(\mathbf{m}) = \frac{1}{2} \parallel \mathbf{d}^{obs} - \mathbf{d}^{cal}(\mathbf{m}) \parallel^{\mathbf{2}} \ . \tag{8.1}$$

The minimization of the misfit function is a nonlinear problem where from measured wavefields one attempts to estimate the P-wave velocity model $\mathbf{m}$. We omit the derivation of the frequency-domain FWI algorithm and just state that in each iteration we adopt the l-BFGS method (Nocedal, 1980) to generate model perturbations $\Delta\mathbf{m}_k$ where k indicates the iteration number. During the iterative process, for each frequency group, the source encoding scheme is kept constant. We remind the reader that the perturbation $\Delta\mathbf{m}_k$ corresponds, in our case, to a gradient contaminated by cross-talk artifacts introduced by source encoding (Anagaw and Sacchi, 2014). We propose the following model update scheme:

$$\mathbf{u}_k = \mathbf{m}_k + \alpha\Delta\mathbf{m}_k \tag{8.2}$$

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \beta(\mathscr{P}\mathbf{u}_k - \mathbf{m}_{\mathbf{k}})\,, \tag{8.3}$$

where $\mathbf{u}_k$ denotes a temporary update of the model. The scalar α is the step length determined via line search. The scalar $\beta \in (0, 1)$ controls the tradeoff between descending toward the minimum of the cost function J and introducing contrast in the model update. The operator $\mathscr{P}$ symbolizes the edge-preserving smoothing filter. Algorithm 8.1 is the pseudocode for the proposed multi-scale FWI with edge-preserving smoothing. Equation 3 can also be written as follows:

$$\mathbf{m}_{k+1} = (1 - \beta)\, \mathbf{m}_k + \beta\, \mathscr{P}\mathbf{u}_k\,. \tag{8.4}$$

Note that equation 4 shows that in each iteration, we keep the current model update weighted by $(1 - \beta)$ plus the edge-enhanced model weighted by β. The scalar β is a small number to guarantee convergence to the minimum of the cost function J. In other words, expression 8.4 constraints the new update $\mathbf{m}_{k+1}$ to remain close $\mathbf{m}_k$ to guarantee converge to the minimum of the cost function J.

Algorithm 8.1 Pseudo-code: Multiscale FWI using quasi-Newton l-BFGS method with edge-preserving smoothing. Source encoding kept constant for each group frequency.

```
Estimation of $\mathbf{m} = \underset{\mathbf{m}}{argmin}\ J(\mathbf{m})$
*Initialization:* $\mathbf{m}_0$
**for** $\omega_i$ in 1: $N_\omega$ **do**
    **for** $k$ in 1: $max\_iter$ **do**
        Compute perturbation $\Delta\mathbf{m}_k$ via l-BFGS
        Estimate $\alpha$ via line search
        $\mathbf{u}_k = \mathbf{m}_k + \alpha\triangle\mathbf{m}_k$
        Apply edge-preserving smoothing and update the model
        $\mathbf{m}_{k+1} = \mathbf{m}_k + \beta(\mathscr{P}\mathbf{u}_k - \mathbf{m}_k)$
    **end for**
**end for**
```

8.2.2 Edge-preserving smoothing

For the 2D edge-preserving smoothing operator design, we follow the procedure presented by AlBinHassan et al. (2006). We consider a 2D velocity model and select 5 × 5 pixels moving windows. The central point of each 5 × 5 window is the point to which one assigns the estimate of the edge-preserving filter. In AlBinHassan et al. (2006) paper, a total of 9 neighbouring overlapping windows around the central point are chosen: four hexagonal windows, four pentagonal windows, and one square window. We adopted the same procedure but with 10 overlapping windows around the central point, where the extra window is the full 5 × 5 window. For each window, we compute its mean and standard deviation. The window with the lowest standard deviation is the most uniform window, and its average value is assigned to the central point. The process as demonstrated in AlBinHassan et al. (2006) smooths homogeneous areas and preserves discontinuities across boundaries.

8.3 Numerical Examples

We consider a portion of the original BP/EAGE velocity model (Billette and Brandsberg-Dahl, 2005), where large velocity contrasts and complex salt bodies are present, as shown in Figure 8.1. Figures 8.1a and b are the true and smooth velocity models used for FWI, respectively. The BP/EAGE velocity model was gridded onto a coarser grid of size 50 m by 50 m. The actual velocity model contains salt domes of high-velocity contrast, subsalt low-velocity zones under the salt, and several anomalies that offer an ideal velocity model to test the algorithm. Synthetic data were generated via a finite-difference method. A total number of 100 sources and 298 receivers were computed. The sources are placed 50 m below the surface, while the receivers on the surface of the earth. A Ricker wavelet of a central frequency of 10. Hz represents the source signature. The source encoding scheme contains four super-shots, and each super-shot includes 25 individual sources. The edge-preserving smoothing filter suppresses incoherent cross-talk noise introduced by the source encoding. A set of thirteen discrete frequencies between 2. Hz and 12. Hz were selected for the input data to our multiscale FWI code and frequencies were grouped. The multiscale frequency grouping consists of five groups with an overlap of 4 frequencies (Anagaw and Sacchi, 2014). The inversion is then carried out in a sequential approach starting from the first group of frequencies (containing low frequencies) to the last group (containing high frequencies). We run the inversion with and without applying the edge-preserving smoothing operator for a maximum of 60 iterations. The parameter $\beta \in (0,1)$ is selected as follows. First, we run a standard FWI until achieving convergence. The algorithm converges after about 60 iterations. We, then, execute the algorithm with edge-preserving smoothing for different values of β for 60 iterations. Finally, we pick β such that the convergence curves of the conventional FWI and the proposed algorithm are similar. In this example, our numerical experiments have indicated that adopting $\beta = 0.05$ yields results with enhanced edges without affecting the minimization of the cost function J.

Figures 8.2a and b depict the reconstructed velocity models obtained via FWI without and with the edge-preserving smoothing operator. We also clarify that the edge-preserving smoothing operator was turned off for the first frequency group. In the earliest stage of the inversion, the algorithm captures the long-wavelength structure of the velocity model, and as it proceeds with the remaining frequency groups, the edge-preserving smoothing filter sharpens the updated model.

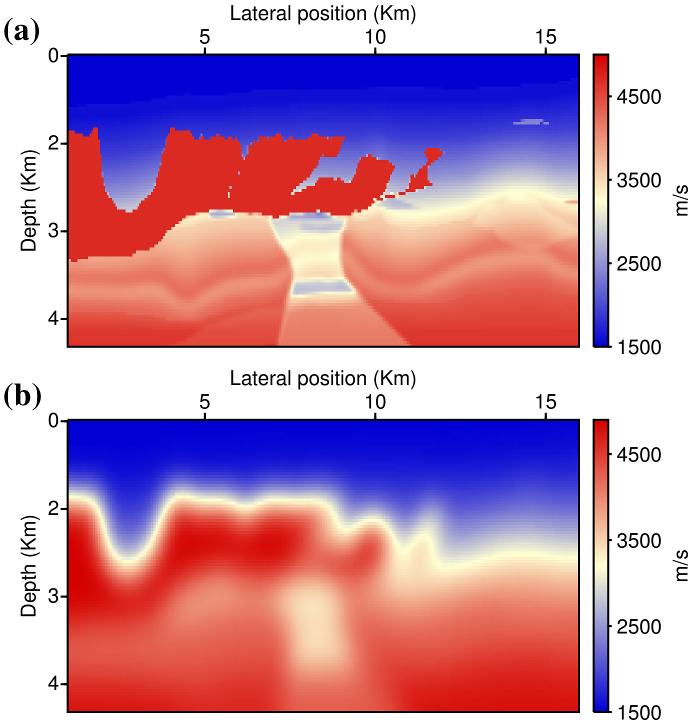


Figure 8.1: Portion of the original BP/EAGE velocity model. True velocity model (a) and smooth velocity model (b). The smooth velocity model was adopted as the starting model for the inversion.

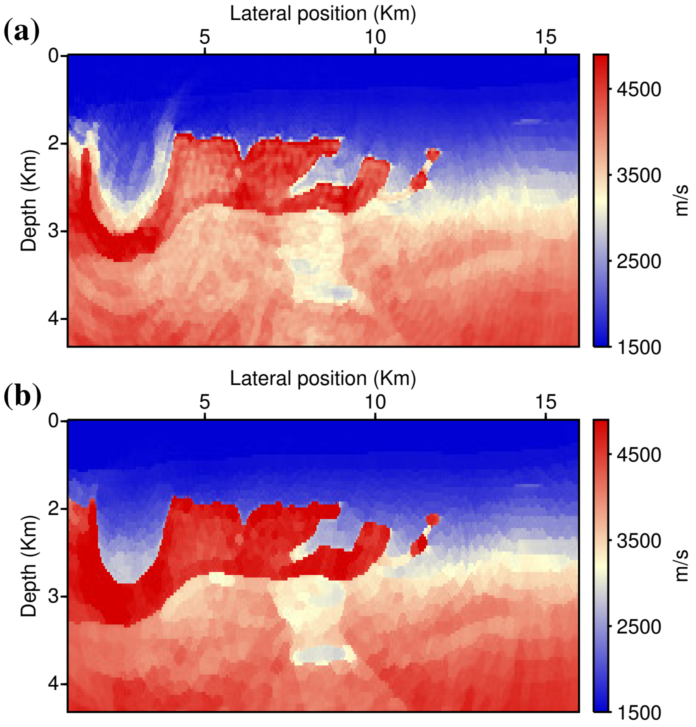


Figure 8.2: (a) Reconstructed velocity model with classical FWI (no edge-preserving smoothing). (b) Inverted velocity model with edge-preserving smoothing FWI update described in this article.

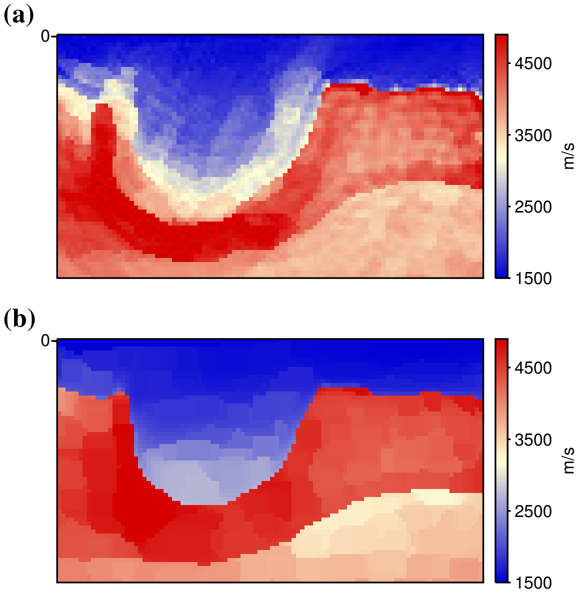


Figure 8.3: (a) and (b) are windows extracted from the top left zone of Figures 8.2a and 2b, respectively.

To complete our analysis, we also show windows of the estimated models. Figures 8.3a and b show the top left sections of the reconstructed velocity. Model updates with edge-preserving smoothing filtering not only provide an image with enhanced resolution but also salt boundaries that were correctly retrieved. The proposed algorithm is also able to delineate both shallow and subsalt velocity anomalies.

Figure 8.4 shows vertical velocity profiles extracted at horizontal position 1.80 Km on the left and at horizontal position 3.20 Km on the right, respectively. Figure 8.5 shows lateral velocity profiles extracted at depth levels 2.50 Km and 3.60 Km, respectively. These selected positions are highlighted by green arrow in Figure 8.2a. Our results highlight that the proposed algorithm yields images with enhanced velocity contrasts.

Figure 8.6 shows the convergence rate comparisons of the algorithm with and without applying EPS in the FWI. The misfit function is normalized by its first misfit function value. Figure 8.6 a&b are the misfit curves for group frequencies of maximum frequency 6 Hz and 9 Hz, respectively. As we see from the convergence rates of the optimization of FWI with EPS for models with high-velocity contrast, the a relative faster reduction in the misfit function is observed when we employ the EPS operator.

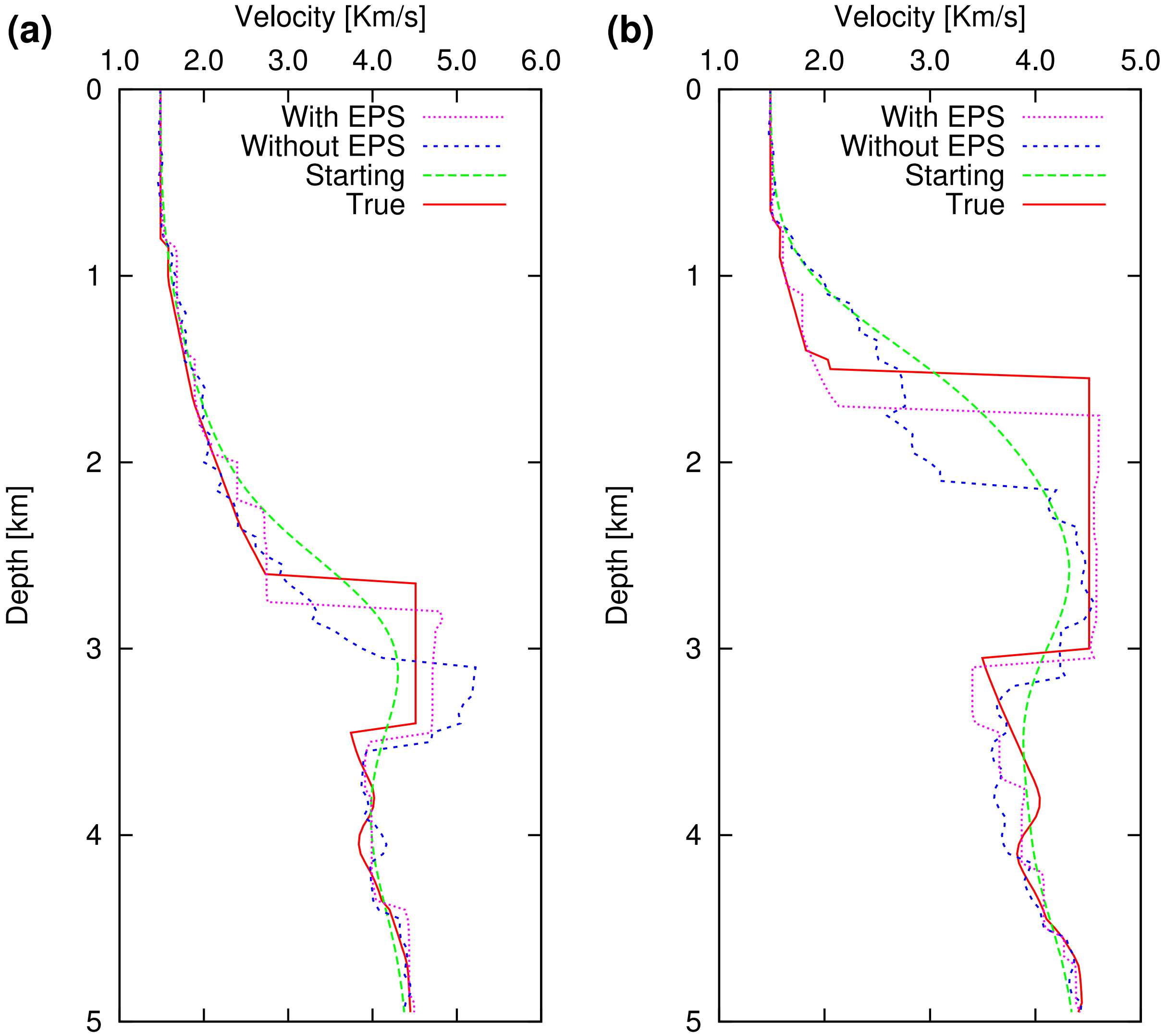


Figure 8.4: Comparison of vertical velocity profiles of the reconstructed velocity models portrayed in Figure 8.2. The velocity profiles are extracted at horizontal positions 1.80 Km (a) and 3.20 Km (b).

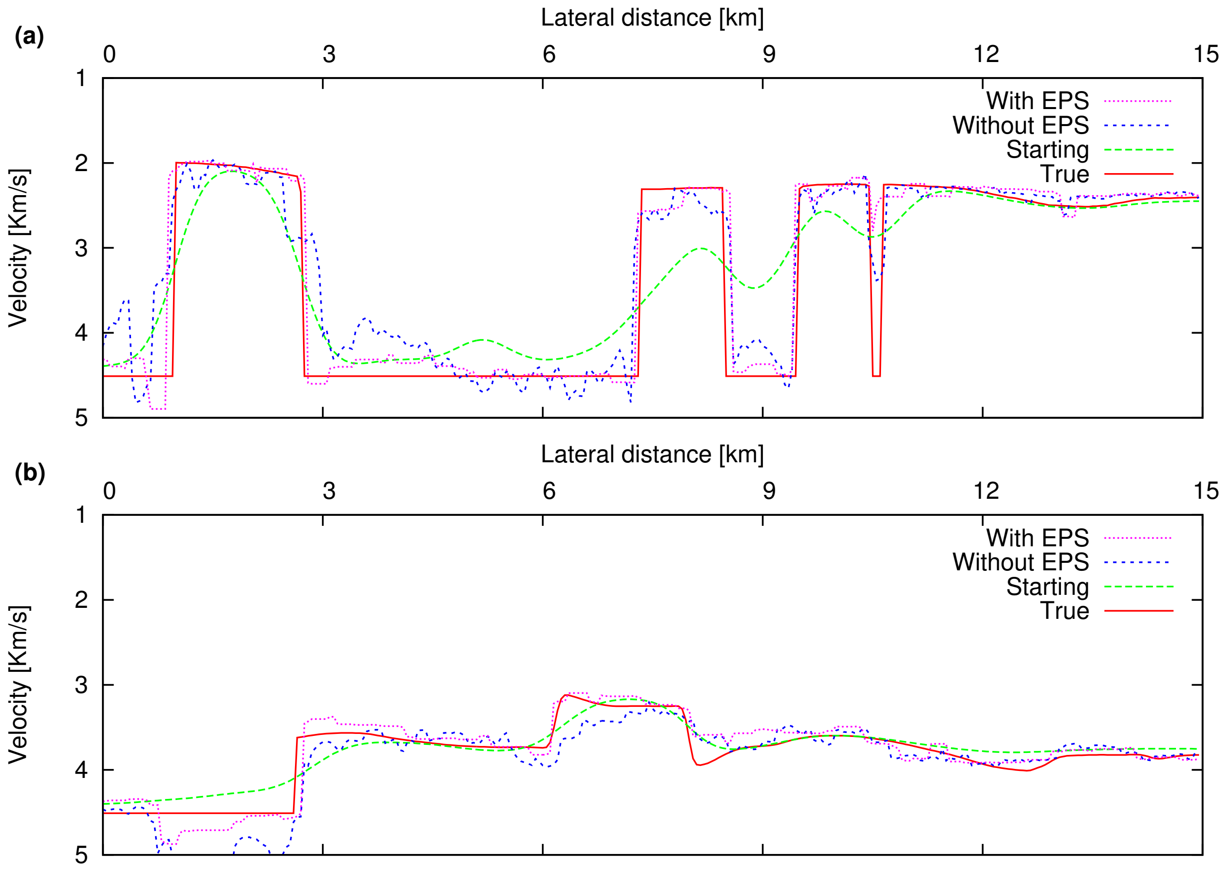


Figure 8.5: Comparison of lateral velocity profiles of the reconstructed velocity models portrayed in Figure 8.2. The lateral velocity profiles are extracted at depth levels 2.50 Km (a) and 3.60 Km (b).

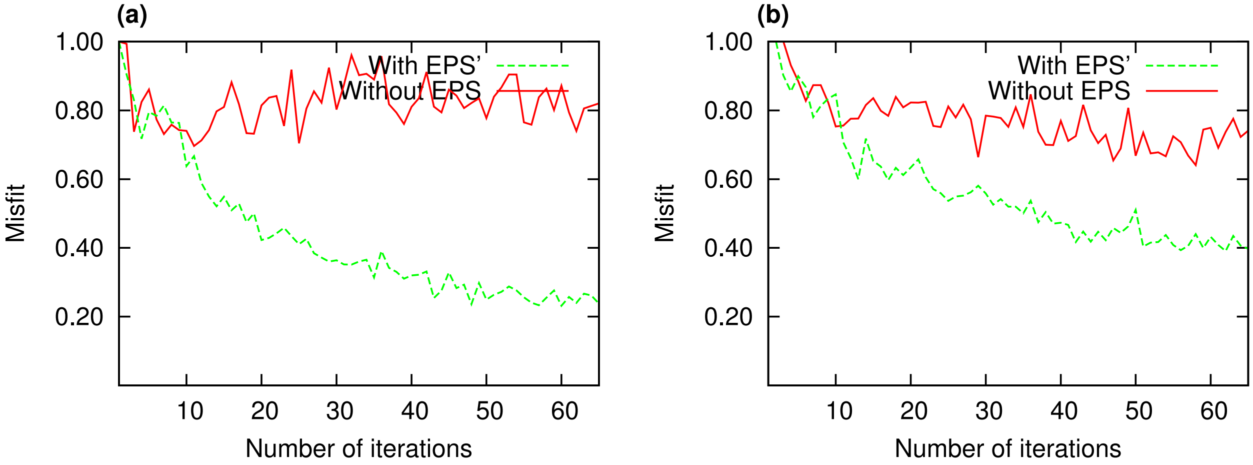


Figure 8.6: Relative data misfit reduction curve with and without applying EPS. Misfit curves for group frequencies of maximum frequency 6 Hz (a) and 9 Hz (b).

8.4 Conclusions

We have developed a 2D FWI algorithm with a model update that includes an edge-preserving smoothing filter to improve the retrieval of high-contrast images. The edge-preserving smoothing technique searches for the most uniform neighbourhood around each point in the image. The average of the window with minimum standard deviation is assigned to the central point. We have equipped our algorithm with a scalar parameter that controls the sharpness of the update while the iterative solution is guaranteed to descend to the minimum of the objective function of the FWI problem.

8.5 ACKNOWLEDGMENTS

We thank the sponsors of Signal Analysis and Imaging Group (SAIG) at the University of Alberta and The Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this project.

References

Akcelik, V., G. Biros, and O. Ghattas, 2002, Parallel multiscale Gauss-Newton-Krylov Methods for inverse wave propagation: 41.

AlBinHassan, N. M., Y. Luo, and M. N. Al-Faraj, 2006, 3d edge-preserving smoothing and applications: Geophysics, **71**, P5–P11.

Anagaw, A., and M. D. Sacchi, 2011, Full waveform inversion with total variation regularization: CSPG CSEG CWLS Convention, 1–4.

Anagaw, A. Y., and M. D. Sacchi, 2012, Edge-preserving seismic imaging using the total variation method: Journal of Geophysics and Engineering, **9**, 138.

———, 2014, Comparison of multifrequency selection strategies for simultaneous-source full-waveform inversion: Geophysics, **79**, R165–R181.

Billette, F., and S. Brandsberg-Dahl, 2005, The 2004 BP velocity benchmark., *in* 67th Annual Internat. Mtg., EAGE, Expanded Abstracts: EAGE, B035.

Brandsberg-Dahl, S., N. Chemingui, A. Valenciano, J. Ramos-Martinez, and Q. Lingyun, 2017, Fwi for model updates in large-contrast media: The Leading Edge, **36**, 81–87.

Charbonnier, P., L. Blanc-Feraud, G. Aubert, and M. Barlaud, 1997, Deterministic edge-preserving regularization in computed imaging: IEEE Transactions on Image Processing, **6**, 298–311.

Constable, S. C., R. L. Parker, and C. G. Constable, 1987, Occams inversion: A practical algorithm for generating smooth models from electromagnetic sounding data: Geophysics, **52**, 289–300.

Esser, E., L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann, 2016, Total-variation regularization strategies in full-waveform inversion: arXiv preprint arXiv:1608.06159.

Farquharson, C. G., and D. W. Oldenburg, July 1998, Non-linear inversion using general measures of data misfit and model structure: Geophysical Journal International, **134**, 213–227(15).

Geman, D., and G. Reynolds, 1992, Constrained restoration and the recovery of discontinuities: IEEE Transactions on Pattern Analysis and Machine Intelligence, **14**, 367–383.

Guitton, A., 2012, Blocky regularization schemes for full-waveform inversion: Geophysical Prospecting, **60**, 870–884.

Künsch, H. R., 1994, Robust priors for smoothing and image restoration: Ann. Inst. Statist. Math, **46**, 1–19.

Lailly, 1983, The seismic inverse problem as a sequence of before-stack migrations: Conference on Inverse Scattering: Theory and Application.

Nocedal, J., 1980, Updating Quasi-Newton matrices with limited storage: Mathematics of Computation, **35**, pp. 773–782.

Peters, B., and F. J. Herrmann, 2017, Constraints versus penalties for edge-preserving full-waveform inversion: The Leading Edge, **36**, 94–100.

Smithyman, B. R., B. Peters, and F. J. Herrmann, 2015, Constrained waveform inversion of colocated VSP and surface seismic data: Presented at the EAGE Annual Conference Proceedings. ((EAGE, Madrid)).

Stekl, I., and R. G. Pratt, 1998, Accurate viscoelastic modeling by frequency-domain finite differences using rotated operators: Geophysics, **63**, 1779–1794.

Tarantola, A., 1987, Inverse problem theory. methods for data fitting and model parameter estimation: Amsterdam: Elsevier.

Valenciano, A. A., M. Brown, A. Guitton, and M. D. Sacchi, 2005, Interval velocity estimation using edge-preserving regularization: SEG Technical Program Expanded Abstracts 2004, 2431–2434.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1–WCC26.

Youzwishen, C., and M. Sacchi, 2006, Edge preserving imaging: Journal of Seismic Exploration, **15**, 45.

Zhang, X., and J. Zhang, 2017, Model regularization for seismic traveltime tomography with an edge-preserving smoothing operator: Journal of Applied Geophysics, **138**, 143 – 153.

Zhdanov, M. S., 2002, Geophysical inverse theory and regularization problems, 1st ed.: Elsevier Science.

Time-domain elastic Gauss-Newton full-waveform inversion via matrix-free adjoint-state method

Ke Chen[1]

Abstract

A time-domain matrix-free elastic Gauss-Newton FWI algorithm is formulated based on elastic least-squares reverse time migration (LSRTM) algorithm. We consider the elastic Gauss-Newton FWI as an iterative elastic LSRTM problem. The proposed algorithm consists of two loops of iterations: the outer Gauss-Newton nonlinear iterations and the inner conjugate gradient least-squares (CGLS) linear iterations. The Gauss-Newton search direction in each outer FWI iteration is computed using the matrix-free CGLS algorithm. This step is equivalent to apply an elastic LSRTM on data residual, with the Jacobian operator as elastic Born modeling operator and the adjoint of Jacobian operator as elastic RTM operator. The CGLS algorithm can be safely used for solving the Gauss-Newton search direction because our discretized numerical versions of elastic Born and RTM operators passed the dot-product test. The inner CGLS linear iterations are preconditioned using the elastic pseudo-Hessian operator. Our algorithm is matrix-free that only requires the forward Jacobian and adjoint Jacobian operator applied "on the fly" to vectors. The operators are applied on vectors efficiently via the adjoint-state method. We use the proposed algorithm to simultaneously invert for P- and S-wave velocities.The proposed elastic Gauss-Newton FWI generates better inverted models than the nonlinear conjugate gradient method based elastic FWI. More importantly, the elastic Gauss-Newton FWI can decouple the crosstalk between P- and S-wave velocity models, which is an important problem in multiparameter FWI.

[1] Email: `ke7@ualberta.ca`

9.1 Introduction

Full-waveform inversion (FWI) (Bamberger et al., 1982; Lailly, 1983; Tarantola, 1984; Virieux and Operto, 2009) aims at estimating subsurface model parameters using the full-wave mode data recorded by seismic receivers. It is a nonlinear and ill-posed inverse problem. Tarantola (1984) formulates time-domain acoustic FWI as a nonlinear least-squares inverse problem (Tarantola and Valette, 1982). The inverse problem was proposed to be solved using the iterative method. Tarantola (1984) adopts the adjoint-state method (Lions, 1971) to compute the gradient of the objective function to avoid the expensive explicit computation of the Fréchet derivative (Bamberger et al., 1982; Lailly, 1983). Gauthier et al. (1986) conduct the first numerical study of 2D acoustic FWI in time domain on synthetic seismic reflection and transmission data. Tarantola (1986) extends the FWI theory to isotropic elastic case and analyzes the radiation patterns of point diffractors for different parameters. For inverting reflection seismic data, Tarantola (1986) suggests first optimizing the P-wave impedance, second optimizing the S-wave impedance, and then optimizing the density. Mora (1987b, 1988) numerically study the 2D time-domain elastic FWI of synthetic reflection and transmission multicomponent data. Crase et al. (1990) propose a robust time-domain elastic FWI based on ℓ_1 norm or Cauchy criterion minimization and apply it on real marine streamer data. More recent, 2D time-domain elastic FWI has been applied to marine seismic streamer data (Shipp and Singh, 2002) and multicomponent OBC seismic data (Sears et al., 2008, 2010). Kohn et al. (2012) investigate the influence of model parameterization in 2D time-domain elastic full-waveform inversion. Matharu and Sacchi (2017) discuss the encoded source time-domain elastic FWI. Time-domain elastic FWI has also been extended to three dimensional (Epanomeritakis et al., 2008; Guasch, 2011; Vigh et al., 2014; Borisov and Singh, 2015; Raknes et al., 2015; Albertin et al., 2016). However, it is still computational challenging even on modern high-performance computer cluster.

To mitigate the crosstalk among multiparameters in elastic FWI, several strategies have been introduced recently. Wang et al. (2016) propose to use the block-diagonal pseudo-Hessian (Jin et al., 1992) for preconditioning the gradient in elastic FWI. Ren and Liu (2016) and Wang and Cheng (2017) propose to use wavenumber-domain wavefield decomposition (Zhang and McMechan, 2010) to separate the P- and S-wave field before gradient computing to reduce the crosstalk in elastic FWI. Pan et al. (2017) propose to approximately estimate the crosstalks and subtract them from the gradient.

In this paper, we propose a time-domain matrix-free elastic Gauss-Newton FWI algorithm based on elastic least-squares reverse time migration (LSRTM) algorithm (Chen and Sacchi, 2017). We chose to adopt the first-order velocity-stress elastic wave-equation system (Virieux, 1986) for forward modeling and the inversion is parameterized in terms of P- and S-wave velocities. The proposed algorithm consists of two loops of iterations: the outer Gauss-Newton nonlinear iterations (Nocedal and Wright, 2006) and the inner conjugate gradient least-squares (CGLS) linear iterations (Hestenes and Stiefel, 1952; Paige and Saunders, 1982). The outer nonlinear iteration uses parabola fitting line search (Vigh et al., 2009) to estimate the step size. The Gauss-Newton search direction in each outer FWI iteration is computed using the matrix-free CGLS algorithm. We recognize that this step is actually equivalent to apply an elastic LSRTM on data residual (Chen and Sacchi, 2017), with the Jacobian operator as elastic Born modeling operator and the adjoint of Jacobian operator as elastic RTM operator. The CGLS algorithm can be safely used for solving the Gauss-Newton search direction because our discretized numerical versions of elastic Born and RTM operators passed the dot-product test (Mora, 1987a; Claerbout, 1992). In the inner CGLS linear iterations, the step size is analytically calculated without of the need of line search. The inner CGLS linear iterations are preconditioned using the elastic pseudo-Hessian operator (Shin et al., 2001a; Chen and Sacchi, 2017). Our algorithm is matrix-free that only requires the forward Jacobian and adjoint Jacobian operator applied "on the fly" to vectors. The operators are applied on vectors efficiently via the adjoint-state method (Lions, 1971). We use the proposed algorithm to simultaneously invert for P- and S-wave velocities. The proposed elastic Gauss-Newton FWI generates better inverted models than the nonlinear conjugate gradient method based elastic FWI. More importantly, the elastic Gauss-Newton FWI can decouple the crosstalk between P- and S-wave velocity models.

The first discussion of Gauss-Newton method for time-domain FWI problem can be traced back to Tarantola (1984) where they call it the "total inversion" method (Tarantola and Valette, 1982). Time-domain elastic Gauss-Newton FWI has also been investigated in Sheen et al. (2006) and Epanomeritakis et al. (2008). However, both Sheen et al. (2006) and Epanomeritakis et al. (2008) do not consider the multiparameter crosstalk problem. Epanomeritakis et al. (2008) only inverted for the shear modulus. In Sheen et al. (2006), the Jacobian matrix and Hessian matrix are explicitly using the reciprocity of the Green's function (Tarantola, 1988; Shin et al., 2001b), which is too expensive for large-scale problem. Recently, Pan et al. (2016) discuss the Gauss-Newton and full-Newton method in elastic HTI media. However, it is also based on explicitly computing and inverting the Hessian operator which impedes the application to realistic scale problem.

This paper is organized as follows: First, we describe the system of equations that we have adopted to forward model elastic wavefields. Then, we discuss the general time-domain elastic Gauss-Newton FWI algorithm. Subsequently, we propose to formulate the step of solving Gauss-Newton search direction as an elastic LSRTM problem. We discuss the elastic Born and RTM operators and the numerical adjointness of those operators. Furthermore, we propose to solve the Gauss-Newton search direction using the CGLS algorithm and also propose to precondition the CGLS algorithm by elastic pseudo-Hessian. Then, we summarize the time-domain matrix-free elastic Gauss-Newton FWI algorithm. In the last section, we provide numerical examples that permit us to evaluate the performance of the proposed algorithm.

9.2 Theory

9.2.1 Time-domain, heterogeneous, isotropic elastic wave equation

We assume a heterogeneous, isotropic elastic earth media. The propagation of seismic wave is governed by the elastic wave equation (Virieux, 1986)

$$\begin{aligned}
&\rho\frac{\partial v_x}{\partial t}-\left(\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xz}}{\partial z}\right)=0,\\
&\rho\frac{\partial v_z}{\partial t}-\left(\frac{\partial \sigma_{xz}}{\partial x}+\frac{\partial \sigma_{zz}}{\partial z}\right)=0,\\
&\frac{\partial \sigma_{xx}}{\partial t}-(\lambda+2\mu)\frac{\partial v_x}{\partial x}-\lambda\frac{\partial v_z}{\partial z}=f_{xx},\\
&\frac{\partial \sigma_{zz}}{\partial t}-(\lambda+2\mu)\frac{\partial v_z}{\partial z}-\lambda\frac{\partial v_x}{\partial x}=f_{zz},\\
&\frac{\partial \sigma_{xz}}{\partial t}-\mu\left(\frac{\partial v_x}{\partial z}+\frac{\partial v_z}{\partial x}\right)=0,
\end{aligned} \tag{9.1}$$

with zero initial condition and appropriate boundary conditions. In the above equation, v_x and v_z are the horizontal and vertical particle velocity fields, σ_{xx}, σ_{xz} and σ_{zz} are the stress fields. Similarly, ρ indicates density, λ and μ the Lamé parameters, f_{xx} and f_{zz} are the explosive source terms. In the wave equation, we dropped the dependence on spatial and temporal coordinates $\mathbf{x}$ and t of our variables to make the notations concise but we understand that $v_x = v_x(\mathbf{x}, t)$, $\lambda = \lambda(\mathbf{x})$, etc. The elastic wave equation is the state equation of the elastic parameter inversion problem when it is regarded as optimal control problem (Lions, 1971; Plessix, 2006). Abstractly, the elastic wave equation 9.1 can be written in functional form as follows

$$\mathscr{S}(\mathbf{m})\mathbf{u} = \mathbf{f}, \tag{9.2}$$

where $\mathbf{u} = (v_x, v_z, \sigma_{xx}, \sigma_{zz}, \sigma_{xz})^T$ is the wavefield vector in space $\mathbb{U}$, $\mathbf{f} = (0, 0, f_{xx}, f_{zz}, 0)^T$ is the source vector in space $\mathbb{F}$, $\mathscr{S}(\mathbf{m})$ ($\mathscr{S} : \mathbb{U} \to \mathbb{F}$) is the wave equation operator with initial conditions and boundary conditions, and $\mathbf{m} = (\rho, \lambda, \mu)^T$ denotes the model parameter vector in space $\mathbb{M}$. The wavefield $\mathbf{u}$ is linear in the source term $\mathbf{f}$ but is nonlinear in the model $\mathbf{m}$. If the source term is assumed known, $\mathbf{u}$ can be regarded as a nonlinear function of the model parameters $\mathbf{m}$

$$\mathbf{u} = \mathbf{u}(\mathbf{m}). \tag{9.3}$$

In a general heterogenous media, there is no analytic solution for $\mathbf{u}$ given $\mathbf{m}$. A numerical method must be used to solve the forward problem. In this article, a time domain staggered-grid finite-difference (FD) scheme (Virieux, 1986; Levander, 1988) is utilized to discretize the continuous form elastic wave equation 9.1. The unsplit Convolutional Perfectly Matched Layer (C-PML) method is used to absorb incident waves on artificial computational boundaries (Komatitsch and Martin, 2007). In our FD code, we adopted a second-order centered difference scheme in time and a selectable order staggered difference scheme in space.

9.2.2 Time-domain elastic Gauss-Newton full-waveform inversion

FWI estimates the spatially varying coefficients of the wave equation from the seismic data observed on the surface of the earth. FWI is usually formulated as a nonlinear least-squares inverse problem that minimizes the functional (Tarantola, 1984, 1986)

$$\mathscr{J}(\mathbf{m}) = \frac{1}{2}\sum_{i=1}^{N_s} \|\mathbf{d}_i(\mathbf{m}) - \mathbf{d}_i^{obs}\|_2^2, \tag{9.4}$$

where $\mathbf{d}_i(\mathbf{m}) = \mathscr{T}\mathbf{u}_i(\mathbf{m})$ is the forward modeled seismic data for ith shot, $\mathscr{T}$ is the sampling operator, $\mathbf{d}_i^{obs}$ is the observed seismic data for ith shot, and $\|\cdot\|_2$ denotes the ℓ_2 norm of vector. Gradient-based FWI is a local optimization that utilizes the Born approximation in each iteration (Virieux and Operto, 2009). Introducing a model perturbation $\mathbf{m} \to \mathbf{m} + \delta\mathbf{m}$ and a second-order Taylor series expansion, the cost function in the vicinity of $\mathbf{m}$ is given by the following expression

$$\mathscr{J}(\mathbf{m} + \delta\mathbf{m}) = \mathscr{J}(\mathbf{m}) + \frac{\partial \mathscr{J}(\mathbf{m})}{\partial \mathbf{m}}\delta\mathbf{m} + \frac{1}{2}\delta\mathbf{m}^T \frac{\partial^2 \mathscr{J}(\mathbf{m})}{\partial \mathbf{m}^2}\delta\mathbf{m} + \mathscr{O}(\|\delta\mathbf{m}\|^3). \tag{9.5}$$

In the vicinity of $\mathbf{m}$, the cost function is linearized and an optimal model update $\delta\mathbf{m}$ should satisfy $\partial \mathscr{J}(\mathbf{m}+\delta\mathbf{m})/\partial\delta\mathbf{m}=0$. The latter results in the following expression

$$\begin{aligned}\delta\mathbf{m} &= -\left[\frac{\partial^2\mathscr{J}(\mathbf{m})}{\partial\mathbf{m}^2}\right]^{-1}\frac{\partial\mathscr{J}(\mathbf{m})}{\partial\mathbf{m}} \\ &= -\left\{\sum_{i=1}^{N_s}\left[\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)+\left(\frac{\partial^2\mathbf{d}_i}{\partial\mathbf{m}^2}\right)^{\dagger}\left(\mathbf{d}_i-\mathbf{d}_i^{obs}\right)\right]\right\}^{-1}\left[\sum_{i=1}^{N_s}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\mathbf{d}_i-\mathbf{d}_i^{obs}\right)\right],\end{aligned} \tag{9.6}$$

where † denotes the adjoint of an operator, $\partial\mathbf{d}_i/\partial\mathbf{m}$ is the Fréchet derivative or Jacobian operator of $\mathbf{d}$ with respect to $\mathbf{m}$, and the latter is actually the elastic Born approximation operator. In the above equation, the term in the braces is called the full Hessian in the FWI problem (Pratt et al., 1998; Fichtner and Trampert, 2011; Metivier et al., 2013). The second term in the braces corresponds to the second-order multiple scattering. This term is small when the cost function is close to a minimum. Dropping the second-order term results in the Gauss-Newton update (Pratt et al., 1998)

$$\delta\mathbf{m} = -\left[\sum_{i=1}^{N_s}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)\right]^{-1}\left[\sum_{i=1}^{N_s}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\mathbf{d}_i-\mathbf{d}_i^{obs}\right)\right], \tag{9.7}$$

where the term in the first bracket is the called approximated Hessian

$$\mathscr{H} = \sum_{i=1}^{N_s}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right), \tag{9.8}$$

and the term in the second bracket is the gradient

$$\mathscr{G} = \sum_{i=1}^{N_s}\left(\frac{\partial\mathbf{d}_i}{\partial\mathbf{m}}\right)^{\dagger}\left(\mathbf{d}_i-\mathbf{d}_i^{obs}\right). \tag{9.9}$$

With equations 9.8 and 9.9, the Gauss-Newton update 9.7 can be written as

$$\delta\mathbf{m} = -\mathscr{H}^{-1}\mathscr{G}. \tag{9.10}$$

The elastic Gauss-Newton FWI iteratively minimizes the nonlinear least-squares objective function by updating the model in the Gauss-Newton direction $\delta\mathbf{m}$. The updated model at $(k+1)$th iteration can be written as

$$\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} + \eta^{(k)}\delta\mathbf{m}^{(k)} = \mathbf{m}^{(k)} - \eta^{(k)}(\mathscr{H}^{(k)})^{-1}\mathscr{G}^{(k)}, \tag{9.11}$$

where $\mathbf{m}^{(k)}$ is the model at kth iteration, $\eta^{(k)}$ is the step size at kth iteration, and $\delta\mathbf{m}^{(k)}$ is the Gauss-Newton search direction at kth iteration, $\mathscr{H}^{(k)}$ is the approximated Hessian at kth iteration and $\mathscr{G}^{(k)}$ is the gradient at kth iteration. The general time-domain elastic Gauss-Newton FWI algorithm is summarized as Algorithm 9.1

Algorithm 9.1 Time-domain elastic Gauss-Newton FWI

```
    **Initialize**: $\mathbf{m}^{(0)}$
    **for** $k = 0, 1, \cdots$ **while** not converge **do**
        1. Forward modeling: $\mathbf{d}_i^{(k)} = \mathbf{d}_i(\mathbf{m}^{(k)})$, $i = 1, \cdots, N_s$
        2. Compute data residual: $\delta\mathbf{d}_i^{(k)} = \mathbf{d}_i^{obs} - \mathbf{d}_i^{(k)}$, $i = 1, \cdots, N_s$
        3. Compute Gauss-Newton search direction: $\delta\mathbf{m}^{(k)} = -\left(\mathscr{H}^{(k)}\right)^{-1}\mathscr{G}^{(k)}$
        4. Compute step size $\eta^{(k)}$ via parabola fitting line search.
        5. $\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} + \eta^{(k)}\delta\mathbf{m}^{(k)}$
    **end**
```

9.2.3 Solve for Gauss-Newton step via CGLS algorithm combing with adjoint-state method

Formulate Gauss-Newton step as an elastic LSRTM problem

In this section, we will drop the FWI iteration index (k) of the approximated Hessian, gradient and model update to make the notations concise but we understand that they depend on the FWI iteration index. In the definition of Gauss-Newton update (equation 9.11), the approximated Hessian $\mathscr{H}$ needs to be inverted. However, explicitly computing and inverting the Hessian operator are prohibitively expensive for realistic scale problem. Instead, we solve for the Gauss-Newton step update via the matrix-free conjugate gradient least-squares algorithm combing with adjoint-state method. Gauss-Newton update 9.7 can be computed via solving normal equations

$$\sum_{i=1}^{N_s} \left(\frac{\partial \mathbf{d}_i}{\partial \mathbf{m}}\right)^\dagger \left(\frac{\partial \mathbf{d}_i}{\partial \mathbf{m}}\right) \delta\mathbf{m} = -\sum_{i=1}^{N_s} \left(\frac{\partial \mathbf{d}_i}{\partial \mathbf{m}}\right)^\dagger \left(\mathbf{d}_i - \mathbf{d}_i^{obs}\right), \tag{9.12}$$

or equivalently,

$$\mathscr{H}\delta\mathbf{m} = -\mathscr{G}. \tag{9.13}$$

If we use $\mathscr{L}_i$ to denote the Jacobian operator $\partial \mathbf{d}_i/\partial \mathbf{m}$ and $\delta \mathbf{d}_i$ to denote the data residual $\mathbf{d}_i^{obs} - \mathbf{d}_i$ for ith shot, the normal equation 9.12 becomes

$$\sum_{i=1}^{N_s} \mathscr{L}_i^{\dagger} \mathscr{L}_i \delta \mathbf{m} = \sum_{i=1}^{N_s} \mathscr{L}_i^{\dagger} \delta \mathbf{d}_i. \tag{9.14}$$

The solution of above equation is equivalent to solving a least-squares inversion problem that minimizes

$$\mathscr{J}_{gn}(\delta \mathbf{m}) = \frac{1}{2} \sum_{i=1}^{N_s} \| \mathscr{L}_i \delta \mathbf{m} - \delta \mathbf{d}_i \|_2^2. \tag{9.15}$$

We recognized that solving for the Gauss-Newton step (equation 9.15) is actually equivalent to solve an elastic LSRTM problem (Chen and Sacchi, 2017). The Jacobian operator $\mathscr{L}_i$ is the elastic Born modeling operator and the adjoint of Jacobian operator $\mathscr{L}_i^{\dagger}$ is the elastic RTM operator.

Jacobian operator $\mathscr{L}$: elastic Born modeling operator

The elastic Born modeling or Born approximation describes that the incident wavefield hits the scatters and generates the scattered wavefield. It maps from model perturbation to data perturbation. The incident background wavefield for current model is computed using the elastic wave equation

$$\begin{aligned}
&\rho \frac{\partial v_x}{\partial t} - \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z} \right) = 0, \\
&\rho \frac{\partial v_z}{\partial t} - \left(\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z} \right) = 0, \\
&\frac{\partial \sigma_{xx}}{\partial t} - (\lambda + 2\mu) \frac{\partial v_x}{\partial x} - \lambda \frac{\partial v_z}{\partial z} = f_{xx}, \\
&\frac{\partial \sigma_{zz}}{\partial t} - (\lambda + 2\mu) \frac{\partial v_z}{\partial z} - \lambda \frac{\partial v_x}{\partial x} = f_{zz}, \\
&\frac{\partial \sigma_{xz}}{\partial t} - \mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) = 0,
\end{aligned} \tag{9.16}$$

with zero initial condition and appropriate boundary conditions. A perturbation of the model parameters

$$\rho \rightarrow \rho + \delta \rho, \tag{9.17a}$$

$$\lambda \rightarrow \lambda + \delta \lambda, \tag{9.17b}$$

$$\mu \rightarrow \mu + \delta \mu, \tag{9.17c}$$

leads to a perturbation of the wavefields

$$v_x \rightarrow v_x + \delta v_x, \tag{9.17d}$$

$$v_z \rightarrow v_z + \delta v_z, \tag{9.17e}$$

$$\sigma_{xx} \rightarrow \sigma_{xx} + \delta \sigma_{xx}, \tag{9.17f}$$

$$\sigma_{zz} \rightarrow \sigma_{zz} + \delta \sigma_{zz}, \tag{9.17g}$$

$$\sigma_{xz} \rightarrow \sigma_{xz} + \delta \sigma_{xz}. \tag{9.17h}$$

Inserting equation 9.17 into equation 9.16, subtracting equation 9.16, and dropping second and higher order terms leads to the Born approximation for the first-order velocity stress elastic wave equation system (Chen and Sacchi, 2017)

$$\begin{aligned}
&\rho \frac{\partial \delta v_x}{\partial t} - \left(\frac{\partial \delta \sigma_{xx}}{\partial x} + \frac{\partial \delta \sigma_{xz}}{\partial z} \right) = -\delta \varrho \dot{v}_x, \\
&\rho \frac{\partial \delta v_z}{\partial t} - \left(\frac{\partial \delta \sigma_{xz}}{\partial x} + \frac{\partial \delta \sigma_{zz}}{\partial z} \right) = -\delta \varrho \dot{v}_z, \\
&\frac{\partial \delta \sigma_{xx}}{\partial t} - (\lambda + 2\mu) \frac{\partial \delta v_x}{\partial x} - \lambda \frac{\partial \delta v_z}{\partial z} = (\delta \lambda + \delta \mu) \frac{\dot{\sigma}_{xx} + \dot{\sigma}_{zz}}{2(\lambda + \mu)} + \delta \mu \frac{\dot{\sigma}_{xx} - \dot{\sigma}_{zz}}{2\mu}, \\
&\frac{\partial \delta \sigma_{zz}}{\partial t} - (\lambda + 2\mu) \frac{\partial \delta v_z}{\partial z} - \lambda \frac{\partial \delta v_x}{\partial x} = (\delta \lambda + \delta \mu) \frac{\dot{\sigma}_{xx} + \dot{\sigma}_{zz}}{2(\lambda + \mu)} - \delta \mu \frac{\dot{\sigma}_{xx} - \dot{\sigma}_{zz}}{2\mu}, \\
&\frac{\partial \delta \sigma_{xz}}{\partial t} - \mu \left(\frac{\partial \delta v_x}{\partial z} + \frac{\partial \delta v_z}{\partial x} \right) = \delta \mu \frac{\dot{\sigma}_{xz}}{\mu},
\end{aligned} \tag{9.18}$$

with zero initial condition and appropriate boundary conditions. The vector $\delta \mathbf{u} = (\delta v_x, \delta v_z, \delta \sigma_{xx}, \delta \sigma_{zz}, \delta \sigma_{xz})^T$ is the scattered wavefield due to model perturbations $\delta \varrho$, $\delta \lambda$ and $\delta \mu$, and $\mathbf{u} = (v_x, v_z, \sigma_{xx}, \sigma_{zz}, \sigma_{xz})^T$ is the incident wavefield field in the background model ρ, λ, μ, and the over-dot means the time derivative. The right side of equation 9.18 is the so-called "secondary source". The scattered data is obtained by sampling the scattered wavefield at the receiver positions $\delta \mathbf{d} = \mathscr{T} \delta \mathbf{u}$. For elastic FWI problem, parameterization in wave speeds is better than parameterization in Lamé parameters (Tarantola, 1986). The model perturbations have the relationship

$$\begin{pmatrix} \delta \varrho \\ \delta \lambda \\ \delta \mu \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ V_p^2 - 2V_s^2 & 2\rho V_p & -4\rho V_s \\ V_s^2 & 0 & 2\rho V_s \end{pmatrix} \begin{pmatrix} \delta \rho \\ \delta V_p \\ \delta V_s \end{pmatrix}, \tag{9.19}$$

where $\mathbf{m} = (\rho, V_p, V_s)^T$ is the background model and $\delta \mathbf{m} = (\delta \rho, \delta V_p, \delta V_s)^T$ is the model perturbation.

Adjoint of Jacobian operator $\mathscr{L}^\dagger$: elastic RTM operator

The elastic RTM operator is the adjoint of the elastic Born modeling operator that maps from data perturbation to model perturbation. The adjoint-state equation for state equation 9.1 can be derived using the adjoint-state method (Chen and Sacchi, 2017)

$$\begin{aligned}
&-\rho\frac{\partial \upsilon_x}{\partial t} + \left(\frac{\partial \varsigma_{xx}}{\partial x} + \frac{\partial \varsigma_{xz}}{\partial z}\right) = \delta d_{\upsilon_x},\\
&-\rho\frac{\partial \upsilon_z}{\partial t} + \left(\frac{\partial \varsigma_{xz}}{\partial x} + \frac{\partial \varsigma_{zz}}{\partial z}\right) = \delta d_{\upsilon_z},\\
&-\frac{\partial \varsigma_{xx}}{\partial t} + (\lambda + 2\mu)\frac{\partial \upsilon_x}{\partial x} + \lambda\frac{\partial \upsilon_z}{\partial z} = 0,\\
&-\frac{\partial \varsigma_{zz}}{\partial t} + (\lambda + 2\mu)\frac{\partial \upsilon_z}{\partial z} + \lambda\frac{\partial \upsilon_x}{\partial x} = 0,\\
&-\frac{\partial \varsigma_{xz}}{\partial t} + \mu\left(\frac{\partial \upsilon_x}{\partial z} + \frac{\partial \upsilon_z}{\partial x}\right) = 0,
\end{aligned} \tag{9.20}$$

with zero final condition and appropriate boundary conditions. In the adjoint-state equation, vector $\mathbf{p} = (\upsilon_x, \upsilon_z, \varsigma_{xx}, \varsigma_{zz}, \varsigma_{xz})^T$ is the adjoint-state wavefield (back-propagated wavefield), vector $\delta\mathbf{d} = (\delta d_{\upsilon_x}, \delta d_{\upsilon_z})^T$ is the data residual. Here, we assume that the observed data are two-component vector particle velocity data. The adjoint model perturbations ($\delta\mathbf{m}^* = (\delta\varrho^*, \delta\lambda^*, \delta\mu^*)^T$) can be expressed as (Chen and Sacchi, 2017)

$$\begin{aligned}
\delta\varrho^* &= -\int(\dot{v}_x\upsilon_x + \dot{v}_z\upsilon_z)dt,\\
\delta\lambda^* &= \int\frac{(\dot{\sigma}_{xx} + \dot{\sigma}_{zz})(\varsigma_{xx} + \varsigma_{zz})}{4(\lambda+\mu)^2}dt,\\
\delta\mu^* &= \int\left[\frac{\dot{\sigma}_{xz}\varsigma_{xz}}{\mu^2} + \frac{(\dot{\sigma}_{xx} + \dot{\sigma}_{zz})(\varsigma_{xx} + \varsigma_{zz})}{4(\lambda+\mu)^2} + \frac{(\dot{\sigma}_{xx} - \dot{\sigma}_{zz})(\varsigma_{xx} - \varsigma_{zz})}{4\mu^2}\right]dt,
\end{aligned} \tag{9.21}$$

where the over-dot means time derivative. In elastic FWI, parameterization in wave speeds is a better than parameterization in Lamé parameters. We use the following parameter transformation

$$\begin{pmatrix}\delta\rho^*\\ \delta V_p^*\\ \delta V_s^*\end{pmatrix} = \begin{pmatrix}1 & V_p^2 - 2V_s^2 & V_s^2\\ 0 & 2\rho V_p & 0\\ 0 & -4\rho V_s & 2\rho V_s\end{pmatrix}\begin{pmatrix}\delta\varrho^*\\ \delta\lambda^*\\ \delta\mu^*\end{pmatrix}, \tag{9.22}$$

to transform the Lamé parameters perturbations to wave speeds perturbations.

Solving Gauss-Newton step using preconditioned CGLS algorithm

It is important to precondition the linear system of equations to accelerate the convergence of CGLS algorithm and stabilize the outer FWI iterations. We adopted the elastic pseudo-Hessian for preconditioning (Chen and Sacchi, 2017). The preconditioned version of elastic LSRTM minimizes

$$\mathscr{J}_{gn}(\delta\tilde{\mathbf{m}}) = \frac{1}{2}\sum_{i=1}^{N_s} \|\mathscr{L}_i \mathscr{P}_H \delta\tilde{\mathbf{m}} - \delta\mathbf{d}_i\|_2^2, \tag{9.23}$$

where $\mathscr{P}_H$ denotes the inverse of the diagonal of pseudo-Hessian. The preconditioned conjugate gradient least squares (PCGLS) algorithm (Bjorck, 1996) can be summarized as Algorithm 9.2. The output of Algorithm 9.2 is the inverted model perturbations $\delta\mathbf{m} = \mathscr{P}_H\delta\tilde{\mathbf{m}} = (\delta V_p, \delta V_s)^T$, which is the Gauss-Newton search direction.

Algorithm 9.2 Preconditioned CGLS algorithm

```
Initialize
    $\delta\mathbf{m}^{(0)} = \mathbf{0}$
    $\mathbf{r}_i^{(0)} = \delta\mathbf{d}_i,\ i = 1, \cdots, N_s$
    $\mathbf{s}^{(0)} = \mathscr{P}_H{}^{\dagger}\left(\sum_{i=1}^{N_s} \mathscr{L}_i^{\dagger}\mathbf{r}_i^{(0)}\right)$
    $\mathbf{p}^{(0)} = \mathbf{s}^{(0)}$
    $\gamma^{(0)} = \|\mathbf{s}^{(0)}\|_2^2$
for $l = 0, 1, \cdots$ while not converge do
    $\mathbf{t}^{(l)} = \mathscr{P}_H\mathbf{p}^{(l)},$
    $\mathbf{q}_i^{(l)} = \mathscr{L}_i\mathbf{t}^{(l)},\ i = 1, \cdots, N_s$
    $\delta^{(l)} = \sum_{i=1}^{N_s} \|\mathbf{q}_i^{(l)}\|_2^2$
    $\alpha^{(l)} = \gamma^{(l)}/\delta^{(l)}$
    $\delta\mathbf{m}^{(l+1)} = \delta\mathbf{m}^{(l)} + \alpha^{(l)}\mathbf{t}^{(l)}$
    $\mathbf{r}_i^{(l+1)} = \mathbf{r}_i^{(l)} - \alpha^{(l)}\mathbf{q}_i^{(l)},\ i = 1, \cdots, N_s$
    $\mathbf{s}^{(l+1)} = \mathscr{P}_H{}^{\dagger}\left(\sum_{i=1}^{N_s} \mathscr{L}_i^{\dagger}\mathbf{r}_i^{(l+1)}\right)$
    $\gamma^{(l+1)} = \|\mathbf{s}^{(l+1)}\|_2^2$
    $\beta^{(l)} = \gamma^{(l+1)}/\gamma^{(l)}$
    $\mathbf{p}^{(l+1)} = \mathbf{s}^{(l+1)} + \beta^{(l)}\mathbf{p}^{(l)}$
end
```

9.2.4 Time-domain matrix-free elastic Gauss-Newton FWI

As discussed in the previous section, we propose to do a time-domain elastic Gauss-Newton FWI using CGLS algorithm to solve the internal Gauss-Newton linear system of equation. It does not need to formulate or invert the Hessian matrix explicitly. Instead, the Hessian is iteratively inverted by the CGLS iterations. The CGLS only requires two operators $\mathscr{L}_i$ and $\mathscr{L}_i^\dagger$ that are applied "on the fly" to vectors. The operators are applied on vectors efficiently via the adjoint-state method. We call this algorithm as time-domain matrix-free elastic Gauss-Newton FWI. It can be summarized as Algorithm 9.3.

Algorithm 9.3 Time-domain matrix-free elastic Gauss-Newton FWI

```
Initialize: $\mathbf{m}^{(0)}$
for $k = 0, 1, \cdots$ while not converge do
    1. Forward modeling: $\mathbf{d}_i^{(k)} = \mathbf{d}_i(\mathbf{m}^{(k)})$, $i = 1, \cdots, N_s$
    2. Compute data residual: $\delta\mathbf{d}_i^{(k)} = \mathbf{d}_i^{obs} - \mathbf{d}_i^{(k)}$, $i = 1, \cdots, N_s$
    3. Compute Gauss-Newton search direction $\delta\mathbf{m}^{(k)}$ by solving
       $$\min_{\delta\mathbf{m}^{(k)}} \ \frac{1}{2}\sum_{i=1}^{N_s} \|\mathscr{L}_i^{(k)}\mathscr{P}_H^{(k)}\delta\mathbf{m}^{(k)} - \delta\mathbf{d}_i^{(k)}\|_2^2$$
       using preconditioned CGLS algorithm 9.2.
    4. Compute step size $\eta^{(k)}$ via parabola fitting line search.
    5. $\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} + \eta^{(k)}\delta\mathbf{m}^{(k)}$
end
```

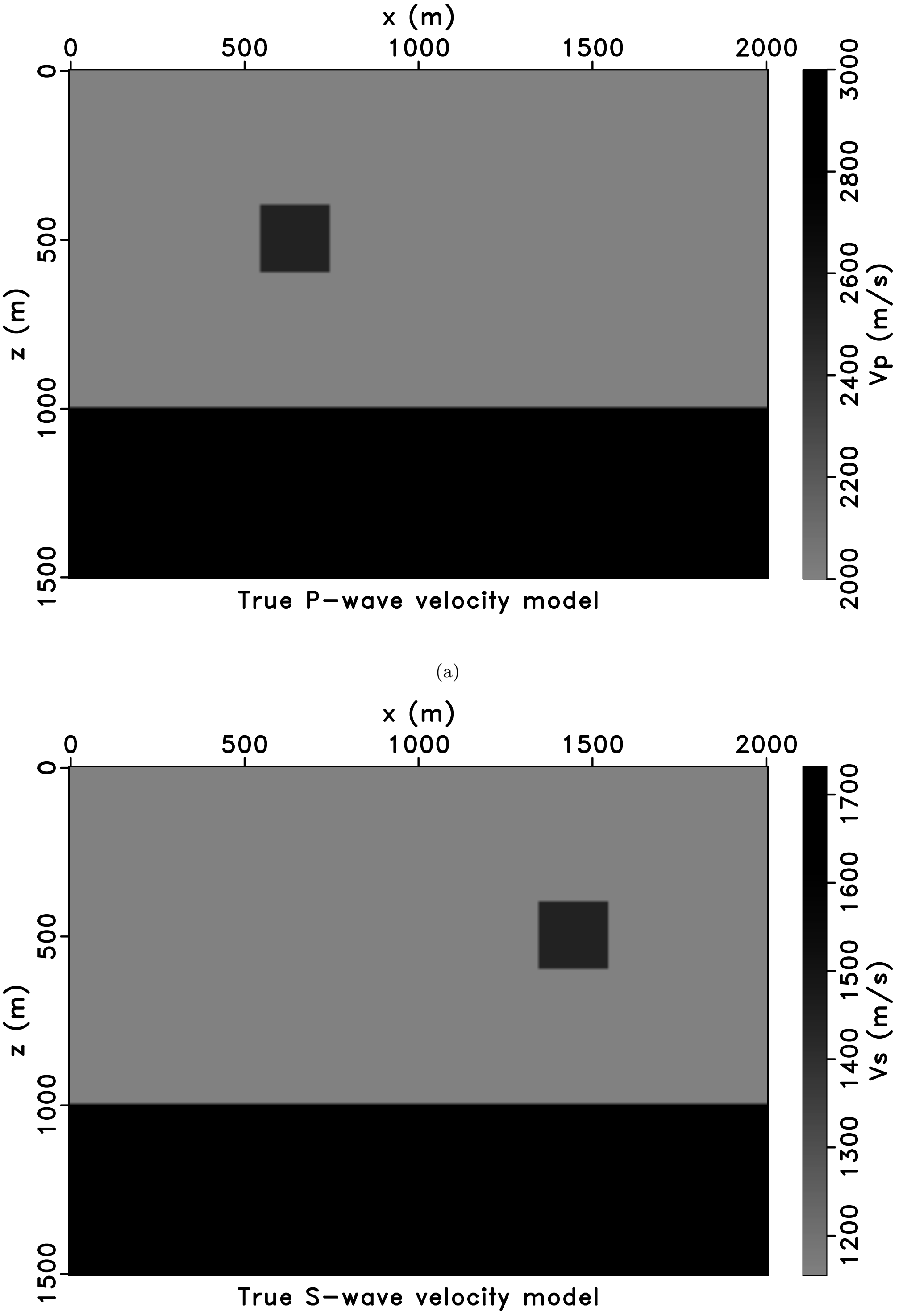
9.3 Examples

The proposed method was tested on the elastic inclusion model and the elastic Marmousi2 model (Martin et al., 2006). The “observed data” are generated with time-domain elastic staggered-grid finite-difference method. The C-PML boundary condition was applied on four boundaries of the model. The observed data were assumed to be vector particle velocity fields. At current stage, we simultaneously invert for P- and S- wave velocities and assume that the density is known. The code for our numerical examples was written in C and parallelized with Message Passing Interface (MPI) over shots.

9.3.1 Elastic inclusion model

Figure 9.1 shows the true P- and S-wave velocity models. The square velocity anomalies are embedded in two layered models. Density is assumed to be constant (2000 kg/m^3). The model has a dimension of 2 km in horizontal axis and 1.5 km in depth with 201 × 151 grid points. There are 51 shots and 201 receivers deployed along the surface that simulates a fixed-spread survey geometry. The shot interval is 40 m and receiver interval is 10 m. The shot depth is 10 m and the receiver depth is 20 m. A 10 Hz central frequency Ricker wavelet is used to simulate an explosive source. The multicomponent observed data are simulated using our elastic finite-difference code. The observed data cubes are shown in Figure 9.2. In this section, we compared the results of conventional elastic FWI based on nonlinear conjugate gradient method (hereafter, we call it elastic NLCG FWI for short) and the proposed elastic Gauss-Newton FWI. In the elastic NLCG FWI, we used elastic pseudo-Hessian to precondition the gradient. And the line search is also based on parabola fitting. Both the elastic NLCG FWI and the elastic Gauss-Newton FWI utilize multiscale method for inversion from the low frequency until the high frequency bands of the data (Bunks et al., 1995). The inverted models from early scale are used as initial models for inversion in later scale. In other words, there is actually another extra loop over frequency bands outside of the Algorithm 9.3. There are in total three loops of iterations for our elastic Gauss-Newton FWI algorithm. The four frequency bands for multiscale inversion is: 0-5 Hz, 0-10 Hz, 0-15 Hz and 0-30 Hz. Figure 9.3 shows the starting P- and S-wave velocity models for elastic NLCG FWI and elastic Gauss-Newton FWI. The initial models contain two layers without of the velocity anomalies.

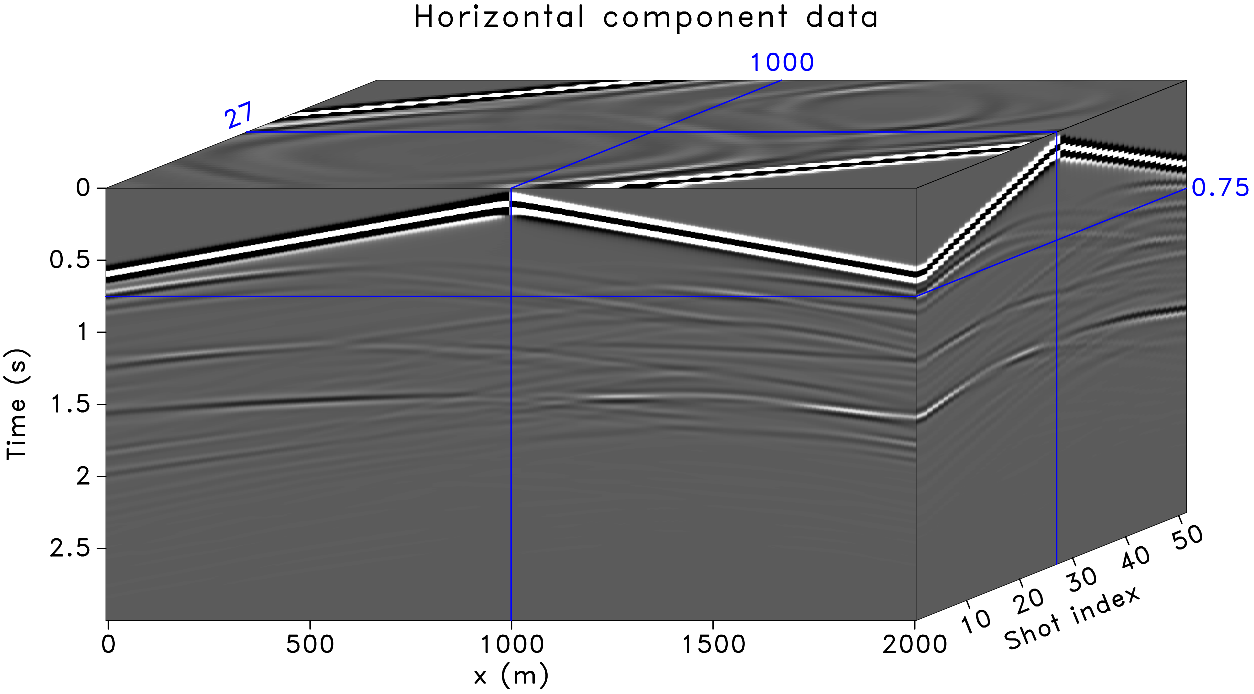
The results of elastic NLCG FWI are shown in Figure 9.4 a and Figure 9.5 a. These results were computed after 20 nonlinear conjugate gradient iterations. The elastic NLCG FWI recovered the



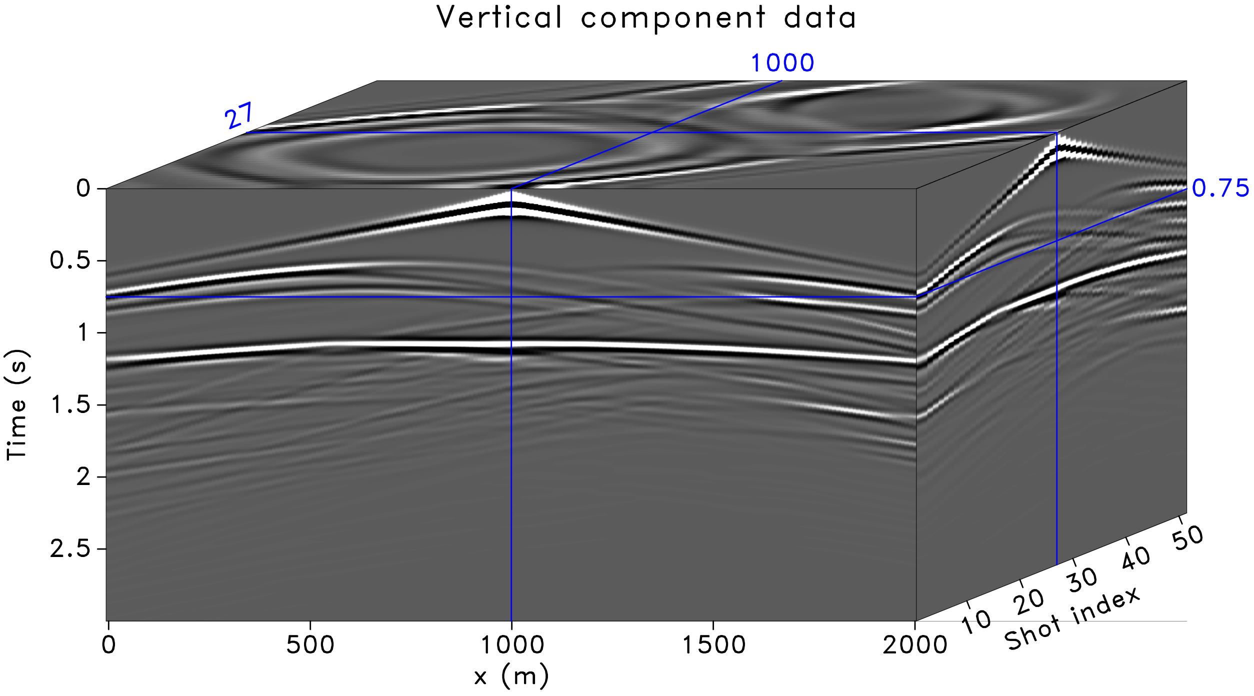
(a)

(b)

Figure 9.1: Elastic inclusion model. (a) True P-wave velocity model. (b) True S-wave velocity model.

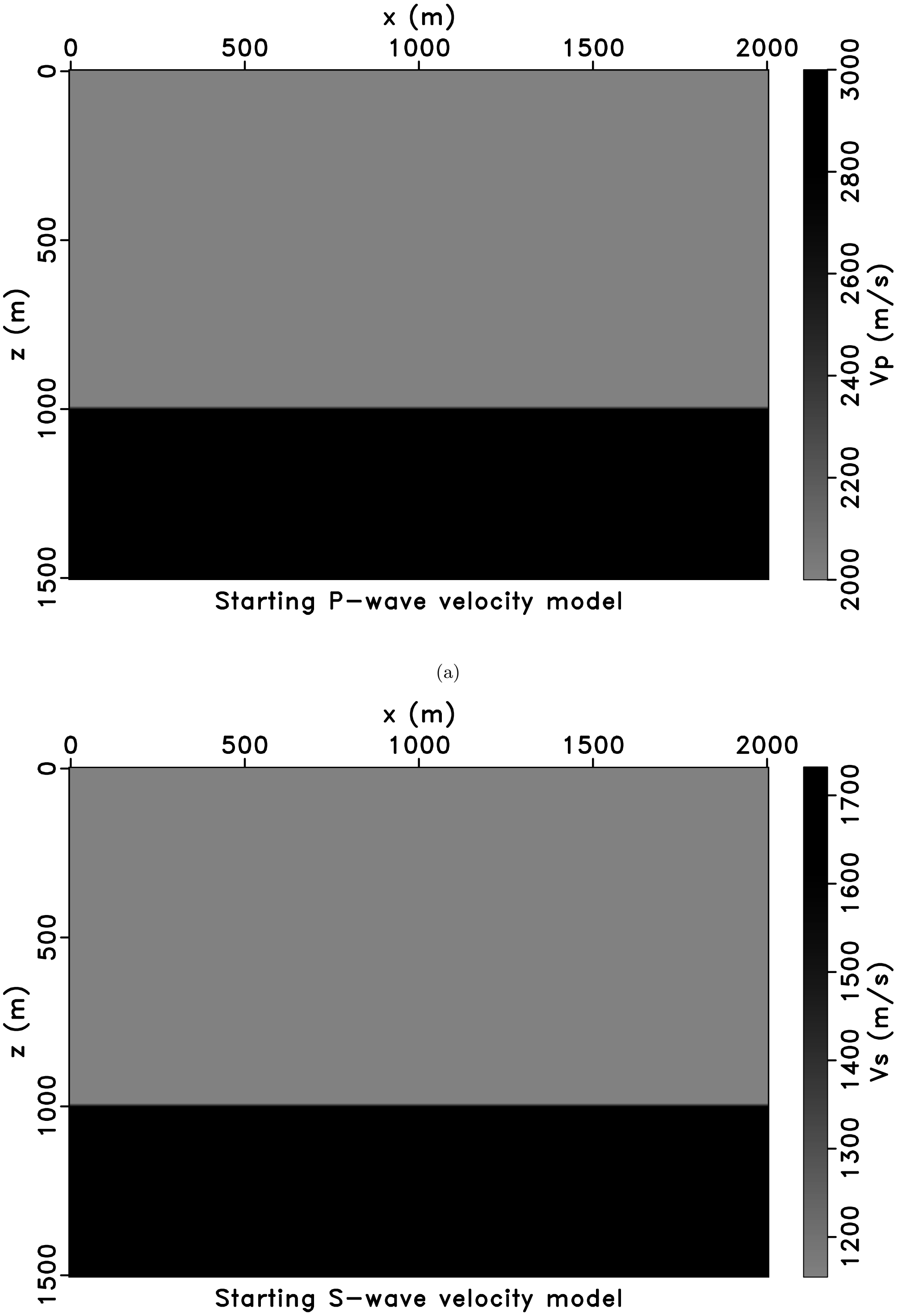


(a)



(b)

Figure 9.2: Prestack multicomponent data. (a) Horizontal particle velocity data. (b) Vertical particle velocity data.



(a)

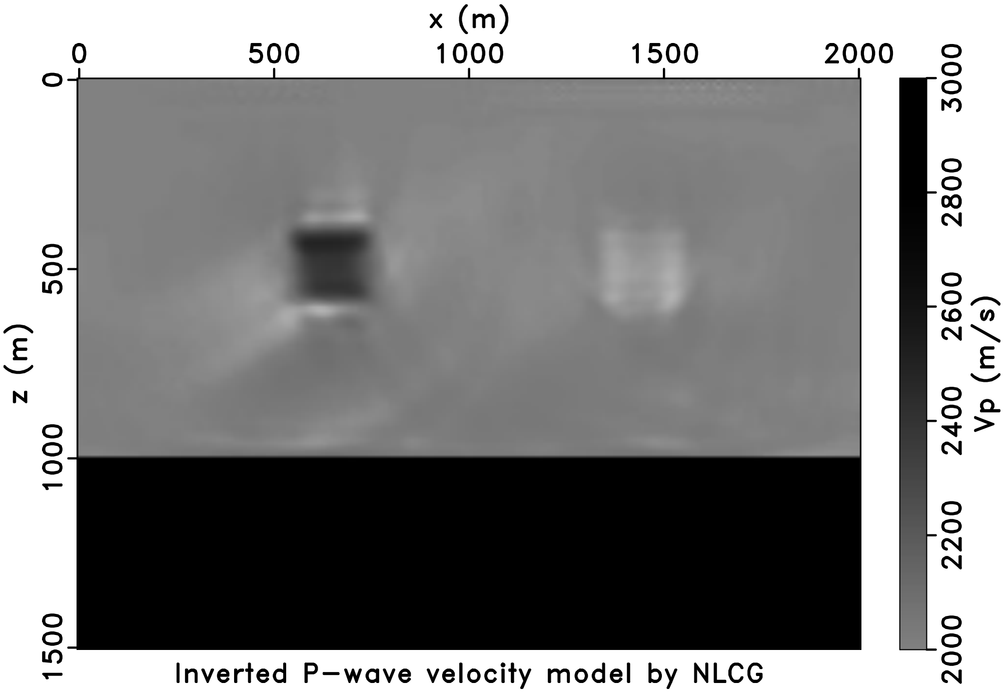
(b)

Figure 9.3: (a) Starting P-wave velocity model. (b) Starting S-wave velocity model.

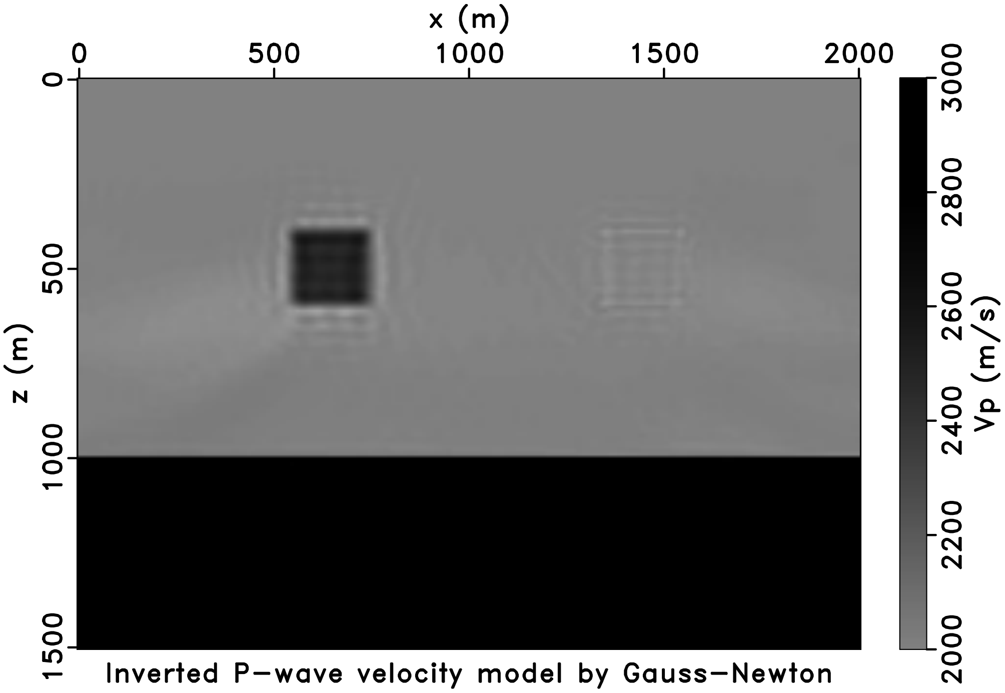
main features of the velocity anomalies. However, the recovered velocities models are not very good. Additionally, there are artifacts in the inverted models. Most important, there is crosstalk between the inverted P-wave and S-wave velocity models. The elastic Gauss-Newton FWI has iterated 10 times for outer FWI loop and 20 times for inner CGLS loop (Figure 9.4 b and Figure 9.5 b). It efficiently suppresses the multiparameter crosstalk. Moreover, the recovered velocity models are much better than the ones obtained by elastic NLCG FWI. The inverted models by elastic Gauss-Newton FWI also contain fewer artifacts. To show the details more clearly, we display the profiles of the inverted P-wave velocity models in Figure 9.6 and the profiles of the inverted S-wave velocity models in Figure 9.7. From the velocity profiles, we can observe that the elastic NLCG FWI does not fully recover the amplitudes of both the P- and S-wave velocity anomalies. The amplitudes are underestimated. Also, the crosstalk from S-wave velocity manifests as negative velocity perturbation in inverted P-wave velocity model. And the crosstalk from P-wave velocity manifests as negative velocity perturbation in inverted S-wave velocity model. However, the proposed elastic Gauss-Newton FWI fully recovers the amplitude of velocity anomalies and also completely removed the crosstalk between P- and S-wave velocity models. We also examine the data fitting of the two elastic FWI algorithms in Figure 9.8 and Figure 9.9. We can see that the data fitting of the elastic Gauss-Newton FWI inverted models (Figure 9.9) is much better than that of the elastic NLCG inverted models (Figure 9.8). We emphasize that image plots in Figure 9.8 and Figure 9.9 have been clipped to the same value. Figure 9.10 compares the converge curves of the elastic NLCG FWI and elastic Gauss-Newton FWI for four different frequency bands. The relative data misfit for frequency band 0-f Hz ($f = 5, 10, 15, 30$) is defined as

$$misfit = \frac{\sum_{i=1}^{N_s} \|\mathbf{d}_i^{0-fHz} - \mathbf{d}_i^{obs\,0-fHz}\|_2^2}{\sum_{i=1}^{N_s} \|\mathbf{d}_i^{obs\,0-fHz}\|_2^2}. \tag{9.24}$$

We can conclude that the elastic Gauss-Newton converges much faster than the elastic NLCG FWI. And elastic Gauss-Newton FWI allows better data fitting than the elastic NLCG FWI.

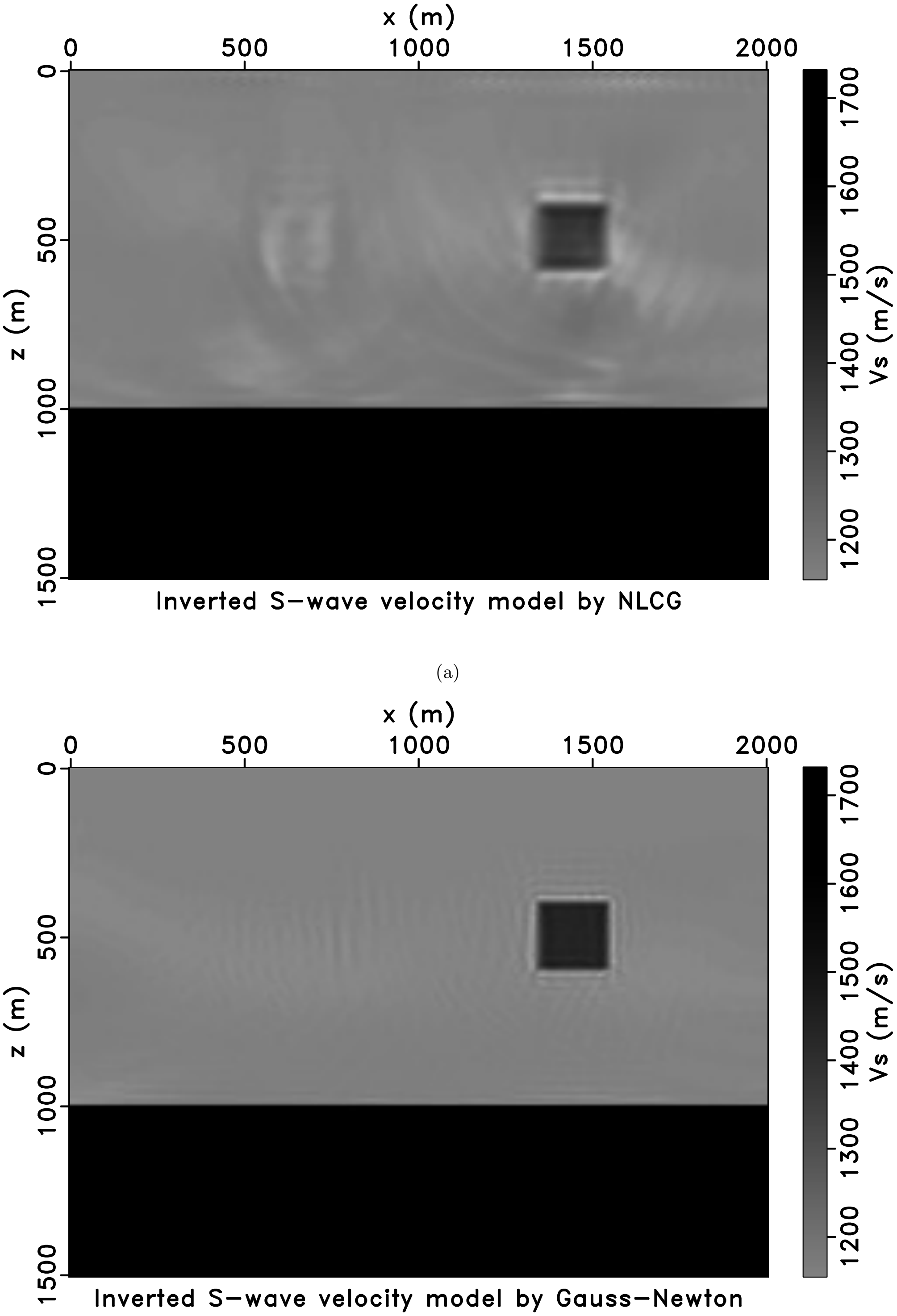


(a)



(b)

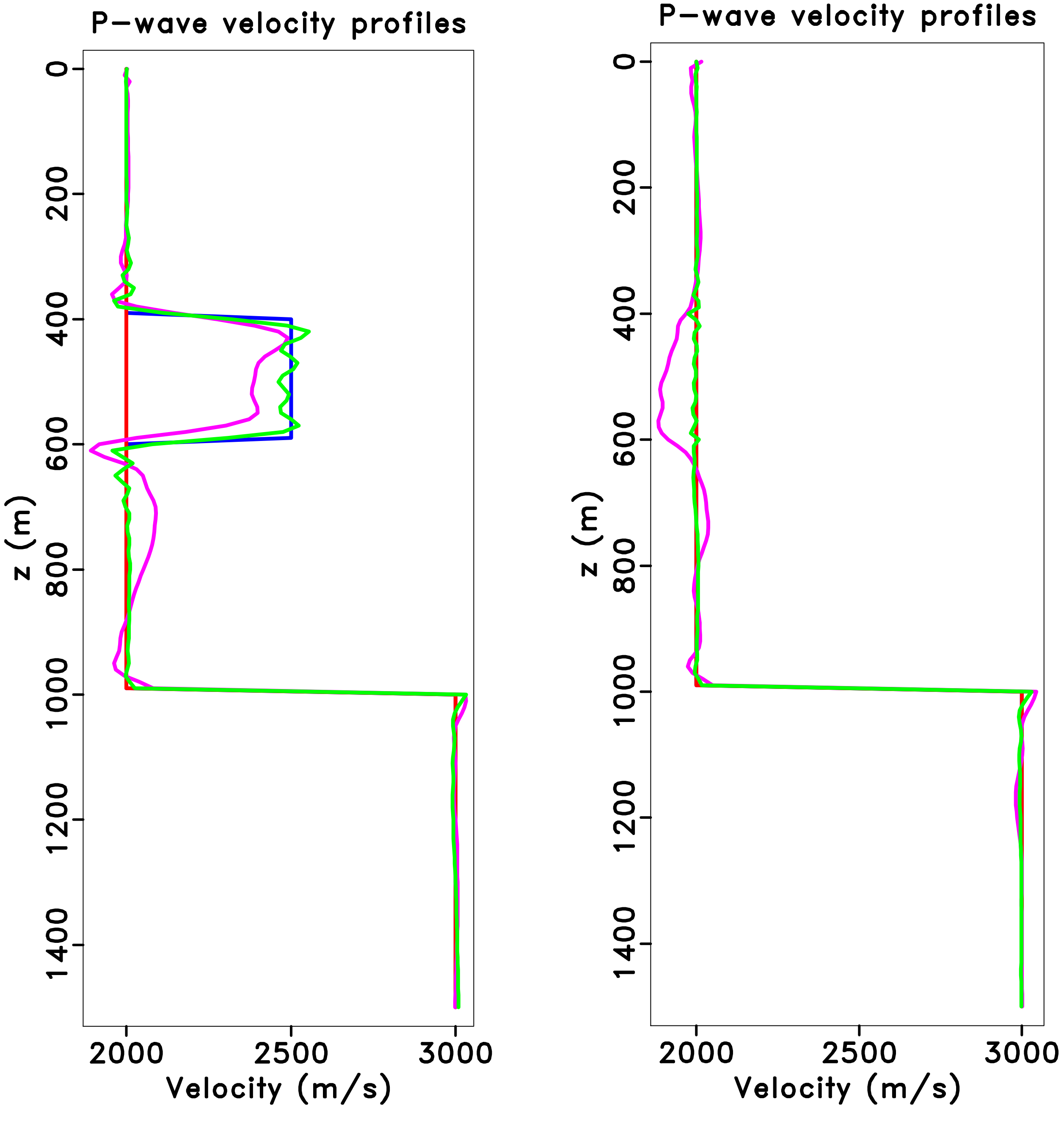
Figure 9.4: (a) Inverted P-wave velocity model by elastic NLCG FWI. (b) Inverted P-wave velocity model by elastic Gauss-Newton FWI.



(a)

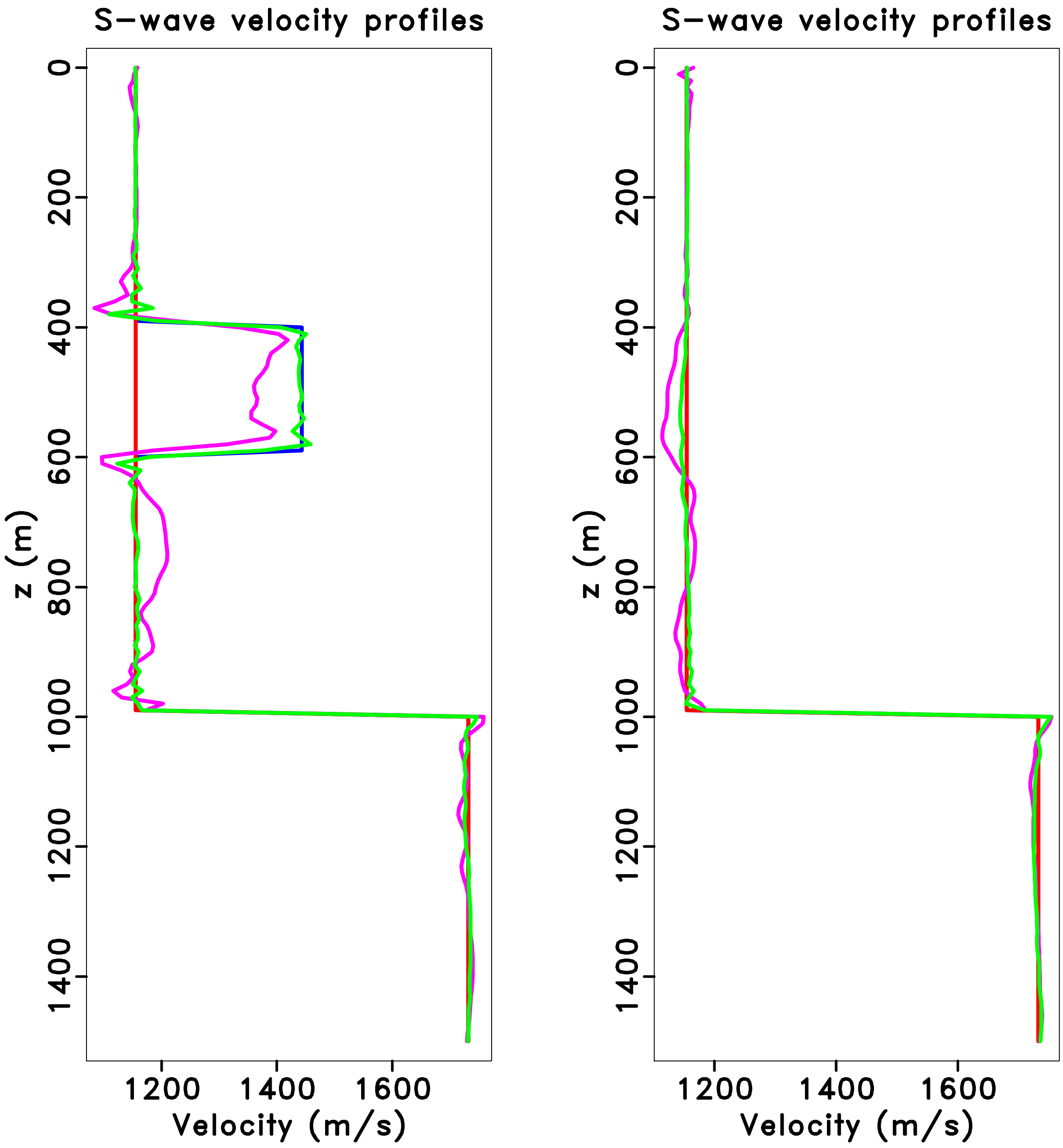
(b)

Figure 9.5: (a) Inverted S-wave velocity model by elastic NLCG FWI. (b) Inverted S-wave velocity model by elastic Gauss-Newton FWI.



(a) (b)

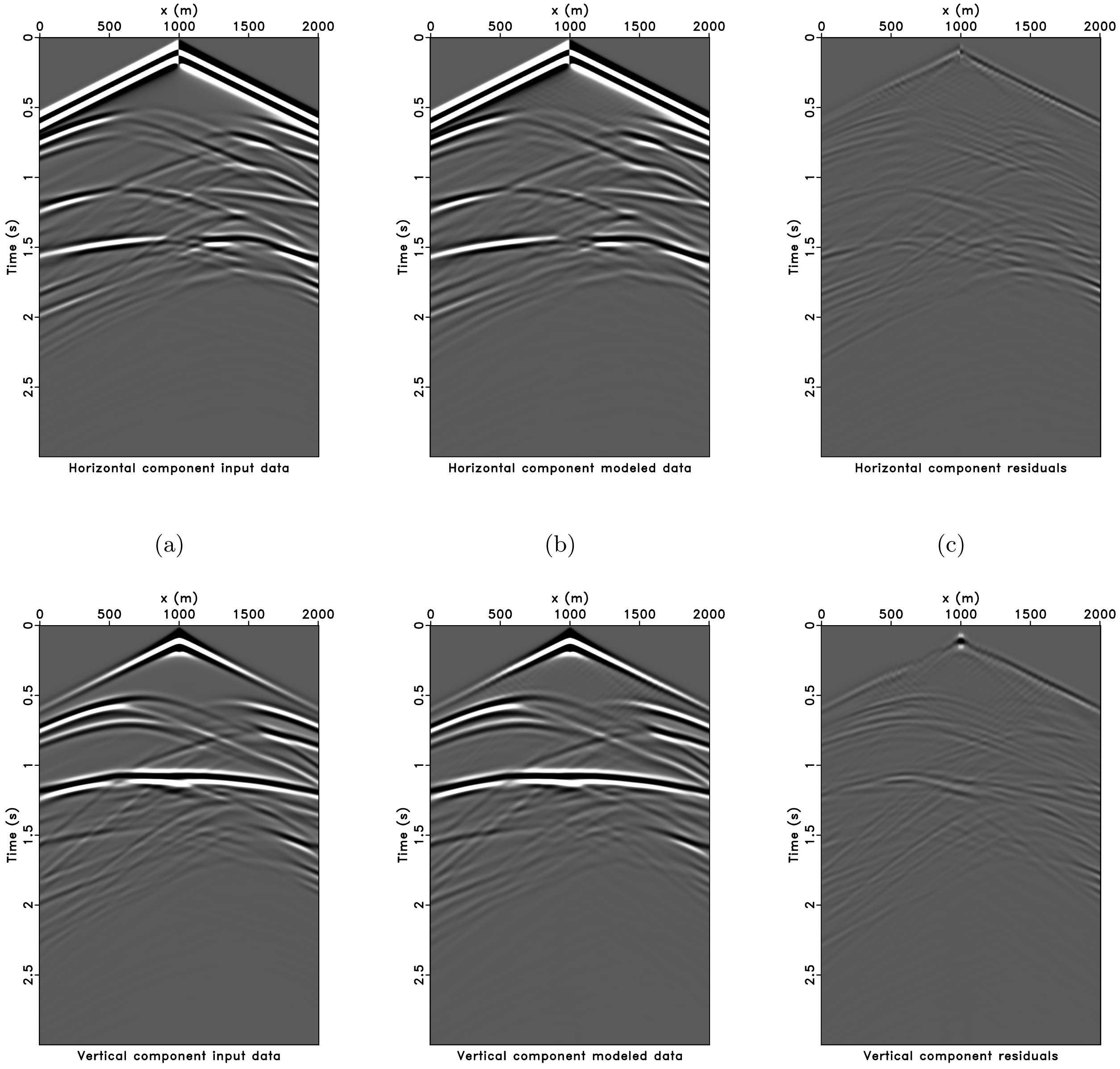
Figure 9.6: Profiles of inverted P-wave velocity models at (a) $x = 875.5$ m and (b) $x = 1785$ m. Blue: true model; Red: starting model; Purple: inverted model by elastic NLCG FWI; Green: inverted model by elastic Gauss-Newton FWI.



(a)

(b)

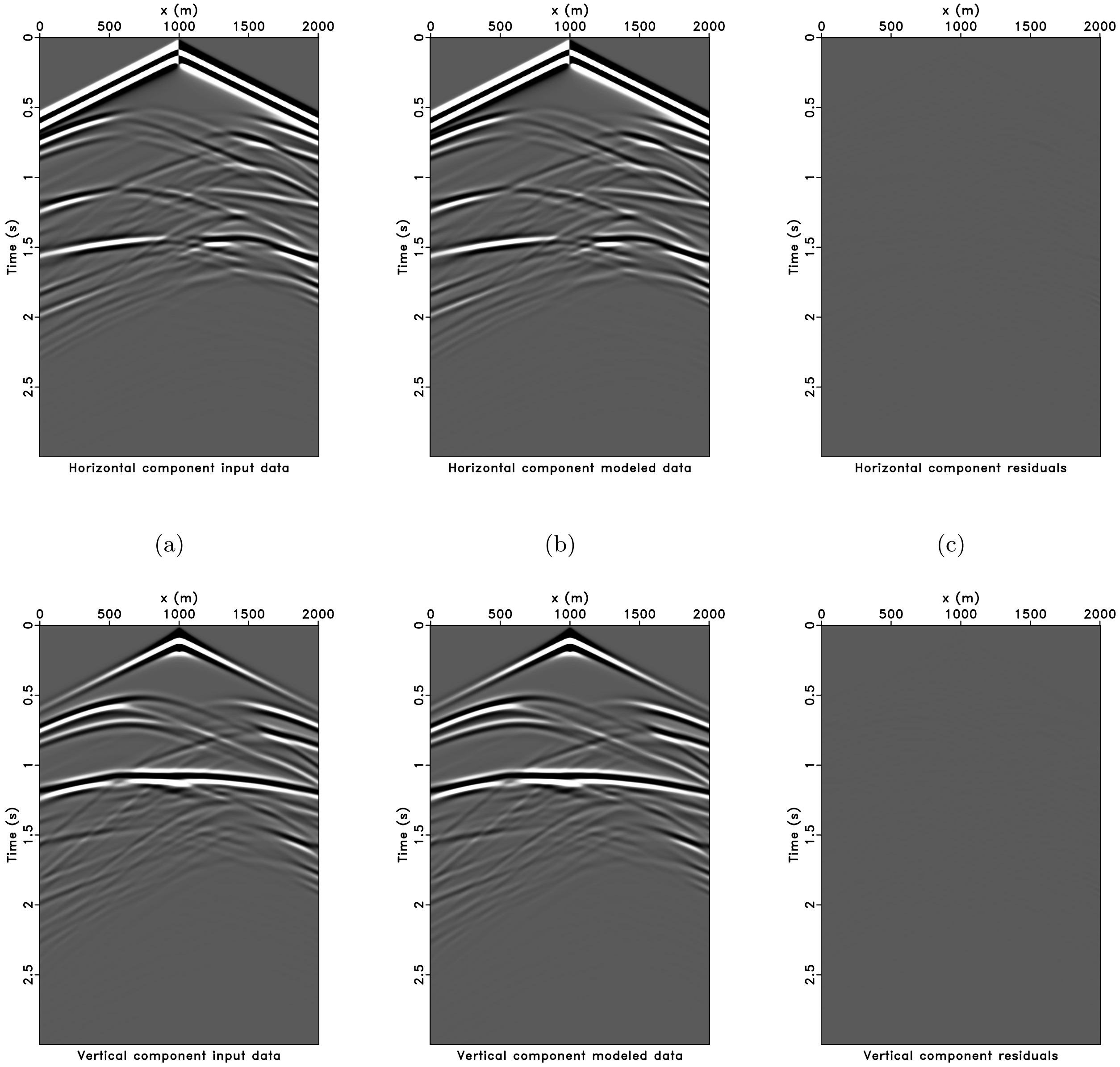
Figure 9.7: Profiles of inverted S-wave velocity models at (a) $x = 1785$ m and (b) $x = 875.5$ m. Blue: true model; Red: starting model; Purple: inverted model by elastic NLCG FWI; Green: inverted model by elastic Gauss-Newton FWI.



(a) (b) (c)

(d) (e) (f)

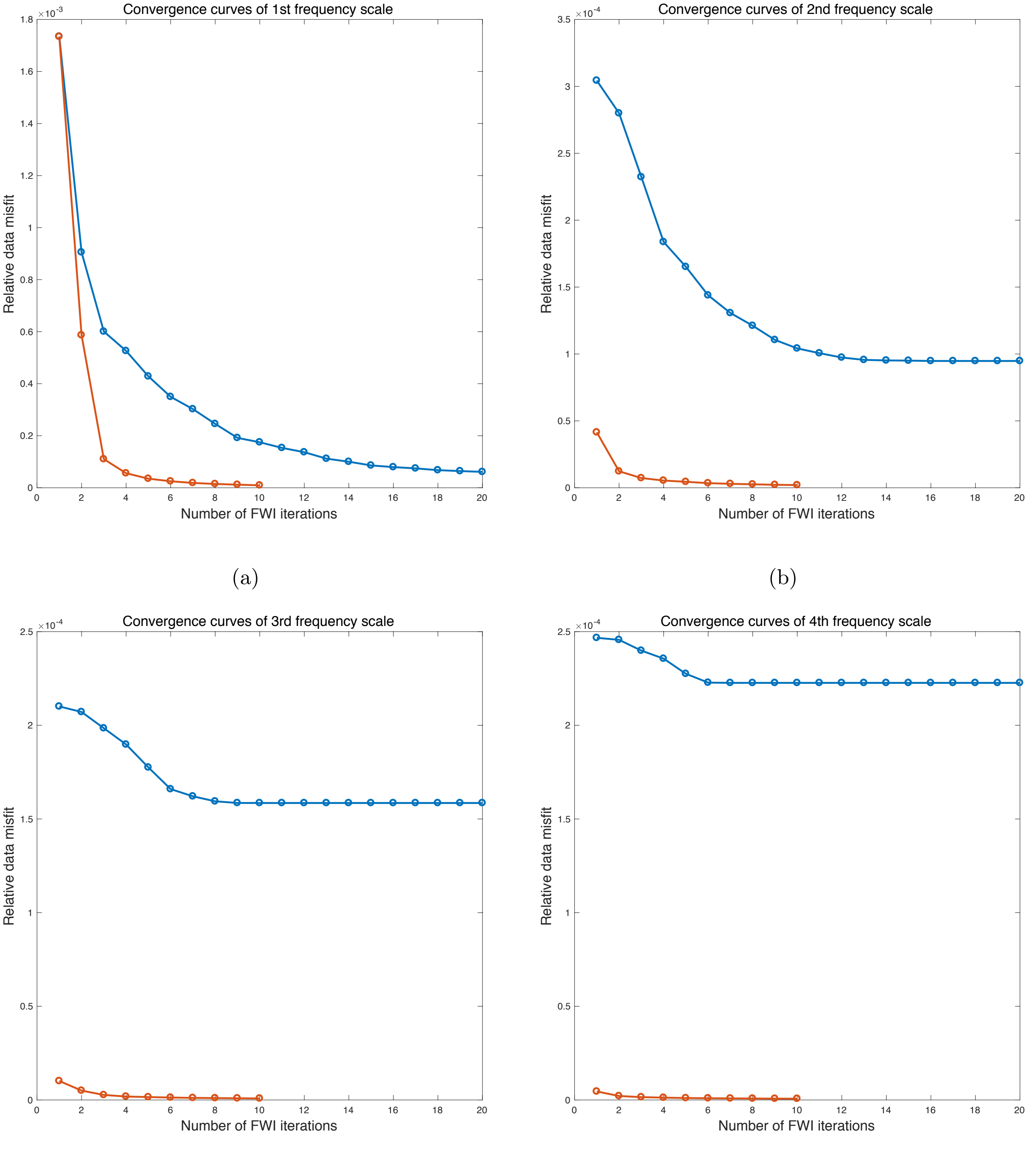
Figure 9.8: Data and residuals for elastic NLCG FWI. (a) Observed horizontal component data of shot at $x = 1000$ m. (b) Horizontal component data modeled by elastic NLCG FWI inverted models. (c) Horizontal component data residual. (d) Observed vertical component data of shot at $x = 1000$ m. (e) Vertical component data modeled by elastic NLCG FWI inverted models. (f) Vertical component data residual.



(a) (b) (c)

(d) (e) (f)

Figure 9.9: Data and residuals for elastic Gauss-Newton FWI. (a) Observed horizontal component data of shot at $x = 1000$ m. (b) Horizontal component data modeled by elastic Gauss-Newton FWI inverted models. (c) Horizontal component data residual. (d) Observed vertical component data of shot at $x = 1000$ m. (e) Vertical component data modeled by elastic Gauss-Newton FWI inverted models. (f) Vertical component data residual.



(a)

(b)

(c)

(d)

Figure 9.10: Relative data misfit curves for elastic NLCG FWI and elastic Gauss-Newton FWI in frequency band (a) 0 - 5 Hz, (b) 0 - 10 Hz, (c) 0 - 15 Hz, (d) 0 - 30 Hz. Blue: data misfit curves for elastic NLCG FWI. Red: data misfit curves for elastic Gauss-Newton FWI.

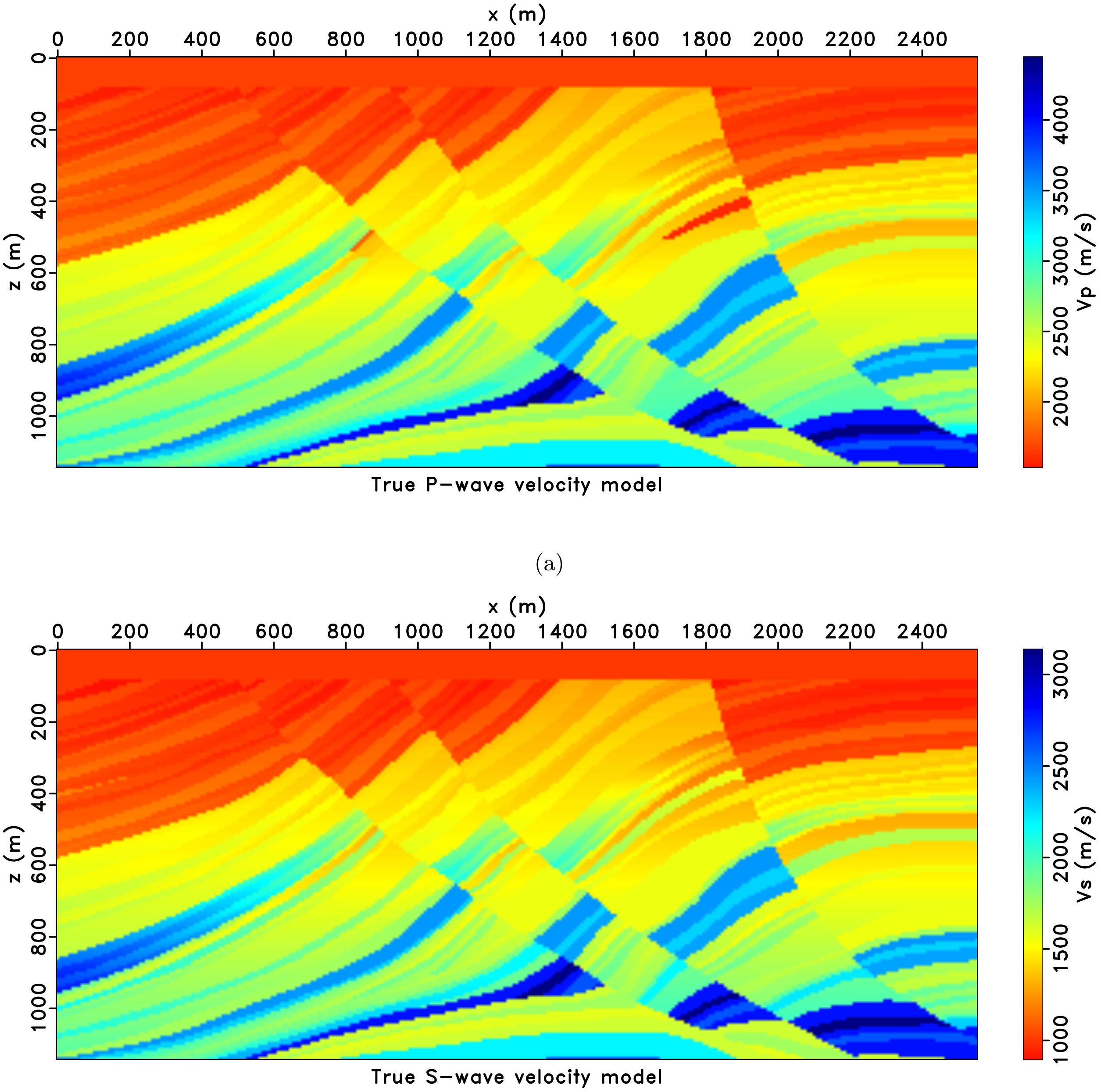
9.3.2 Elastic Marmousi2 model

Figure 9.11 shows the true P- and S-wave velocity models of the elastic Marmousi2 model. The P- and S-wave velocity models are uncorrelated, which pose difficulties for elastic FWI. Density is assumed to be constant (2000 kg/m^3). The model has a dimension of 2550 m in horizontal axis and 1139 m in depth with 301×135 grid points. There are 61 shots and 301 receivers deployed along the surface that simulates a fixed-spread survey geometry. The shot interval is 42.5 m and receiver interval is 8.5 m. The shot depth is 8.5 m and the receiver depth is 17 m. A 10 Hz central frequency Ricker wavelet is used to simulate an explosive source. The multicomponent observed data are simulated using our elastic finite-difference code. The observed data cubes are shown in Figure 9.12. The four frequency bands for multiscale inversion is: 0-2 Hz, 0-5 Hz, 0-10 Hz and 0-30 Hz. Figure 9.13 shows the starting P- and S-wave velocity models for elastic NLCG FWI and elastic Gauss-Newton FWI. The initial models are 1D linearly increasing velocities models.

The results of elastic NLCG FWI are shown in Figure 9.14 a and Figure 9.15 a. These results were computed after 50 nonlinear conjugate gradient iterations. The recovered velocities models are not very good. The elastic Gauss-Newton FWI has iterated 10 times for outer FWI loop and 40 times for inner CGLS loop (Figure 9.14 b and Figure 9.15 b). It efficiently suppresses the multiparameter crosstalk. Moreover, the recovered velocity models are much better than the ones obtained by elastic NLCG FWI. To show the details more clearly, we display the profiles of the inverted P-wave velocity models in Figure 9.16 and the profiles of the inverted S-wave velocity models in Figure 9.17. From the velocity profiles, we can observe that the elastic NLCG FWI does not correctly recover the amplitudes of both the P- and S-wave velocity anomalies. However, the proposed elastic Gauss-Newton FWI recovers the amplitude of velocity models very well. We also examine the data fitting of the two elastic FWI algorithms in Figure 9.18 and Figure 9.19. We can see that the data fitting of the elastic Gauss-Newton FWI inverted models (Figure 9.19) is much better than that of the elastic NLCG inverted models (Figure 9.18). We emphasize that image plots in Figure 9.18 and Figure 9.19 have been clipped to the same value. Figure 9.20 compares the converge curves of the elastic NLCG FWI and elastic Gauss-Newton FWI for four different frequency bands. We can conclude that the elastic Gauss-Newton converges much faster than the elastic NLCG FWI. And elastic Gauss-Newton FWI allows better data fitting than the elastic NLCG FWI.

9.4 Conclusions

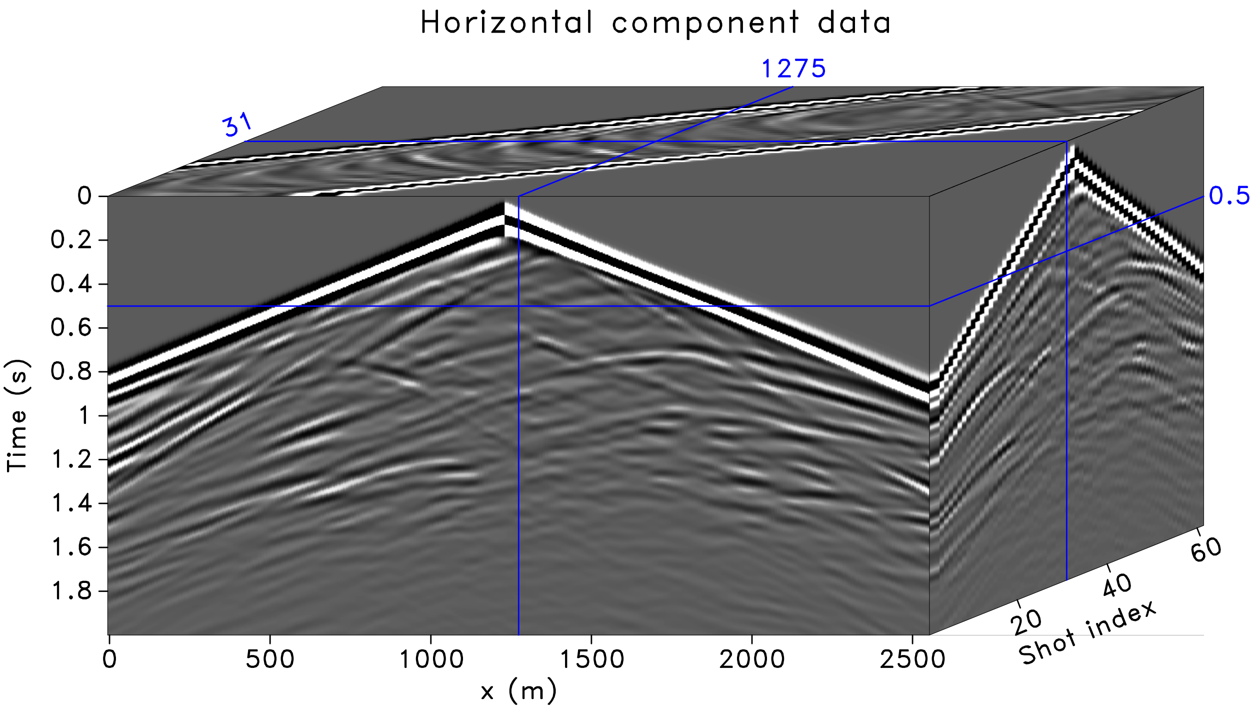
We propose a time-domain matrix-free elastic Gauss-Newton FWI algorithm based on elastic LSRTM algorithm. We formulate the elastic Gauss-Newton FWI as an iterative elastic LSRTM problem.



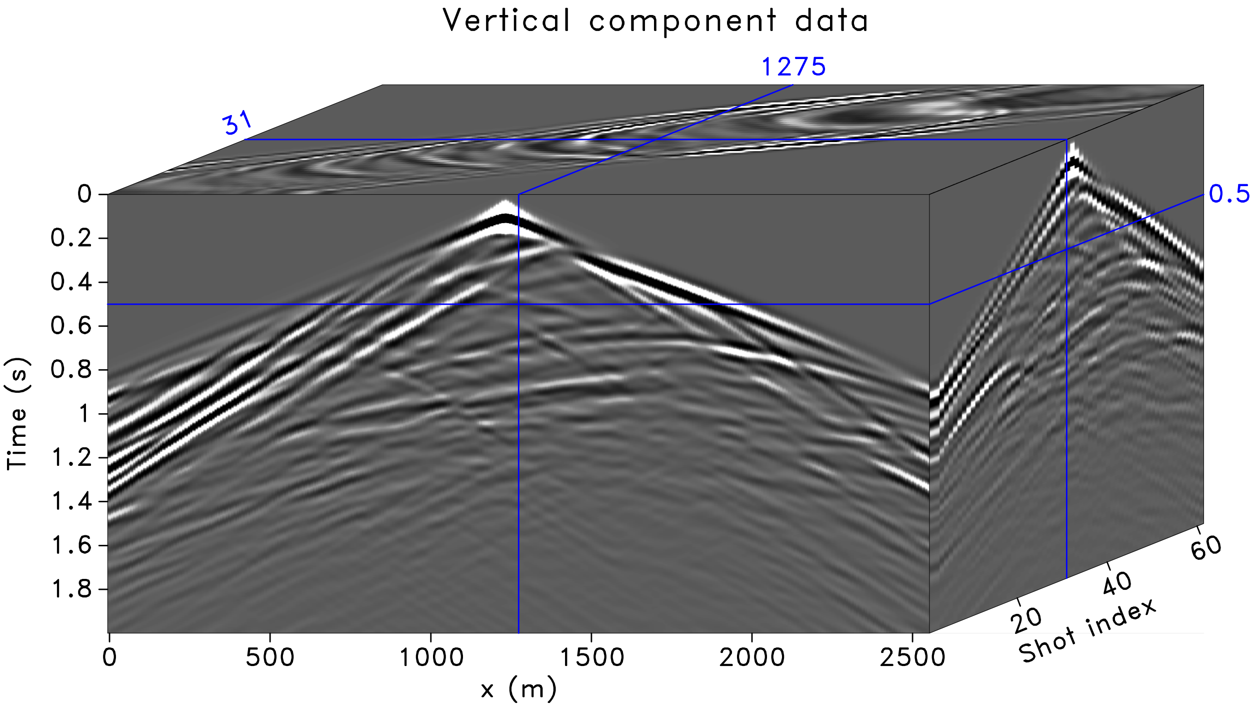
(a)

(b)

Figure 9.11: Elastic Marmousi2 model. (a) True P-wave velocity model. (b) True S-wave velocity model.

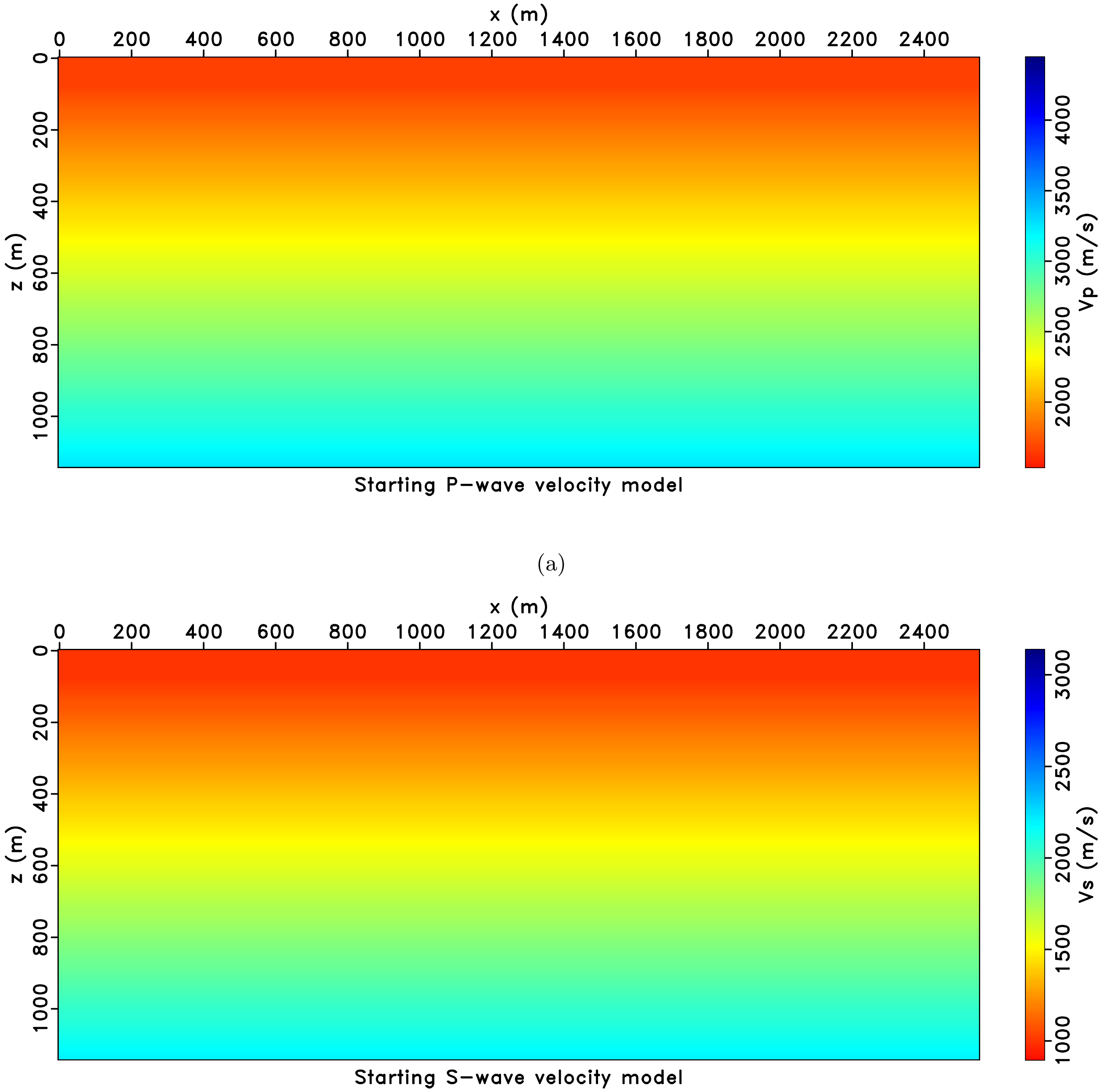


(a)



(b)

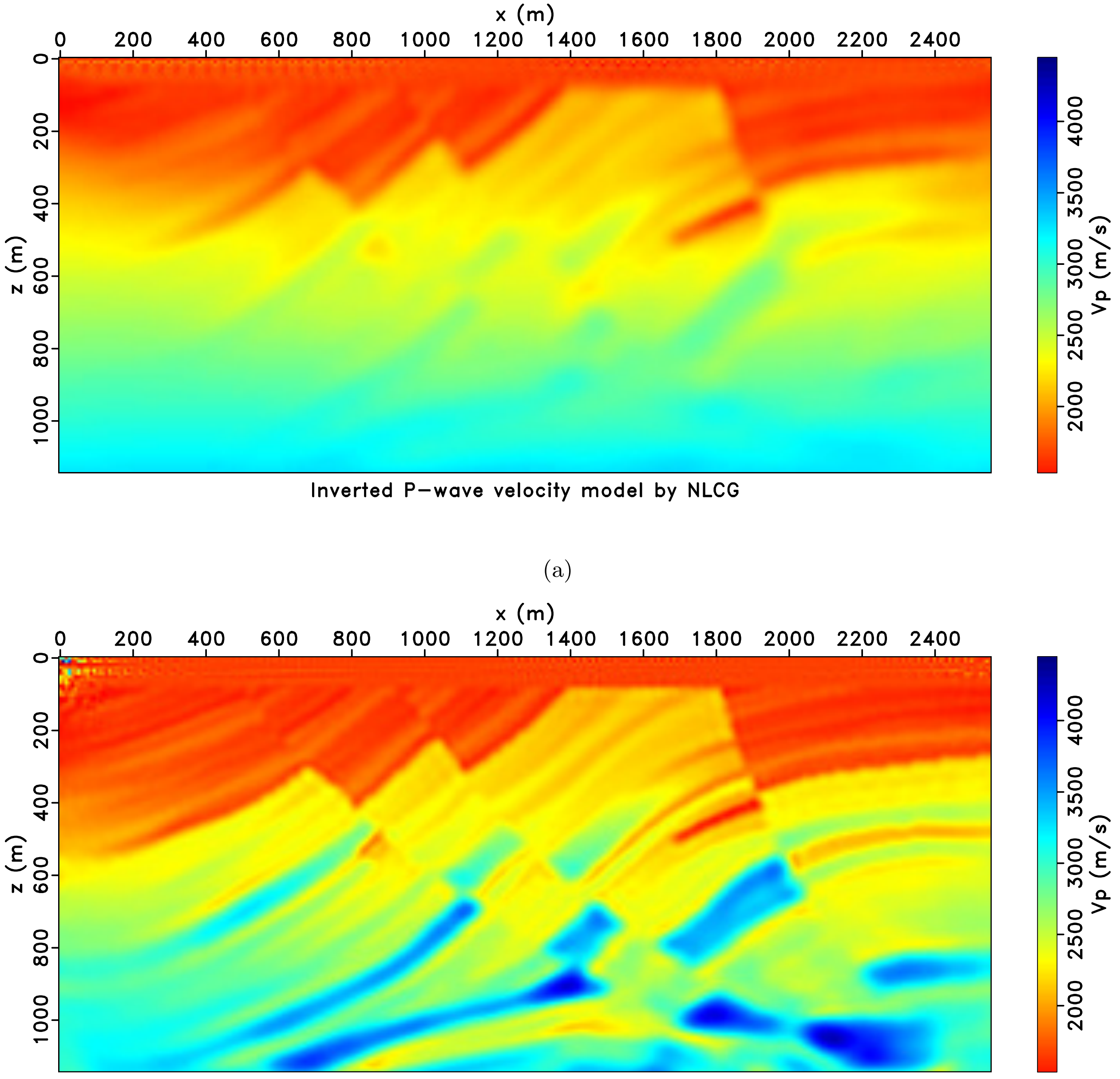
Figure 9.12: Prestack multicomponent data. (a) Horizontal particle velocity data. (b) Vertical particle velocity data.



(a)

(b)

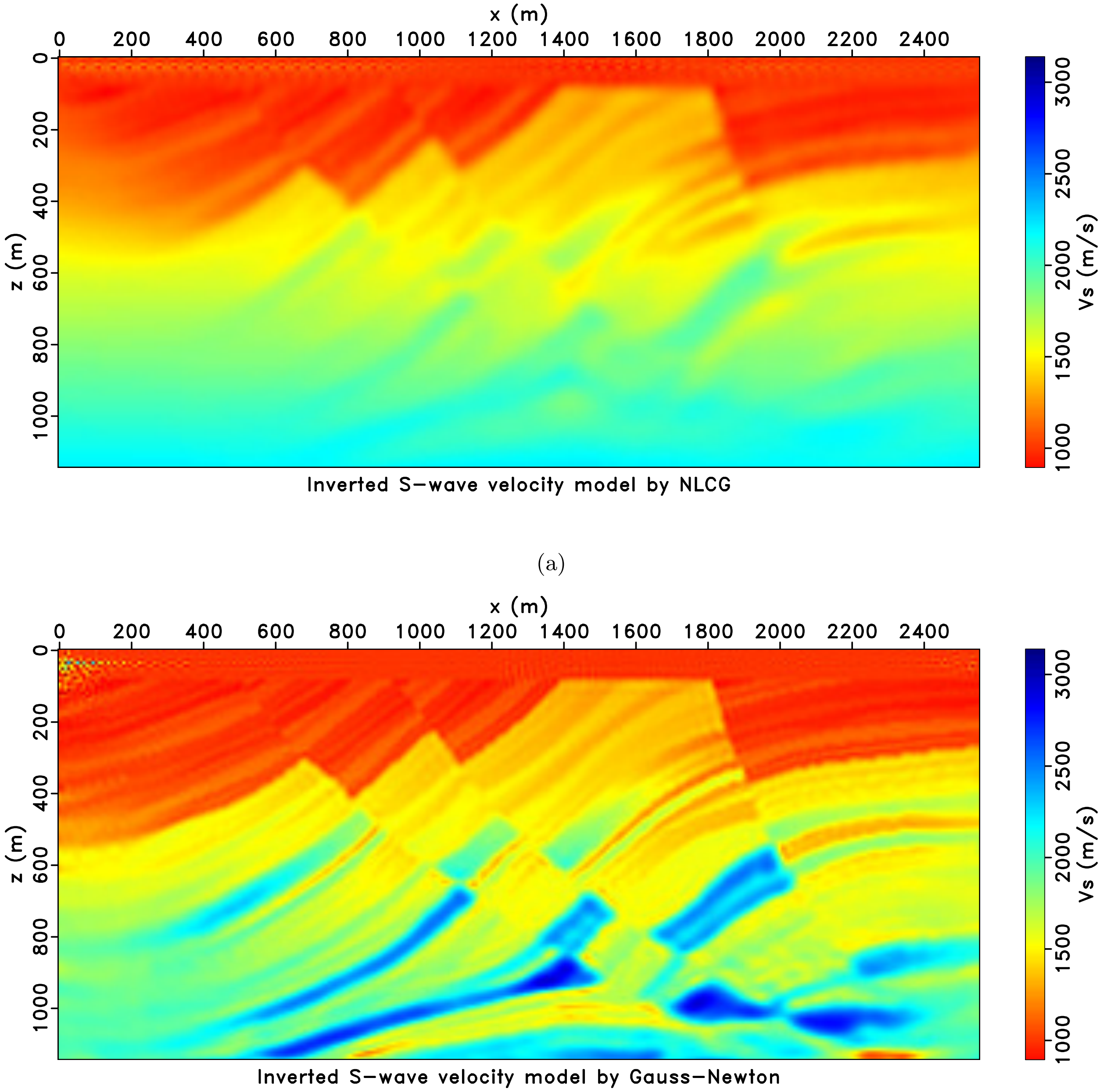
Figure 9.13: (a) Starting P-wave velocity model. (b) Starting S-wave velocity model.



(a)

(b)

Figure 9.14: (a) Inverted P-wave velocity model by elastic NLCG FWI. (b) Inverted P-wave velocity model by elastic Gauss-Newton FWI.



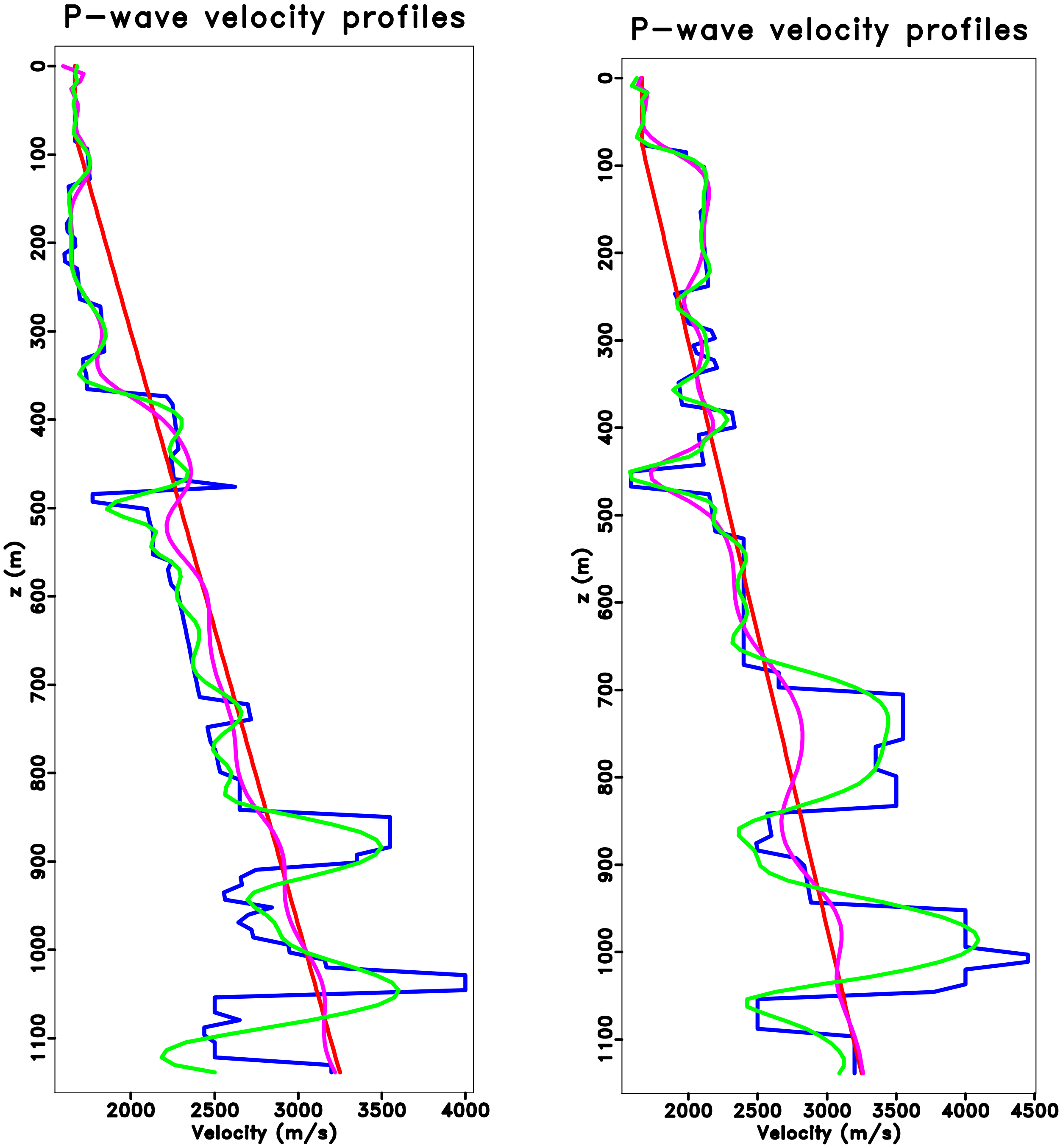
Inverted S−wave velocity model by NLCG

(a)

Inverted S−wave velocity model by Gauss−Newton

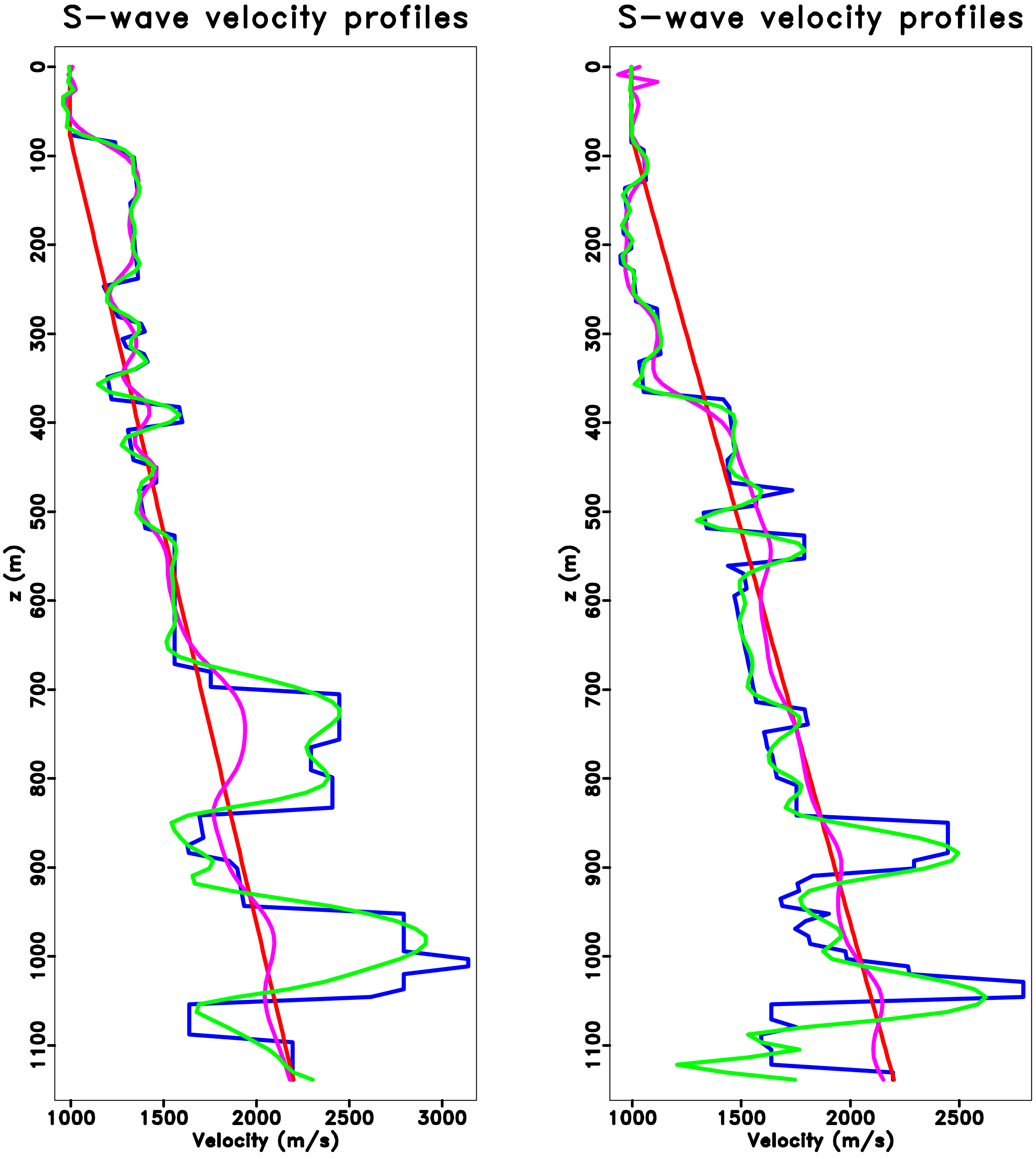
(b)

Figure 9.15: (a) Inverted S-wave velocity model by elastic NLCG FWI. (b) Inverted S-wave velocity model by elastic Gauss-Newton FWI.



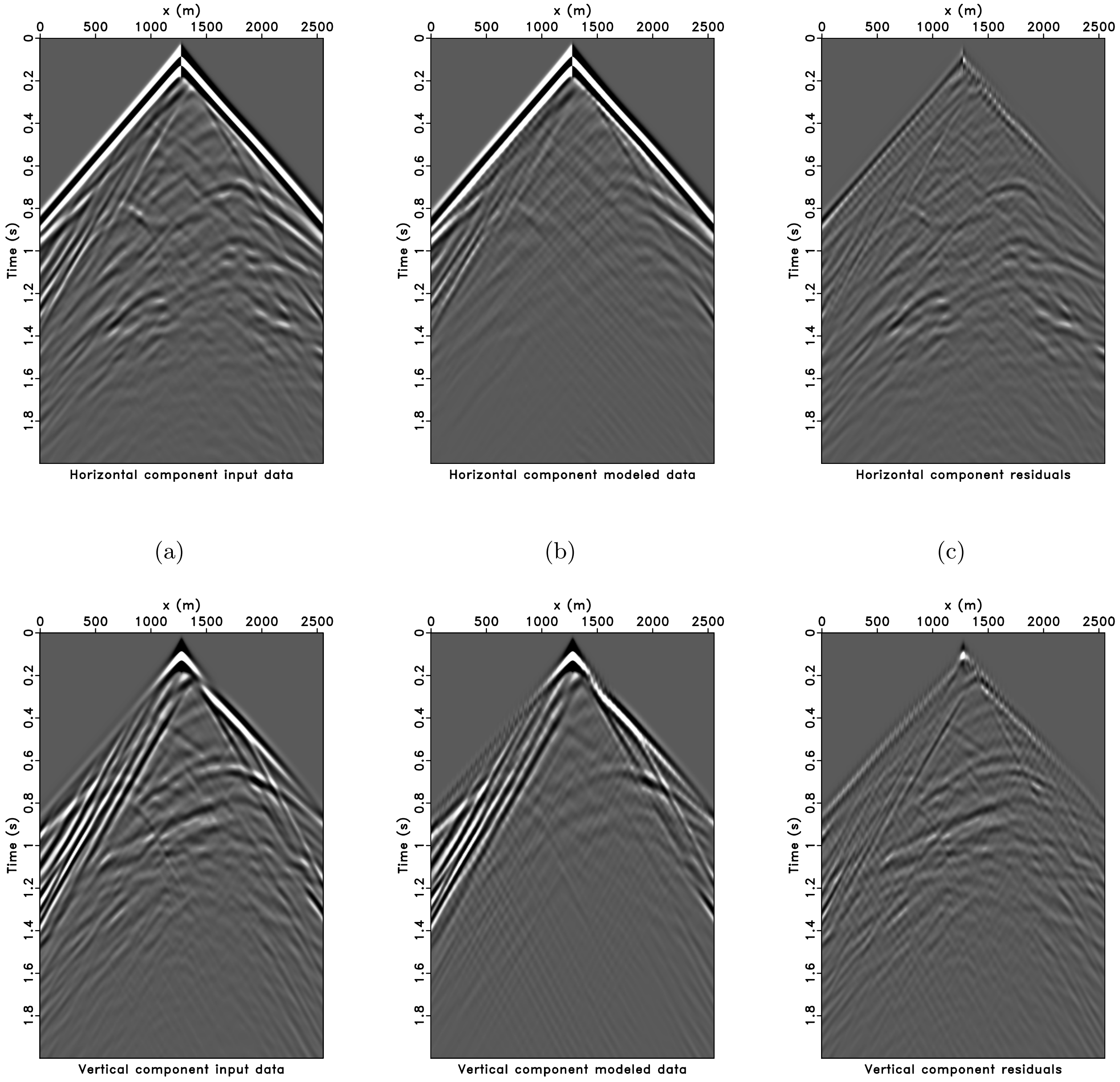
(a) (b)

Figure 9.16: Profiles of inverted P-wave velocity models at (a) $x = 650$ m and (b) $x = 1520$ m. Blue: true model; Red: starting model; Purple: inverted model by elastic NLCG FWI; Green: inverted model by elastic Gauss-Newton FWI.



(a) (b)

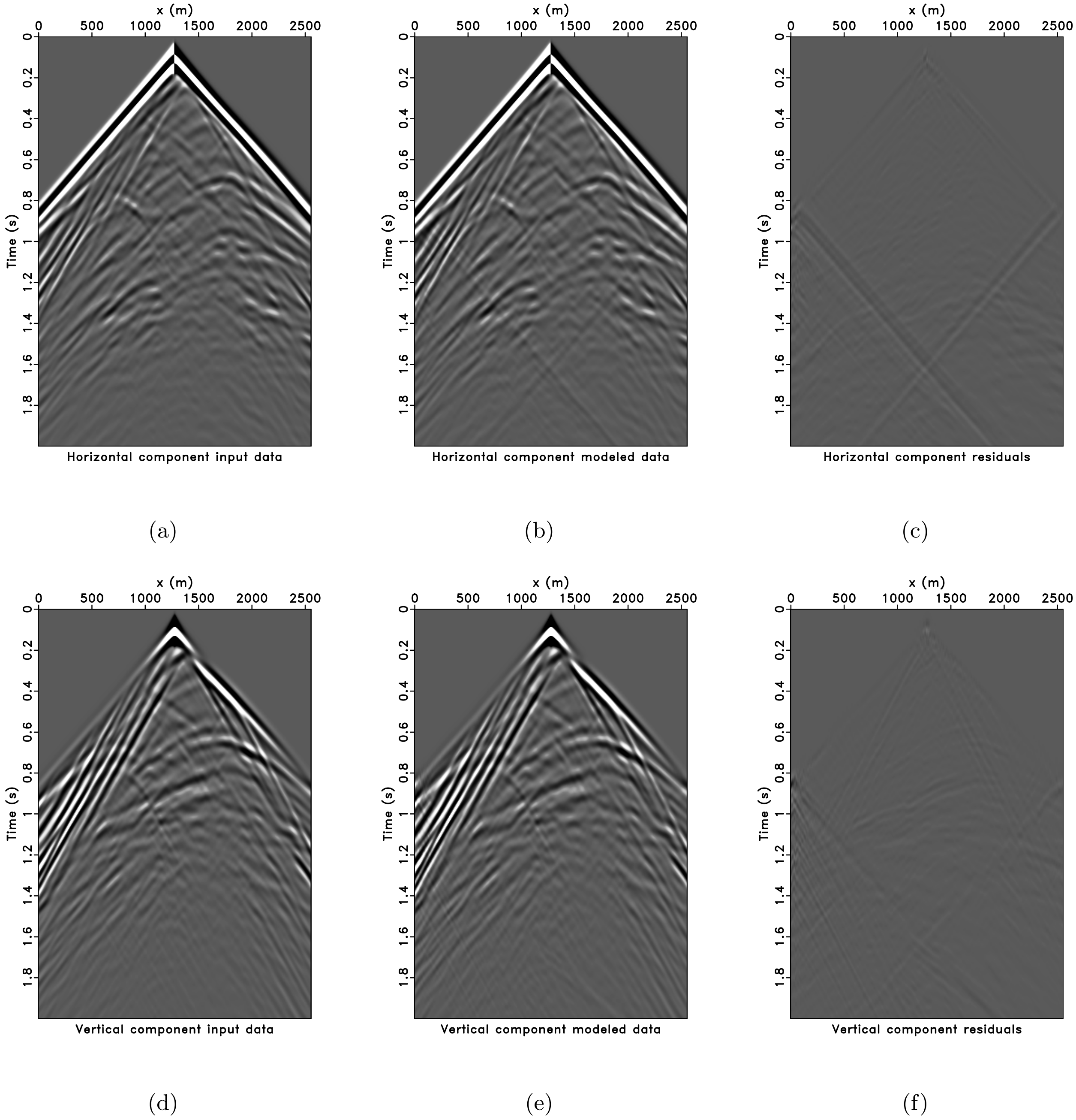
Figure 9.17: Profiles of inverted S-wave velocity models at (a) $x = 1450$ m and (b) $x = 590$ m. Blue: true model; Red: starting model; Purple: inverted model by elastic NLCG FWI; Green: inverted model by elastic Gauss-Newton FWI.



(a) (b) (c)

(d) (e) (f)

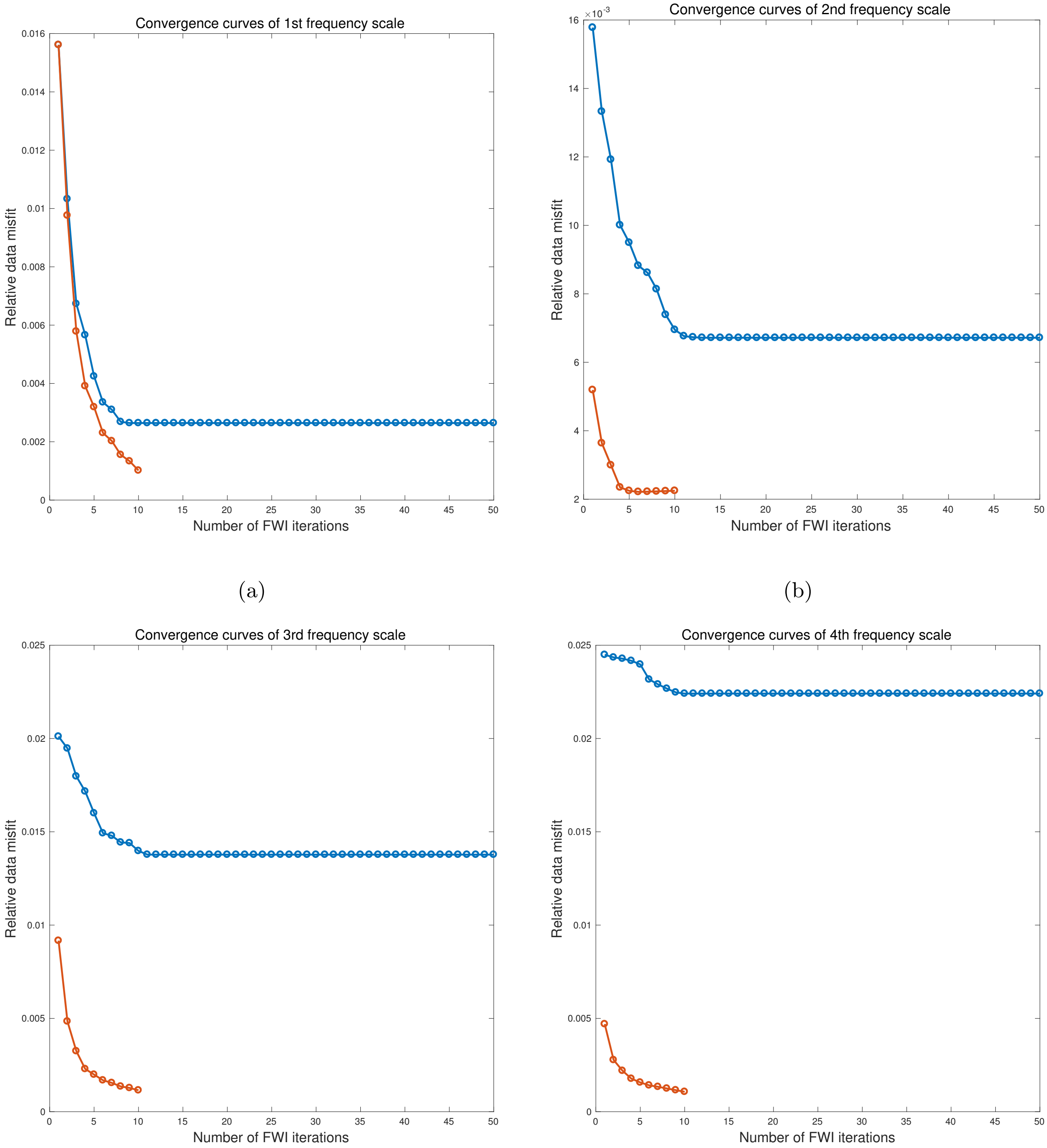
Figure 9.18: Data and residuals for elastic NLCG FWI. (a) Observed horizontal component data of shot at $x = 1000$ m. (b) Horizontal component data modeled by elastic NLCG FWI inverted models. (c) Horizontal component data residual. (d) Observed vertical component data of shot at $x = 1000$ m. (e) Vertical component data modeled by elastic NLCG FWI inverted models. (f) Vertical component data residual.



(a) (b) (c)

(d) (e) (f)

Figure 9.19: Data and residuals for elastic Gauss-Newton FWI. (a) Observed horizontal component data of shot at $x = 1000$ m. (b) Horizontal component data modeled by elastic Gauss-Newton FWI inverted models. (c) Horizontal component data residual. (d) Observed vertical component data of shot at $x = 1000$ m. (e) Vertical component data modeled by elastic Gauss-Newton FWI inverted models. (f) Vertical component data residual.



(a)

(b)

(c)

(d)

Figure 9.20: Relative data misfit curves for elastic NLCG FWI and elastic Gauss-Newton FWI in frequency band (a) 0 - 2 Hz, (b) 0 - 5 Hz, (c) 0 - 10 Hz, (d) 0 - 30 Hz. Blue: data misfit curves for elastic NLCG FWI. Red: data misfit curves for elastic Gauss-Newton FWI.

The proposed algorithm consists of two loops of iterations: the outer Gauss-Newton nonlinear iterations and the inner conjugate gradient least-squares (CGLS) linear iterations. The outer nonlinear iteration uses parabola fitting line search to estimate the step size. The Gauss-Newton search direction in each outer FWI iteration is computed using the matrix-free CGLS algorithm. We point out that this step is actually equivalent to apply an elastic LSRTM on data residual, with the Jacobian operator as elastic Born modeling operator and the adjoint of Jacobian operator as elastic RTM operator. The CGLS algorithm can be safely used for solving the Gauss-Newton search direction because our discretized numerical versions of elastic Born and RTM operators passed the dot-product test. In the inner CGLS linear iterations, the step size is analytically calculated without of the need of line search. The inner CGLS linear iterations are preconditioned using the elastic pseudo-Hessian operator. Our algorithm is matrix-free that only requires the forward Jacobian and adjoint Jacobian operator applied "on the fly" to vectors. The operators are applied on vectors efficiently via the adjoint-state method. We use the proposed algorithm to simultaneously invert for P- and S-wave velocities.The proposed elastic Gauss-Newton FWI generates better inverted models than the nonlinear conjugate gradient method based elastic FWI. More importantly, the elastic Gauss-Newton FWI can decouple the crosstalk between P- and S-wave velocity models.

9.5 Acknowledgments

We thank the sponsors of the Signal Analysis and Imaging Group (SAIG) at the University of Alberta for financial support. High-performance clustered computing resources were provided by Compute Canada (Cedar and Graham). Figures in this paper were plotted using the Madagascar open-source software package.

References

Albertin, U., P. Shen, A. Sekar, T. Johnsen, C. Wu, K. Nihei, and K. Bube, 2016, 3D orthorhombic elastic full-waveform inversion in the reflection domain from hydrophone data: 86th Annual International Meeting, SEG, Expanded Abstracts, 1094–1098.

Bamberger, A., G. Chavent, C. Hemon, and P. Lailly, 1982, Inversion of normal incidence seismograms: Geophysics, **47**, 757–770.

Bjorck, A., 1996, Numerical methods for least squares problems: Society for Industrial and Applied Mathematics.

Borisov, D., and S. C. Singh, 2015, Three-dimensional elastic full waveform inversion in a marine environment using multicomponent ocean-bottom cables: a synthetic study: Geophysical Journal International, **201**, 1215–1234.

Bunks, C., F. M. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: Geophysics, **60**, 1457–1473.

Chen, K., and M. D. Sacchi, 2017, Elastic least-squares reverse time migration via linearized elastic full-waveform inversion with pseudo-Hessian preconditioning: Geophysics, **82**, S341–S358.

Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications Cambridge, Massachusetts, USA.

Crase, E., A. Pica, M. Noble, J. McDonald, and A. Tarantola, 1990, Robust elastic nonlinear waveform inversion: Application to real data: Geophysics, **55**, 527–538.

Epanomeritakis, I., V. Akcelik, O. Ghattas, and J. Bielak, 2008, A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion: Inverse Problems, **24**, 034015.

Fichtner, A., and J. Trampert, 2011, Hessian kernels of seismic data functionals based upon adjoint techniques: Geophysical Journal International, **185**, 775–798.

Gauthier, O., J. Virieux, and A. Tarantola, 1986, Two-dimensional nonlinear inversion of seismic waveforms: Numerical results: Geophysics, **51**, 1387–1403.

Guasch, L., 2011, 3D elastic full-waveform inversion: Ph.D. thesis, Imperial College London.

Hestenes, M. R., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, 409–436.

Jin, S., R. Madariaga, J. Virieux, and G. Lambar, 1992, Two-dimensional asymptotic iterative elastic inversion: Geophysical Journal International, **108**, 575–588.

Kohn, D., D. De Nil, A. Kurzmann, A. Przebindowska, and T. Bohlen, 2012, On the influence of model parametrization in elastic full waveform tomography: Geophysical Journal International, **191**, 325–345.

Komatitsch, D., and R. Martin, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: Geophysics, **72**, SM155–SM167.

Lailly, P., 1983, The seismic inverse problem as a sequence of before-stack migrations: Conference on Inverse Scattering: Theory and Applications. SIAM, Philadelphia, 206–220.

Levander, A. R., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Lions, J. L., 1971, Optimal control of systems governed by partial differential equations: Springer Berlin Heidelberg.

Martin, G. S., R. Wiley, and K. J. Marfurt, 2006, Marmousi2: An elastic upgrade for marmousi: The Leading Edge, **25**, 156–166.

Matharu, G., and M. Sacchi, 2017, Feasibility testing of simultaneous source elastic full-waveform inversion: 87th Annual International Meeting, SEG, Expanded Abstracts, 1573–1577.

Metivier, L., R. Brossier, J. Virieux, and S. Operto, 2013, Full waveform inversion and the truncated Newton method: SIAM Journal on Scientific Computing, **35**, B401–B437.

Mora, P., 1987a, Elastic wavefield inversion: Ph.D. thesis, Stanford University.

———, 1987b, Nonlinear two-dimensional elastic inversion of multioffset seismic data: Geophysics, **52**, 1211–1228.

———, 1988, Elastic wave-field inversion of reflection and transmission data: Geophysics, **53**, 750–759.

Nocedal, J., and S. J. Wright, 2006, Numerical Optimization: Springer.

Paige, C. C., and M. A. Saunders, 1982, LSQR: An algorithm for sparse linear equations and sparse least squares: ACM Trans. Math. Softw., **8**, 43–71.

Pan, W., K. A. Innanen, G. F. Margrave, M. C. Fehler, X. Fang, and J. Li, 2016, Estimation of elastic constants for HTI media using Gauss-Newton and full-Newton multiparameter full-waveform inversion: Geophysics, **81**, R275–R291.

Pan, W., K. A. Innanen, and Y. Yuan, 2017, Interparameter tradeoff quantification and reduction in isotropic-elastic full-waveform inversion: 87th Annual International Meeting, SEG, Expanded Abstracts, 1539–1544.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Pratt, R. G., C. Shin, and G. J. Hick, 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Raknes, E. B., B. Arntsen, and W. Weibull, 2015, Three-dimensional elastic full waveform inversion using seismic data from the Sleipner area: Geophysical Journal International, **202**, 1877–1894.

Ren, Z., and Y. Liu, 2016, A hierarchical elastic full-waveform inversion scheme based on wavefield separation and the multistep-length approach: Geophysics, **81**, R99–R123.

Sears, T., S. Singh, and P. Barton, 2008, Elastic full waveform inversion of multi-component OBC seismic data: Geophysical Prospecting, **56**, 843–862.

Sears, T. J., P. J. Barton, and S. C. Singh, 2010, Elastic full waveform inversion of multicomponent ocean-bottom cable seismic data: Application to Alba field, U. K. North sea: Geophysics, **75**, R109–R119.

Sheen, D.-H., K. Tuncay, C.-E. Baag, and P. J. Ortoleva, 2006, Time domain Gauss-Newton seismic waveform inversion in elastic media: Geophysical Journal International, **167**, 1373–1384.

Shin, C., S. Jang, and D.-J. Min, 2001a, Improved amplitude preservation for prestack depth migration by inverse scattering theory: Geophysical Prospecting, **49**, 592–606.

Shin, C., K. Yoon, K. J. Marfurt, K. Park, D. Yang, H. Y. Lim, S. Chung, and S. Shin, 2001b, Efficient calculation of a partial derivative wavefield using reciprocity for seismic imaging and inversion: Geophysics, **66**, 1856–1863.

Shipp, R. M., and S. C. Singh, 2002, Two-dimensional full wavefield inversion of wide-aperture marine seismic streamer data: Geophysical Journal International, **151**, 325–344.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

———, 1986, A strategy for nonlinear elastic inversion of seismic reflection data: Geophysics, **51**, 1893–1903.

———, 1988, Theoretical background for the inversion of seismic waveforms, including elasticity and attenuation: Birkhäuser Basel.

Tarantola, A., and B. Valette, 1982, Generalized nonlinear inverse problems solved using the least squares criterion: Reviews of Geophysics, **20**, 219–232.

Vigh, D., K. Jiao, D. Watts, and D. Sun, 2014, Elastic full-waveform inversion application using multicomponent measurements of seismic data collection: Geophysics, **79**, R63–R77.

Vigh, D., E. W. Starr, and J. Kapoor, 2009, Developing earth models with full waveform inversion: The Leading Edge, **28**, 432–435.

Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889–901.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1–WCC26.

Wang, T., and J. Cheng, 2017, Elastic full waveform inversion based on mode decomposition: the approach and mechanism: Geophysical Journal International, **209**, 606–622.

Wang, Y., L. Dong, Y. Liu, and J. Yang, 2016, 2D frequency-domain elastic full-waveform inversion using the block-diagonal pseudo-Hessian approximation: Geophysics, **81**, R247–R259.

Zhang, Q., and G. A. McMechan, 2010, 2D and 3D elastic wavefield vector decomposition in the wavenumber domain for VTI media: Geophysics, **75**, D13–D26.

Source encoding in multi-parameter full waveform inversion

G. Matharu [1]

Abstract

Source encoding techniques alleviate the computational burden of sequential-source full waveform inversion (FWI) by considering multiple sources simultaneously rather than independently. The reduced data volume requires fewer forward/adjoint simulations per non-linear iteration. Applications of source-encoded full waveform inversion (SEFWI) have thus far focused on mono-parameter acoustic inversion. We extend SEFWI to the multi-parameter case with applications presented for elastic isotropic inversion. Estimating multiple parameters can be challenging as perturbations in different parameters can prompt similar responses in the data. We investigate the relationship between source encoding and parameter trade-off by examining the multi-parameter source-encoded Hessian. Hessian probing is employed to demonstrate the convergence of the expected source-encoded Hessian, to the Hessian of conventional FWI. The convergence implies that the parameter trade-off in SEFWI is comparable to that observed in FWI. A series of synthetic inversions are conducted to establish the feasibility of source-encoded multi-parameter FWI. We demonstrate that SEFWI requires fewer overall simulations than FWI to achieve a target model error for various first-order optimization methods. An inversion for spatially inconsistent $P-$ (α) and S-wave (β) velocity models, corroborates the expectation of comparable parameter trade-off in SEFWI and FWI.

[1] Email: gian@ualberta.ca

10.1 Introduction

Since its conception in the 1980s (Lailly, 1983; Tarantola, 1984, 1986; Mora, 1987), full waveform inversion (FWI) has matured from a mathematical concept, to a viable imaging technique used to estimate physical parameters in the Earth's subsurface. The advent of modern supercomputers coupled with algorithmic advances have led to a flurry of successes in FWI in both exploration (Pratt et al., 1998; Shipp and Singh, 2002; Sears et al., 2008; Virieux and Operto, 2009; Brossier et al., 2009; Krebs et al., 2009; Prieux et al., 2013a) and global (Tape et al., 2009; Fichtner et al., 2009; Zhu et al., 2015) seismology.

The computational cost of FWI remains a limiting factor for large scale 3D applications on real data. As practitioners accommodate larger and larger datasets, efficient algorithms are crucial to ensuring FWI remains tractable. Potential transitions to more complex physics (e.g. acoustic to elastic) further compound the computational cost. The bulk of the computational expense in FWI originates from computing numerical solutions to multiple partial differential equations *per source* at each iteration. The linear dependence of the cost on the number of sources hampers the scalability of FWI for large datasets. Source encoding is an approach that effectively reduces the dimensionality of the data (Romero et al., 2000; Krebs et al., 2009). By considering multiple simultaneously rather than independently, source encoding reduces the computational cost of an FWI iteration.

Source encoding was originally proposed by Romero et al. (2000) to reduce the cost of shot-record migration. They replaced individual sources by a smaller number of encoded sources, where each encoded source represents a weighted linear combination of individual sources. The weights —known as encoding functions— were chosen as random phase-shifts and are necessary to reduce cross-talk artefacts in the corresponding migration image. Cross-talk artefacts arise from the interaction of different sources with one another in the imaging condition. Romero et al. (2000) significantly reduced the cost of shot-record migration whilst maintaining an acceptable image quality.

Simultaneous sources (without source encoding) were later utilized in global tomography (Capdeville et al., 2005). While the method demonstrated success in synthetic tests, real data examples were hindered by missing data. Vigh and Starr (2008) synthesized plane-wave gathers for acoustic FWI using a deterministic form of time-shift encoding. Krebs et al. (2009) extended source-encoding to FWI and presented two significant results. The first, was the adoption of polarity encoding functions that had the advantageous property of not increasing simulation time, unlike phase/time-shift based encoding schemes. The second, was that by randomizing the encoding functions at each iteration, cross-talk artefacts were almost entirely eliminated from the inverted models. Additional studies in source encoded migration/FWI (SEFWI) have explored the properties of cross-talk (Schuster et al., 2011; Ben-Hadj-Ali et al., 2011), strategies for non fixed-spread receivers (Routh et al., 2011; Choi and Alkhalifah, 2012), the use of second-order optimization methods (Anagaw and Sacchi, 2014; Castellanos et al., 2015), and stochastic optimization methods in place of source encoding (Haber et al., 2012; van Leeuwen and Herrmann, 2013).

The estimation of multiple independent parameters with limited data, poses a significant challenge in multi-parameter FWI. Changes in different physical properties in the Earth's subsurface, can provoke similar responses in the data. This introduces trade-off between the parameters during inversion, meaning that it can be difficult, or impossible to resolve different parameters uniquely. The model, model parametrization, acquisition geometry, and data, all affect the resolvability of individual parameters (Tarantola, 1986; Pratt et al., 1998; Operto et al., 2013). Information on the resolution of model parameters is contained in the Hessian of the objective function. Ideally, second-order optimization would be used to account for the Hessian during computation of the model updates. While second-order optimization methods have been explored in FWI (Epanomeritakis et al., 2008; Métivier et al., 2013; Anagaw and Sacchi, 2014; Pan et al., 2016), they are computationally expensive and do not consistently demonstrate improvements over quasi-Newton methods such as L-BFGS. In lieu of incorporating the Hessian, parameter trade-off can be limited through a combination of data-driven inversion strategies (Shipp and Singh, 2002; Sears et al., 2008; Prieux et al., 2013a,b) and appropriate selection of the inversion parameters (Tarantola, 1986; Plessix and Cao, 2011; Köhn et al., 2012; Operto et al., 2013; Gholami et al., 2013; Alkhalifah and douard Plessix, 2014). Understanding parameter trade-off is necessary for accurate model appraisal, particularly when first-order optimization methods are used.

In this paper, we investigate the feasibility of source-encoded elastic full waveform inversion. Emphasis is placed on understanding the influence of source encoding on the inversion of multiple parameters. Specifically, we seek to determine how source encoding affects parameter trade-off in multi-parameter inversion. While our study focuses on the isotropic, elastic case, our treatment is independent of a specific parametrization so as to be applicable to the general case of multi-parameter inversion.

The paper is structured as follows. Section 2 provides a brief review of FWI, source-encoded FWI, and the optimization algorithms associated with each method. Section 3 introduces challenges associated with multi-parameter inversion. The multi-parameter Hessian, with and without source encoding, is examined to determine the influence of source encoding on parameter trade-off. Section

4 presents a series of numerical experiments catered towards testing specific components of source-encoded FWI. We present results describing the efficiency gain, parameter trade-off, and flexibility of SEFWI relative to FWI. Conclusions from the study are presented in section 5.

10.2 Theory: Full waveform inversion

Full waveform inversion can be formulated mathematically as a PDE-constrained optimization problem of the form

$$\begin{aligned} \underset{m}{\text{minimize}} \quad & J(m), \\ \text{subject to} \quad & F(u, m) = 0, \end{aligned} \tag{10.1}$$

where the functional $J(m)$, is dependent on model parameters m. u is known as the state variable and $F(u, m)$ is the state or forward equation (Plessix, 2006). In the context of FWI, the forward equation corresponds to the seismic wave equation, represented symbolically it reads as

$$\mathbf{L}(\mathbf{m})\mathbf{u} = \mathbf{s}. \tag{10.2}$$

The state variable is the particle displacement $\mathbf{u}(\mathbf{x}, t)$ excited by an external source $\mathbf{s}(\mathbf{x}, t)$. Time is denoted by $t \in [0, T]$ and $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ denotes spatial coordinates with dimensions $d = 1, 2, 3$. For the sake of brevity, we omit the spatial and temporal dependencies of variables after they are first introduced, provided that no ambiguities arise from the omission. The model parameters $\mathbf{m}(\mathbf{x}) = [m_1(\mathbf{x}), m_2(\mathbf{x}), \ldots, m_{N_p}(\mathbf{x})]^T$, represent N_p independent physical properties of the Earth's subsurface; T denotes the transpose. $\mathbf{L}(\mathbf{m})$ is a linear differential operator that characterizes the seismic wave equation and can accommodate varying degrees of physical complexity e.g. acoustic, elastic, isotropic/anisotropic etc. The physics incorporated into the forward modelling operator $\mathbf{L}(\mathbf{m})$, dictate the physical properties comprised in $\mathbf{m}$. In this study, $\mathbf{L}(\mathbf{m})$ refers to the isotropic, elastic wave equation in the time domain (Aki and Richards, 2002),

$$\mathbf{L}(\rho, \lambda, \mu)\ [\cdot] = \rho(\mathbf{x})\frac{\partial^2}{\partial t^2}[\cdot] - (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\nabla\nabla \cdot [\cdot] - \mu(\mathbf{x})\nabla \times \nabla \times [\cdot], \tag{10.3}$$

where $[\cdot]$ is a place-holder for the variable acted upon by $\mathbf{L}(\mathbf{m})$. ∇ denotes the spatial gradient operator. $\rho(\mathbf{x})$, $\lambda(\mathbf{x})$, and $\mu(\mathbf{x})$ are density and the Lamé parameters, respectively; μ is also known as the shear modulus. The particular choice of $\mathbf{L}(\mathbf{m})$ in eq. (10.3) does not lead to a loss of generality in the forthcoming discussions on FWI and multi-parameter source-encoded FWI.

The misfit functional $J(\mathbf{m})$ —also known as the objective function— quantifies the difference between observed and synthetic data by comparing an observable quantity. The most prevalent choice of objective function for FWI is the least-squares waveform misfit functional

$$J(\mathbf{m}) = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \int_T |\mathbf{u}_s(\mathbf{x}_r, t; \mathbf{m}) - \mathbf{d}_s(\mathbf{x}_r, t)|^2 \, \mathrm{d}t. \tag{10.4}$$

The simulated multi-component data $\mathbf{u}_s(\mathbf{x}_r, t; \mathbf{m})$ are recorded at the *r-th* receiver and generated by the *s-th* source $\mathbf{s}_s$ for model $\mathbf{m}$. A similar definition is applicable for the observed data $\mathbf{d}_s(\mathbf{x}_r, t)$. N_s and N_r denote the number of sources and receivers, respectively. The least-squares waveform misfit (10.4) is a non-linear functional owing to the non-linear dependence of $\mathbf{u}$ on $\mathbf{m}$ (Virieux and Operto, 2009).

Non-linear functionals can be minimized via iterative, gradient-based minimization algorithms (Plessix, 2006). This approach was first demonstrated by Lailly (1983) and Tarantola (1984) who formulated FWI as a linearized inverse problem. Solutions to eq. (10.1) can be estimated by iteratively updating the model parameters via

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \nu^k \delta \mathbf{m}^k, \tag{10.5}$$

where k denotes the iteration number, ν^k is a scalar step length, and the model perturbation/update is $\delta\mathbf{m}^k(\mathbf{x}) = [\delta m_1(\mathbf{x}), \delta m_2(\mathbf{x}), \ldots, \delta m_{N_p}(\mathbf{x})]^T$. A suitable ν_k can be estimated using various line-search algorithms (Nocedal and Wright, 2006). In optimization literature, the model update $\delta\mathbf{m}^k$ is referred to as the search/descent direction and can be derived from the gradient of the objective function with respect to the model parameters.

FWI gradient

The gradient of $J(\mathbf{m})$ with respect to $\mathbf{m}$, $\nabla_m J$, can be calculated efficiently using the adjoint-state method; for a complete description of the method, the reader is referred to Tarantola (1984); Mora (1987); Tromp et al. (2005); Plessix (2006); Fichtner et al. (2006). Gradient computation via the adjoint-state method requires solutions to the forward (10.2) and adjoint wave equation

$$\mathbf{L}^\dagger(\mathbf{m})\mathbf{u}^\dagger = \mathbf{s}^\dagger. \tag{10.6}$$

The linear differential operator $\mathbf{L}^\dagger$ characterizes the adjoint wave equation, $\mathbf{u}^\dagger(\mathbf{x}, t)$ is the adjoint wavefield, and $\mathbf{s}^\dagger(\mathbf{x}, t)$ is known as the adjoint source. The adjoint wave equation adheres to the same physics as the forward wave equation, but operates in reverse time ($T \to 0$) and replaces $\mathbf{s}$ with adjoint source $\mathbf{s}^\dagger$. Initial conditions on displacement and velocity are replaced by analogous terminal conditions at time T. The precise form of $\mathbf{s}^\dagger$ is determined by the choice of objective function. The adjoint source for the least-squares waveform misfit functional is

$$\mathbf{s}_s^\dagger(\mathbf{x},t) = \sum_{r=1}^{N_r} \left[\mathbf{u}_s(\mathbf{x}_r,t;\mathbf{m}) - \mathbf{d}_s(\mathbf{x}_r,t)\right] \delta(\mathbf{x}-\mathbf{x}_r). \tag{10.7}$$

The adjoint wave equation backpropagates the waveform residuals from the receiver locations into the subsurface. The same modelling code can be used to solve the forward (10.2) and adjoint (10.6) wave equations. Given $\mathbf{u}$ and $\mathbf{u}^\dagger$, the time-domain expression for the derivative of J with respect to $\mathbf{m}$ is

$$\nabla_m J(\mathbf{x}) = -\sum_{s=1}^{N_s} \int_T \mathbf{u}_s^\dagger(\mathbf{x},t) \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{m}} \mathbf{u}_s(\mathbf{x},t) \ \mathrm{d}t. \tag{10.8}$$

Computation of $\nabla_m J$ requires N_s forward simulations and N_s adjoint simulations at each iteration. The computational cost of an FWI iteration therefore grows linearly with N_s. This linear dependence can become prohibitive when N_s is large. Source encoding effectively reduces the number of sources by considering multiple sources simultaneously rather than independently.

10.2.1 Source-encoded FWI

The linear dependence of $\mathbf{u}$ with respect to $\mathbf{s}$, allowed Krebs et al. (2009) to reformulate eq. (10.4) to accommodate simultaneous sources under the assumption of a fixed-spread acquisition. The revised misfit functional may be expressed as

$$\hat{J}(\mathbf{m}) = \frac{1}{2} \sum_{e=1}^{N_e} \sum_{r=1}^{N_r} \int_T \left|\hat{\mathbf{u}}_e(\mathbf{x}_r,t;\mathbf{m}) - \hat{\mathbf{d}}_e(\mathbf{x}_r,t)\right|^2 \ \mathrm{d}t, \tag{10.9}$$

where $\hat{\mathbf{d}}_e$ are the encoded data and $\hat{\mathbf{u}}_e$ are the synthetic data generated by encoded source $\hat{\mathbf{s}}_e$. The number of encoded sources N_e is selected such that $N_e < N_s$. We use a circumflex to signify source-encoded variables or those associated with SEFWI. Source encoding reduces the data volume by a factor equal to N_s/N_e thereby reducing the number of PDE solves required per iteration. Maximal data compression is achieved when every individual sources is combined into a single encoded source.

Let S denote the set containing all the sources in a given acquisition. We synthesize N_e encoded sources from N_e mutually disjoint subsets of S. Formally, $S = \cup_{i=1}^{N_e} S_i$ where $S_i \cap S_j = \emptyset,\ \{\forall i, j =$

$1, \ldots, N_e, i \neq j\}$. Mutually disjoint subsets ensure that individual sources are not repeated over multiple encoded sources. This restriction is not essential to the formulation of SEFWI; however, we impose it to simplify the forthcoming treatment of SEFWI. Following this definition, $\hat{\mathbf{s}}_e$ and $\hat{\mathbf{d}}_e$ are formed from the linear combinations,

$$\hat{\mathbf{s}}_e(\mathbf{x}, t) = \sum_{s \in S_e} q_s^e(t) * \mathbf{s}_s(\mathbf{x}, t), \tag{10.10}$$

$$\hat{\mathbf{d}}_e(\mathbf{x}, t) = \sum_{s \in S_e} q_s^e(t) * \mathbf{d}_s(\mathbf{x}, t), \tag{10.11}$$

where $q_s^e(t)$ are source-specific encoding functions for the *e-th* encoded source. Convolution in the time domain is denoted by $*$. Encoding functions are discussed in detail in section 2.2.

SEFWI gradient

The SEFWI gradient is computed using the adjoint-state method in conjunction with an adjoint source derived from eq. (10.9),

$$\hat{\mathbf{s}}_e^{\dagger}(\mathbf{x}, t) = \sum_{r=1}^{N_r} \Big[\hat{\mathbf{u}}_e(\mathbf{x}_r, t; \mathbf{m}) - \hat{\mathbf{d}}_e(\mathbf{x}_r, t) \Big] \, \delta(\mathbf{x} - \mathbf{x}_r). \tag{10.12}$$

Source encoding introduces cross-talk artefacts into the gradient that are a consequence of zero-lag correlations (10.8) between forward and adjoint wavefields that do not correspond to the same source. For the case of pure simultaneous sources, i.e. if $q_s^e(t) = 1 \; \{\forall s \in S_e, e = 1, \ldots, N_e\}$, the derivative of $\hat{J}$ with respect to $\mathbf{m}$ is

$$\begin{aligned} \nabla_m \hat{J}(\mathbf{x}) &= -\sum_{e=1}^{N_e} \int_T \hat{\mathbf{u}}_e^{\dagger}(\mathbf{x}, t) \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{m}} \hat{\mathbf{u}}_e(\mathbf{x}, t) \; \mathrm{d}t, \qquad (10.13) \\ &= -\sum_{e=1}^{N_e} \int_T \sum_{i \in S_e} \mathbf{u}_i^{\dagger}(\mathbf{x}, t) \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{m}} \sum_{j \in S_e} \mathbf{u}_j(\mathbf{x}, t) \; \mathrm{d}t, \\ &= \nabla_m J(\mathbf{x}) - \underbrace{\sum_{i=1}^{N_s} \sum_{\substack{j=1 \\ j \neq i}}^{N_s} \int_T \mathbf{u}_j^{\dagger}(\mathbf{x}, t) \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{m}} \mathbf{u}_i(\mathbf{x}, t) \; \mathrm{d}t}_{\text{Cross-talk term}}. \qquad (10.14) \end{aligned}$$

The simplification in eq. (10.14) is valid when subsets S_i are mutually disjoint. The second term on the right hand side of eq. (10.14) represents cross-talk artefacts that compromise the accuracy of the desired gradient. For pure simultaneous sources, the cross-talk artefacts stack as coherent noise over the course of SEFWI iterations resulting in inaccurate models (Romero et al., 2000; Krebs et al., 2009). The influence of cross-talk artefacts can be ameliorated via source encoding; further details are described in the following section.

10.2.2 Source encoding

The frequency domain representation of $\nabla_m \hat{J}$ with general source-encoding is

$$\nabla_m \hat{J}(\mathbf{x}) = -\sum_{e=1}^{N_e} \Bigg[\sum_{i \in S_e} \left\langle Q_i^e(\omega) U_i{}^{\dagger}(\mathbf{x},\omega),\ Q_i^e(\omega) \frac{\partial \mathbf{L}}{\partial \mathbf{m}} U_i(\mathbf{x},\omega) \right\rangle_\omega \\ \underbrace{+ \sum_{i \in S_e} \sum_{\substack{j \in S_e \\ j \neq i}} \left\langle Q_i^e(\omega) U_i{}^{\dagger}(\mathbf{x},\omega),\ Q_j^e(\omega) \frac{\partial \mathbf{L}}{\partial \mathbf{m}} U_j(\mathbf{x},\omega) \right\rangle_\omega \Bigg]}_{\text{Cross-talk term}}, \tag{10.15}$$

where $Q_i^e(\omega) = \mathscr{F}\{q_i^e(t)\}$, $U(\mathbf{x},\omega) = \mathscr{F}\{\mathbf{u}(\mathbf{x},t)\}$, $U^{\dagger}(\mathbf{x},\omega) = \mathscr{F}\{\mathbf{u}^{\dagger}(\mathbf{x},t)\}$, and $\mathscr{F}$ is the Fourier transform operator. The frequency-space inner product $\langle \cdot,\cdot \rangle_\omega$ between two arbitrary complex-valued functions f and g, is defined as

$$\langle f, g \rangle_\omega := \int_\omega \bar{f}(\omega) g(\omega)\ \mathrm{d}\omega. \tag{10.16}$$

Complex conjugation is denoted by a bar above a variable. When the encoding functions form an orthonormal basis i.e. $\bar{Q}_i^e(\omega) Q_j^e(\omega) = \delta_{ij}$, eq. (10.15) reduces to the standard FWI gradient. In practice, we seek random encoding functions that exhibit the property

$$\mathbb{E}[\bar{Q}_i^e(\omega) Q_j^e(\omega)] = \delta_{ij}. \tag{10.17}$$

Equation (10.17) states that the expected inner product between any two random encoding functions is a Kronecker delta function. Previous studies have used this condition to select appropriate encoding functions, some of which we present here.

Random time-shift encoding employs functions of the form $Q_s(\omega) = e^{i\omega\tau_s}$, where τ_s is a random time shift associated with the *s-th* source. Schuster et al. (2011) demonstrated that for random variables $\tau_s \sim \mathscr{N}(0, \sigma^2)$, that are independent and identically distributed $\forall s = 1, \ldots, N_s$, the expected inner product between two random encoding functions is given by

$$\mathbb{E}[\bar{Q}_i(\omega)Q_j(\omega)] = \begin{cases} 1, & \text{if } i = j \\ e^{-\omega^2\sigma^2}, & \text{if } i \neq j. \end{cases} \tag{10.18}$$

The cross-term ($i \neq j$) stems from the characteristic function of the random variable τ_s. Schuster et al. (2011) conclude that cross-talk noise may be attenuated if the variance of the time shifts is much greater than the period $2\pi/\omega$. Lower frequency signals, therefore require larger time shifts to reduce cross-talk terms in the gradient.

Plane-wave encoding is a specialized form of time-shift encoding where the time shifts are controlled by ray parameter $\mathbf{p}$ and source position $\mathbf{x}_s$ (Vigh and Starr, 2008). The encoding functions are given by

$$Q_s(\mathbf{x}_s, \mathbf{p}, \omega) = e^{i\omega\mathbf{p}\cdot(\mathbf{x}_s - \mathbf{x}_0)}, \tag{10.19}$$

where $\mathbf{x}_0$ is the location of the plane-wave origin. Plane-wave encoding uses all of the available sources within an acquisition to synthesize plane-wave gathers. The summation over encoded sources in eq. (10.9) is replaced by a summation over ray parameters. Cross-talk artefacts can be minimized by sampling over a sufficiently broad range of ray parameters (Vigh and Starr, 2008).

Time-shift encoding schemes extend the duration of seismic traces resulting in longer simulation times in FWI. While the time shifts are typically much shorter than the duration of a trace, the increased simulation time is undesirable. Polarity encoding avoids this by opting for encoding functions of the form

$$Q_i(\omega) = p_i. \tag{10.20}$$

The discrete random variable p_i takes values of +1 or -1 with equal probability. The orthonormality condition of eq. (10.17) is satisfied and simulation time remains unaffected.

Source encoding alone is not sufficient to suppress cross-talk artefacts in SEFWI. The encoding functions should be randomized at regular intervals, ideally after every iteration (Krebs et al., 2009). Randomizing the encoding functions ensures that the cross-talk terms stack as incoherent noise over the course of SEFWI. In a theoretical analysis of cross-talk, Schuster et al. (2011) defined the cross-talk signal-to-noise ratio (c-SNR) as

$$\text{c-SNR} = \frac{\|v\|}{\|\hat{v}^{(est)} - v\|}. \tag{10.21}$$

In this definition $v(\mathbf{x})$ represents an arbitrary scalar field and $\hat{v}(\mathbf{x})$ is its source-encoded equivalent. The gradient is one such example. The L_2 norm ($\|\cdot\|$) follows a standard definition:

$$\|\psi\| := (\int_\Omega |\psi(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x})^{\frac{1}{2}}, \tag{10.22}$$

for an arbitrary real-valued function $\psi(\mathbf{x})$. $\hat{v}^{(est)}$ is a stack of N random realizations of a source-encoded scalar field, where realization $\hat{v}^{(i)}$ is an instance of $\hat{v}$ for a particular random source encoding. Formally,

$$\hat{v}^{(est)} = \sum_{i=1}^{N} \hat{v}^{(i)}. \tag{10.23}$$

Schuster et al. (2011) demonstrated that the c-SNR grows $\propto \sqrt{N}$. The effect of stacking random realizations is mimicked by randomizing encoding functions at each iteration of SEFWI. Applied to source-encoded migration, the number of iterations required to achieve a c-SNR comparable to sequential source migration, is equal to the number of individual sources in an encoded source.

10.2.3 Gradient-based optimization

In the vicinity of an initial model $\mathbf{m}^0$, $J(\mathbf{m}^0)$ can be approximated by a locally quadratic function following a second-order Taylor expansion. The perturbation $\delta\mathbf{m}$ that minimizes the quadratic approximation is obtained by solving the Newton system of equations, represented symbolically as

$$\mathbf{H}\delta\mathbf{m} = -\mathbf{g}, \tag{10.24}$$

where $\mathbf{g} = \nabla_m J$ and $\mathbf{H} = \nabla_m^2 J$ denote the gradient and Hessian of $J(\mathbf{m}^0)$, respectively. Henceforth, we adopt letter symbols for the FWI/SEFWI gradients and Hessians for readability. The Hessian for the least-squares waveform misfit functional is

$$\begin{aligned}\mathbf{H}(\mathbf{x},\mathbf{y}) = \sum_{s=1}^{N_s}\sum_{r=1}^{N_r} \Bigg[& \int_T \frac{\partial \mathbf{u}_s(\mathbf{x}_r,t)}{\partial \mathbf{m}(\mathbf{x})} \frac{\partial \mathbf{u}_s(\mathbf{x}_r,t)}{\partial \mathbf{m}(\mathbf{y})} \, \mathrm{d}t \\ & + \int_T \frac{\partial^2 \mathbf{u}_s(\mathbf{x}_r,t)}{\partial \mathbf{m}(\mathbf{x}) \partial \mathbf{m}(\mathbf{y})} [\mathbf{u}_s(\mathbf{x}_r,t) - \mathbf{d}_s(\mathbf{x}_r,t)] \, \mathrm{d}t \Bigg].\end{aligned} \tag{10.25}$$

The first term on the right hand side of eq. (10.25) is the Gauss-Newton Hessian $\mathbf{H}^a$. The second term relates to second-order scattering effects and is routinely neglected (Pratt et al., 1998). $\mathbf{H}^a$ acts as a defocusing operator and smears perturbations in space. Conversely, the inverse Hessian acts as a focusing operator. The nature of the focusing/defocusing is linked to the source-receiver acquisition geometry (Pratt et al., 1998). In multi-parameter inversion, $\mathbf{H}^a$ carries information pertaining to the trade-off between physical parameters. The multi-parameter Hessian is explored in section 3.1.

The non-linearity of the FWI objective function introduces numerous local minima into the search space. FWI performs a local optimization; therefore, initial models that are not sufficiently close to the global minimum may converge to a local minima instead. Provided that the inversion is initiated with a model sufficiently close to the true model, the objective function can be minimized using gradient-based optimization algorithms.

Gradient-based optimization algorithms can be divided into first- and second-order methods, contingent on whether the Hessian is utilized by algorithm. First-order techniques rely solely on gradient information to generate search directions, whereas second-order techniques use the Hessian in some capacity. The method of steepest descent (SD) sets the search direction as the negative gradient; however, search directions may be poorly scaled and result in slow convergence. The non-linear conjugate gradient algorithm (NLCG) computes search directions as a linear combination of the current and prior iteration gradients. NLCG provides higher convergence rates than SD while remaining easy to implement.

Convergence rates in SD and NLCG can be accelerated through the use of preconditioners. A preconditioned steepest descent step takes the form

$$\delta\mathbf{m} = -\mathbf{P}^{-1}\mathbf{g}, \tag{10.26}$$

where $\mathbf{P}$ is a preconditioning operator. A comparison of eqs. (10.24) and (10.26), indicates that the Hessian operator is an ideal preconditioner. In practice, $\mathbf{P}$ is chosen to mimic the action of $\mathbf{H}$ while being significantly easier to compute and invert. The Hessian contributes to balancing amplitudes in the gradient that are deficient due to geometrical spreading or inadequate illumination. The pseudo-Hessian approximation is popular choice of preconditioner that fulfils the aforementioned criteria without any additional cost (Shin et al., 2001). Quasi-Newton algorithms, such as the BFGS algorithm or its limited memory variant (L-BFGS), derive approximations to the Hessian, or its inverse, using gradient information from the previous m gradients (Nocedal and Wright, 2006).

Second-order methods utilize the Hessian to solve eq. (10.24) and garner search directions. Direct computation of the Hessian requires access to the Fréchet derivatives $\frac{\partial \mathbf{u}}{\partial \mathbf{m}}$, which in turn requires knowledge of the complete time-history of receiver-side Green's functions for every receiver (Chen et al., 2007). N_r forward simulations are required to compute numerical receiver Green's functions. The computational resources required to construct and store the Hessian are not viable for problems of realistic size. Truncated Newton methods are an economical alternative to full Newton methods. Truncated Newton methods apply the linear conjugate-gradient algorithm to solve eq. (10.24) at each iteration (Epanomeritakis et al., 2008; Métivier et al., 2013, 2014; Anagaw and Sacchi, 2014). The inner conjugate-gradient loop requires the computation of Hessian-vector products. Hessian-vector products can be calculated using second-order adjoint methods (Fichtner and Trampert, 2011a) or with finite-difference approximations. The action of the Hessian on a vector can be computed at a cost comparable to that of a gradient calculation. Métivier et al. (2013, 2014) demonstrated that in complex cases where multi-scattered phases play an important role, truncated Newton methods can provide better inverted models.

Model regularization

FWI is an ill-posed problem meaning an infinite number of models can fit the data equally well (Virieux and Operto, 2009). Model regularization is included explicitly into the objective function. The role of regularization is to stabilize the inversion and make it more well-posed. Model regularization constrains model updates by imposing prior assumptions on the model. In this study, we implement a form of Tikhonov regularization that penalizes deviatoric perturbations from a prior model,

$$R(\mathbf{m}) = \frac{\gamma}{2} \sum_{p}^{N_p} \| m_p - m_p^{prior} \|^2. \tag{10.27}$$

A tunable hyperparameter γ controls the contribution of the regularization term relative to the data misfit. The prior model in eq. (10.27) is taken as the initial model input to FWI/SEFWI.

Optimization algorithms for source-encoded FWI

The source-encoded Hessian $\hat{\mathbf{H}} = \nabla_m^2 \hat{J}$ can vary considerably between iterations where the source encoding is randomized. For this reason, it is not suitable to apply conventional NLCG or L-BFGS. The conjugacy condition cannot be guaranteed and therefore search directions generated by NLCG are not assured to be conjugate pairs. Moghaddam et al. (2013) proposed a heuristic alternative that formed search directions as a weighted sum of prior gradients. Their approach demonstrated higher convergence rates relative to SD when applied to acoustic SEFWI.

We implement the hybrid NLCG and L-BFGS algorithms demonstrated by Huang and Schuster (2012) and Castellanos et al. (2015), respectively. The hybrid algorithms amount to regular-restart versions of their conventional counterparts. After every M-th iteration, the optimization history is reset and the source encoding is randomized. The source encoding does not vary between restart intervals. In this study, we deploy SEFWI with SD and restart variants of NLCG and L-BFGS. We do not differentiate between regular and restart versions of NLCG/L-BFGS in the text. The reader may assume that NLCG/L-BFGS applied to SEFWI corresponds to the restart versions described in this section.

10.3 Multi-parameter inversion

Prior applications of source encoding in migration and FWI have focused on single parameter inversion under the constant-density, acoustic approximation (Romero et al., 2000; Vigh and Starr, 2008; Krebs et al., 2009; Dai et al., 2012; Anagaw and Sacchi, 2014; Castellanos et al., 2015). When considering anisotropic or elastic representations of the Earth, multiple independent model parameters are required to characterize the subsurface. An isotropic elastic medium is adequately described by 3 independent parameters; a potential parametrization is in terms of density (ρ) and the Lamé parameters (λ, μ).

An ideal model parametrization consists of a set of physical parameters that are uniquely resolvable (Tarantola, 1986; Operto et al., 2013). The extent to which a model perturbation can be resolved uniquely, is dictated by the choice of model parametrization, acquisition geometry, background model, and bandwidth of the data. A poor choice of model parametrization or inadequate subsurface illumination can lead to ambiguities between different parameters. Parameter trade-off is the phenomena where changes in different parameters elicit similar responses in the data. A classic example is the velocity-depth ambiguity associated with reflection travel times. Further complications arise when model parameters lie in the null space of the problem. Perturbations to these parameters will not register in the data making such parameters unresolvable.

Within a given parametrization, certain parameters have a greater influence on the data than others. A good parametrization prioritises the accurate reconstruction of parameters that most strongly influence the kinematics of the data. For example, P-wave velocity controls the kinematics of compressional waves, whereas density primarily influences reflection amplitudes. A parametrization that allows for the broadband reconstruction of P-wave velocity should be favoured. Tarantola (1986) compared radiation patterns derived from the Born approximation to assess parameter trade-off and resolution. The study concluded that a parametrization of density (ρ), P-wave velocity (α), and S-wave velocity (β) was suitable for broadband reconstruction of α while limiting parameter trade-offs.

10.3.1 Multi-parameter Hessian

Parameter trade-offs manifest mathematically in the multi-parameter Hessian. The Hessian carries information pertaining to the strength of parameter trade-offs along with the spatial resolution afforded by the acquisition geometry. Neglecting the Hessian in multi-parameter inversion introduces inaccuracies into the inversion due to erroneous inter-parameter mappings. The multi-parameter Hessian operator exhibits a block structure and may be expressed in matrix form as

$$\mathbf{H}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \mathbf{H}_{m_1 m_1}(\mathbf{x},\mathbf{y}) & \dots & \mathbf{H}_{m_1 m_p}(\mathbf{x},\mathbf{y}) \\ \vdots & \ddots & \\ \mathbf{H}_{m_p m_1}(\mathbf{x},\mathbf{y}) & & \mathbf{H}_{m_p m_p}(\mathbf{x},\mathbf{y}) \end{bmatrix}. \tag{10.28}$$

The Newton equations in terms of the multi-parameter Hessian operator are

$$\sum_{j=1}^{N_p} \int \mathbf{H}_{m_i m_j}(\mathbf{x},\mathbf{y}) \delta m_j(\mathbf{y}) \ \mathrm{d}\mathbf{y} = -g_i(\mathbf{x}). \tag{10.29}$$

Equation (10.29) states that the gradient for the $i\text{-}th$ model parameter, is a linear combination of the true model perturbations, weighted by the relevant block elements from the Hessian. Second-order model updates are obtained by applying the inverse Hessian operator to the gradient vector. Due to the expense of Newton based methods, the Hessian in eq. (10.29) is often replaced with a diagonal preconditioning operator to give

$$\int \mathbf{P}_{m_i m_i}(\mathbf{x},\mathbf{y}) \delta m_i(\mathbf{y}) \ \mathrm{d}\mathbf{y} \approx -g_i(\mathbf{x}). \tag{10.30}$$

The lack of off-diagonal contributions ($i \neq j$) in $\mathbf{P}$ is a source of uncorrected inter parameter mappings that manifest during inversion. When inter-parameter mappings are not corrected, it is important to understand the nature of such mappings for proper model appraisal. A concern arises when using first-order optimization for SEFWI. The cross-talk artefacts in the SEFWI gradient will map across multiple parameters contributing to additional parameter trade-offs. To better understand the behaviour of parameter trade-off in SEFWI, we examine the source-encoded, multi-parameter Hessian.

10.3.2 Source-encoded multi-parameter Hessian

In the frequency domain, the source-encoded Gauss-Newton Hessian ($\hat{\mathbf{H}}^a$) can be written as

$$\hat{\mathbf{H}}^a(\mathbf{x},\mathbf{y}) = \sum_{e=1}^{N_e}\sum_{r=1}^{N_r}\left\langle \sum_{i\in S_e} Q_i^e(\omega)\frac{\partial \mathbf{u}_i(\mathbf{x}_r,\omega)}{\partial \mathbf{m}(\mathbf{x})}, \sum_{j\in S_e} Q_j^e(\omega)\frac{\partial \mathbf{u}_j(\mathbf{x}_r,\omega)}{\partial \mathbf{m}(\mathbf{y})}\right\rangle_\omega, \tag{10.31}$$

where $\frac{\partial \mathbf{u}_i(\mathbf{x}_r,\omega)}{\partial \mathbf{m}(\mathbf{x})}$ are the Fourier transformed Fréchet derivatives. Equation (10.31) can be simplified to

$$\hat{\mathbf{H}}^a(\mathbf{x},\mathbf{y}) = \mathbf{H}^a + \sum_{e=1}^{N_e}\sum_{r=1}^{N_r}\sum_{i\in S_e}\sum_{\substack{j\in S_e\\ j\neq i}}\left\langle Q_i^e(\omega)\frac{\partial \mathbf{u}_i(\mathbf{x}_r,\omega)}{\partial \mathbf{m}(\mathbf{x})}, Q_j^e(\omega)\frac{\partial \mathbf{u}_j(\mathbf{x}_r,\omega)}{\partial \mathbf{m}(\mathbf{y})}\right\rangle_\omega, \tag{10.32}$$

once again assuming that the subsets S_i are mutually disjoint. The cross-talk terms in the source-encoded Hessian are comparable to those in the source-encoded gradient (10.15). This implies that the cross-talk in $\hat{\mathbf{H}}^a$ can also be attenuated by selecting orthonormal encoding functions (10.17). The symbolic representation of the Newton equations in SEFWI is,

$$(\mathbf{H}^a + \mathbf{H}^c)\delta\mathbf{m} = -(\mathbf{g} + \mathbf{g}^c), \tag{10.33}$$

where $\mathbf{H}^c$ and $\mathbf{g}^c$ are the cross-talk components of the source-encoded Hessian and gradient, respectively. $\mathbf{H}^c$ will map model perturbations into $\mathbf{g}^c$. If $\delta\mathbf{m}$ is computed using first-order gradient techniques, i.e. by neglecting $\hat{\mathbf{H}}^a$, estimates of $\delta\mathbf{m}$ will exhibit erroneous inter-parameter mappings associated with $\mathbf{g}^c$. To verify that source encoding can be used to suppress cross-talk in the Hessian, we perform an analysis that involves probing the multi-parameter Hessian.

10.3.3 Hessian probing

In the vicinity of the true model, the resolvability of a model perturbation can be assessed by computing

$$\mathbf{H}^{-g}\mathbf{H}\delta\mathbf{m}_{true} = \delta\mathbf{m} \tag{10.34}$$

where $\mathbf{H}^{-g}$ is the generalized inverse of the Hessian, $\delta\mathbf{m}_{true}$ is a true model perturbation, and $\delta\mathbf{m}$ is an estimated model perturbation (Fichtner and Trampert, 2011b). $\mathbf{H}^{-g}\mathbf{H}$ acts as a resolution operator and $\mathbf{H}^{-g}\mathbf{H}\delta\mathbf{m}_{true}$ is as a point spread function (PSF) that describes how model perturbations are smeared in space. After replacing $\mathbf{H}^{-g}$ by an identity, the Hessian acts as a conservative approximation to the true resolution operator. Likewise, $\mathbf{H}\delta\mathbf{m}_{true}$ provides an estimate of the true PSF (Fichtner and Leeuwen, 2015). Henceforth, we use the term PSF to refer to the approximate PSF $\mathbf{H}\delta\mathbf{m}$. PSFs can be interpreted as resolution proxies in the vicinity of the true model or, more generally, as weighted row averages of $\mathbf{H}$ in the discrete case.

For fixed $\delta\mathbf{m}$, we define a realization of $\hat{\mathbf{H}}\delta\mathbf{m}$ as the PSF computed for a particular set of random encoding functions. When the encoding functions satisfy eq. (10.17), the expected PSF in SEFWI is

$$\mathbb{E}[\hat{\mathbf{H}}\delta\mathbf{m}] = \mathbb{E}[\hat{\mathbf{H}}]\delta\mathbf{m} = \mathbf{H}\delta\mathbf{m}. \tag{10.35}$$

The convergence of $\hat{\mathbf{H}} \to \mathbf{H}$ was noted by Tang (2009) and used for efficient access to the Hessian in mono-parameter acoustic migration/FWI. The expected PSF is estimated from an ensemble average over N random realizations of $\hat{\mathbf{H}}\delta\mathbf{m}$,

$$\mathbb{E}[\hat{\mathbf{H}}\delta\mathbf{m}] \approx \frac{1}{N}\sum_{i=1}^{N} \hat{\mathbf{H}}^{(i)}\delta\mathbf{m}. \tag{10.36}$$

As a numerical test, we compute expected PSFs for perturbations in a homogeneous background model. The test focuses on the two parameter case where $\mathbf{m}(\mathbf{x}) = [\alpha(\mathbf{x}), \beta(\mathbf{x})]^T$. The model is discretized on a 100 x 100 grid with a spacing of 10 m. 16 sources and 50 receivers are evenly distributed along the surface. We use model perturbations of the form

$$\delta\mathbf{m}(\mathbf{x}) = [\delta\alpha(\mathbf{x}), \delta\beta(\mathbf{x})]^T = \begin{cases} [c, 0]^T, & \text{for } \mathbf{x} = \mathbf{x}_c \\ [0, 0]^T, & \text{otherwise} \end{cases}$$

where $\mathbf{x}_c$ is the central grid point and c is a constant value, taken as 1% of the background model in this example. Perturbations are applied to one parameter at a time thereby allowing us to target individual block elements of the Hessian operator (10.28). We compute PSFs with the sequential-source Hessian for reference.

Figure 10.1 depicts PSFs associated with $\mathbf{H}$ and $\hat{\mathbf{H}}$. For a small numbers of random realizations, prominent cross-talk artefacts are apparent in the expected PSFs. As the number of random realizations increases, $\mathbb{E}[\hat{\mathbf{H}}\delta\mathbf{m}]$ increasingly resembles $\mathbf{H}\delta\mathbf{m}$. For 64 realizations, the expected PSF is almost identical to the reference PSF. We notice some spurious oscillations in $\mathbf{H}_{\alpha\beta}$ (fig. 10.1 (e)) that are not corrected even with 64 realizations. We attribute these oscillations to boundary related artefacts that result from the relatively small grid size.

Figure 10.2 displays the growth of c-SNR as a function of random realizations in an ensemble. The PSFs associated with the different block components of $\hat{\mathbf{H}}$ exhibit similar convergence behaviour

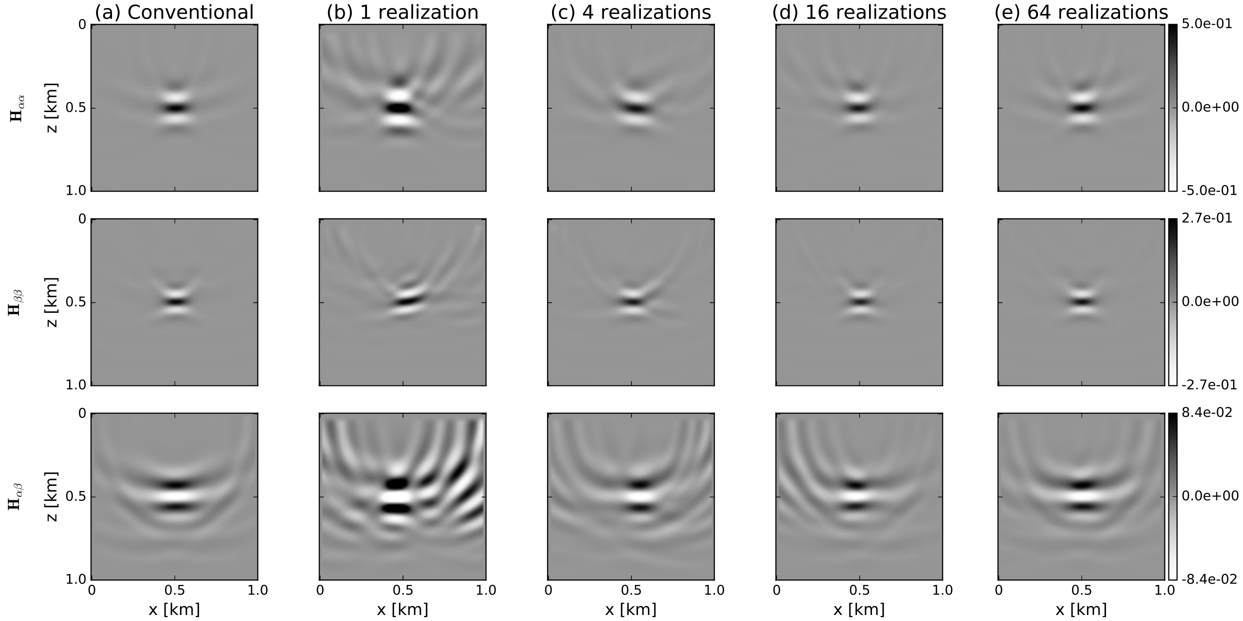


Figure 10.1: A comparison of point spread functions generated for a point scatterer (x = z = 0.5 km) using different block components of the (a) conventional and (b-e) source-encoded Hessian: (top row) $\mathbf{H}_{\alpha\alpha}$, (middle row) $\mathbf{H}_{\beta\beta}$, (bottom row) $\mathbf{H}_{\alpha\beta}$. The ensemble averaged PSFs are presented for a varying number of realizations of the source-encoded Hessian. As the number of realizations increases, cross-talk artefacts are suppressed and the expected PSFs approach the PSFs associated with the conventional Hessian.

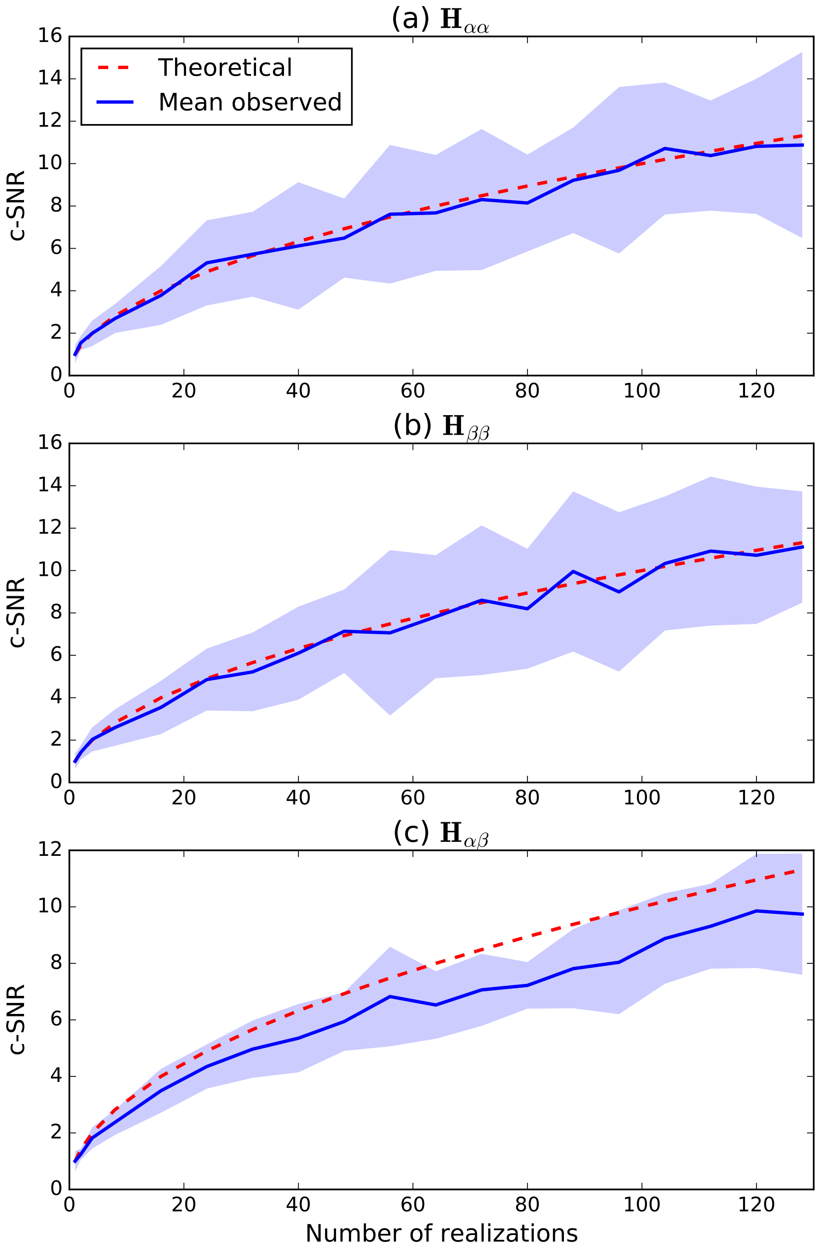


Figure 10.2: PSF c-SNR (solid line) as a function of the number of random realizations in an ensemble. The mean c-SNR for 20 independent trials is plotted with errors bars (solid blue line). The mean c-SNR grows approximately $\propto \sqrt{N}$ (red dashed line). The error bars represent one standard deviation. Each panel is normalized to have c-SNR=1 at the first iteration.

and approximately follow the $\propto \sqrt{N}$ relation defined by Schuster et al. (2011). The bias in the $\mathbf{H}_{\alpha\beta}$ PSF is attributed to the propagation of numerical errors mentioned previously. Hessian-vector products were computed using a finite difference approximation for convenience.

Certain conclusions can be drawn from figures 10.1 and 10.2. Since the cross-talk terms in the Hessian can be attenuated through SEFWI iterations, the parameter trade-off in SEFWI should be comparable to that of FWI. Furthermore, as the cross-talk terms in $\hat{\mathbf{H}}$ are attenuated at the same rate as those in $\hat{\mathbf{g}}$, the number of iterations does not need to be extended to correct for inter-parameter mappings of cross-talk noise. The PSFs corroborate prior studies that claimed the resolution of SEFWI was also comparable to FWI. Due to the property $\mathbb{E}[\hat{\mathbf{H}}\delta\mathbf{m}]$, SEFWI should not exhibit any further sensitivity to model parametrization than already present in FWI.

10.3.4 On the number of inversion parameters

Multi-parameter FWI can invert for independent model parameters sequentially or simultaneously. Sequential inversion can be necessary with Earlier studies opted for sequential inversion when the resolvability of select parameters was limited (Shipp and Singh, 2002). In such cases, inversion strategies prioritise the reconstruction of parameters that have a primary influence on the kinematics. A drawback of sequential inversion is the potential to introduce strong artefacts that result from uncorrected parameter trade-offs. The source of these artefacts was established in earlier discussions on equations 10.29-10.30. If one parameter remains fixed, the value $\delta\mathbf{m}$ does not decrease during the inversion thereby leading to stronger inter-parameter mappings.

10.4 NUMERICAL EXPERIMENTS

We conduct a series of numerical experiments to interrogate specific components of the SEFWI algorithm. Initially, we test the efficiency gain offered by SEFWI when coupled with various optimization algorithms. In a subsequent test, we seek to verify the claim that parameter trade-off in SEFWI is comparable to FWI. Tests with noisy data and early termination are performed to test the sensitivity of the algorithm.

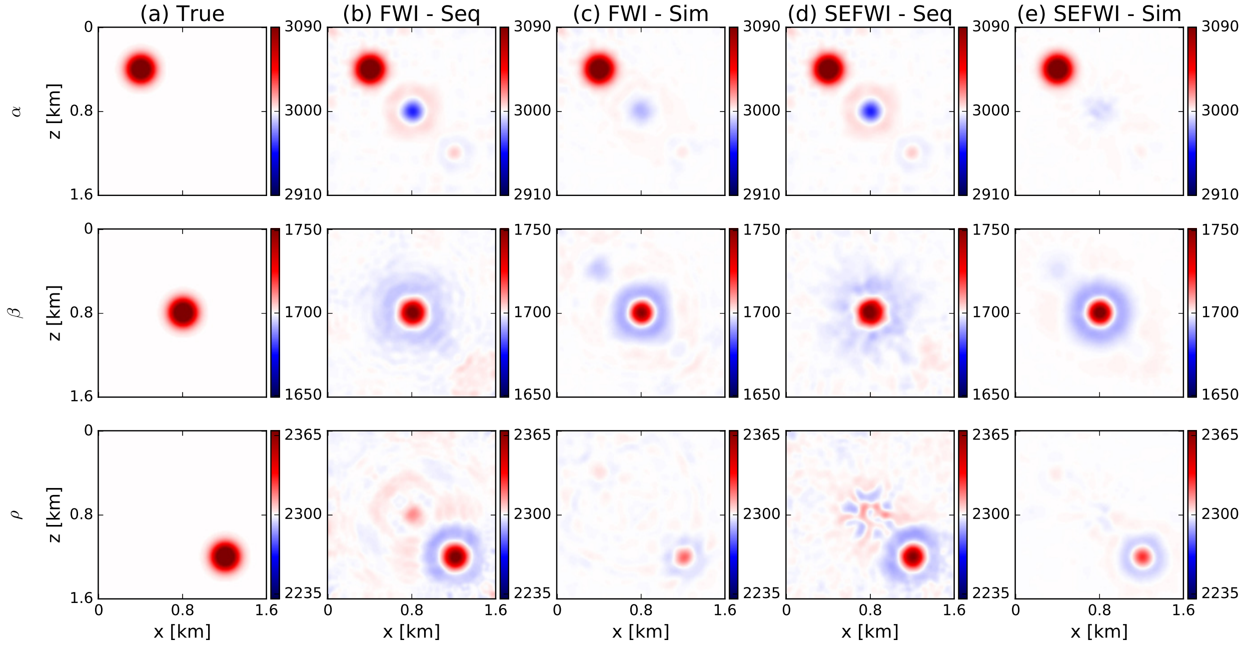


Figure 10.3: A comparison of (b, d) sequential and (c, e) simultaneous inversion of multiple parameters using FWI and SEFWI: (top row) α, (middle row) β, (bottom row) ρ. The impact of parameter trade-off can vary depending on the inversion strategy selected. In general, SEFWI exhibits very similar parameter trade-off to FWI for a particular strategy.

10.4.1 Inversion procedure

The inversion procedure described in this section is applicable to every experiment unless stated otherwise. 'Observed' and synthetic data are generated using 2D time-domain, P-SV finite difference modelling (fourth order in space, second order in time) (Virieux, 1986; Levander, 1988). Convolutional perfectly matched layers are implemented to simulate absorbing boundaries at the edges of the numerical grid (Komatitsch and Martin, 2007). The free surface is replaced with an absorbing boundary. Source inversion is not performed and the true source wavelet is available in all trials.

Elastic models are parametrized in terms of seismic velocities and density. Density is not included as an inversion parameter and is updated empirically via Gardener's relation ($\rho = 310\alpha^{0.25}$) where appropriate (Gardner et al., 1974). Nondimensionalization is applied to the inversion parameters via rescaling of the form $m_p' = m_p/m_0$ (Prieux et al., 2013a). The scaling values m_0 are taken as the mean values of the starting models. The gradient associated with the nondimensionalized parameters is $g_p' = m_0 g_p$. Inversion results are presented in terms of physical parameters. A square-root of depth preconditioner is implemented to compensate for inadequate illumination in deeper regions of the model. We avoid Hessian based preconditioners due to the differences between $\mathbf{H}$ and $\hat{\mathbf{H}}$.

Restart versions of NLCG and L-BFGS are restarted after every 3 and 5 iterations, respectively. The source encoding is altered after every restart. SD and NLCG employ a bracketing line search, whereas conventional and restart versions of L-BFGS use a backtracking line search. Used in conjunction with L-BFGS, the backtracking line search can provide step-lengths at almost no additional cost (Modrak and Tromp, 2016). The line search satisfies the Armijo condition (first Wolfe condition) (Nocedal and Wright, 2006). We do not require the curvature condition (second Wolfe condition) be satisfied as it requires additional gradient computations for each trial step. Source-encoded inversions are performed 5 times to account for the variability introduced by the random source encoding.

10.4.2 Diagnostic quantities

Before proceeding to the examples, we define diagnostic quantities that enable comparisons between SEFWI and FWI. The efficiency gain η is defined as $\eta = \widehat{NS}/NS$ and represents the ratio between the total number of simulations performed in SEFWI ($\widehat{NS}$) and FWI (NS). A simulation refers to any numerical solution of the forward or adjoint wave-equation during gradient computations or the line search. The convergence behaviour of each algorithm is included implicitly in η.

The relative model error is defined for each independent parameter as

$$m_p^{err} = \frac{\|m_p^* - m_p^k\|}{\|m_p^*\|}, \tag{10.37}$$

where m_p^* is the true model for parameter p. Repeated trials in SEFWI are used to compute the mean and local covariances of inverted models in SEFWI. The mean $\tilde{m}_p(\mathbf{x})$ is computed over N independent trials (Castellanos et al., 2015),

$$\tilde{m}_p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} m_p^{(i)}(\mathbf{x}). \tag{10.38}$$

The local covariance i.e. diagonals of the covariance matrix, are obtained via

$$\Sigma_{pq}(\mathbf{x},\mathbf{x}) = \sum_{i=1}^{N} \frac{(m_p^{(i)}(\mathbf{x}) - \tilde{m}_p(\mathbf{x}))(m_q^{(i)}(\mathbf{x}) - \tilde{m}_q(\mathbf{x}))}{N-1}. \tag{10.39}$$

In the absence of noise and regularization, the covariances act as a proxy for the cross-talk noise. This assumes that the mean inverted model is largely devoid of cross-talk artefacts.

10.4.3 Efficiency gain

Inversions are performed on a 20 km x 4.5 km, 2D section of the 3D acoustic SEG/EAGE overthrust model (Aminzadeh et al., 1997). Scaling relations are used to synthesize density ($\rho = 310\alpha^{0.25}$) and S-wave velocity ($\beta = \alpha/\sqrt{3}$) models. Starting models are obtained by convolving the true models with a Gaussian kernel ($\sigma_x = \sigma_z = 700$ m). True and initial α models are displayed in figure 10.4. The seismic experiment consists of 96 explosive sources ($\Delta x_s = 200$ m, $z_s = 25$ m) recorded at 264 multi-component receivers ($\Delta x_r = 75$ m, $z_r = 25$ m). All 96 sources are combined into a single encoded source in SEFWI. The source wavelet is a Ricker wavelet with a dominant frequency of 5 Hz, corresponding to a bandwidth of 0-15 Hz. Inversions are performed using SD, NLCG, and L-BFGS and terminate after 100 non-linear FWI/SEFWI iterations. We invert the full bandwidth data as multi-scale methods are not required in this case. Model regularization did not lead to a discernible improvement in the inverted models for this experiment and is therefore not included.

The convergence behaviour of FWI and SEFWI is summarized in table 10.1 and figure 10.5. For a particular optimization algorithm, SEFWI exhibits slower convergence, per iteration, in data misfit and model error when compared to FWI. The slower convergence is attributed to the presence

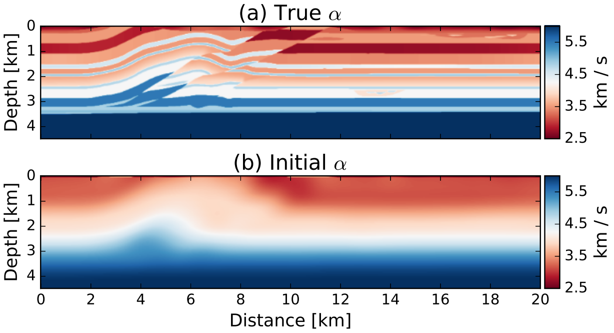


Figure 10.4: SEG/EAGE overthrust model: (a) true and (b) initial α models. Corresponding β and ρ models are synthesized from α using empirical scaling relations.

	Iterations		No. simulations		η	
Optimization	FWI	SEFWI	FWI	SEFWI	vs FWI	vs FWI (L-BFGS)
SD	59	98	44160	568	77.7	8.6
NLCG	33	48	16416	341	48.1	14.4
L-BFGS	24	76	4896	266	18.4	18.4

Table 10.1: SEG/EAGE overthrust inversion. A comparison of the computational resources required by FWI and SEFWI to achieve $\alpha^{err} = 0.65$. Efficiency gain (η) describes the ratio between the total number of simulations required by FWI and SEFWI. As an additional comparison, efficiency gain is computed relative to the most efficient FWI implementation (FWI w/ L-BFGS).

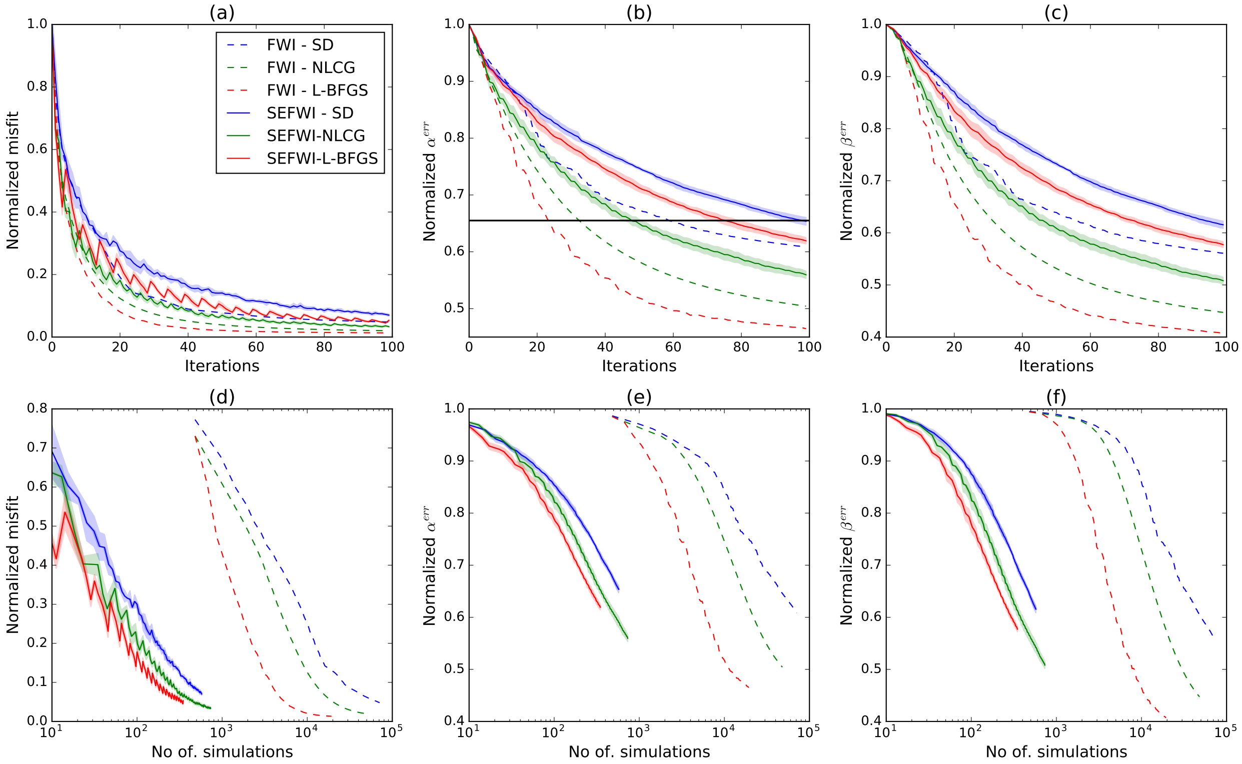


Figure 10.5: Comparison of convergence behaviour for (normalized) misfit, α and β model errors, as a function of (a-c) iterations and (d-f) total number of simulations. Dashed and solid coloured lines are used to display results for FWI and SEFWI, respectively. For SEFWI, lines correspond to mean values of misfit/model error from 5 random trials; the error bands represent one standard deviation. FWI exhibits higher per iteration convergence rates; however, the cost of an FWI iteration is greater than its SEFWI equivalent.

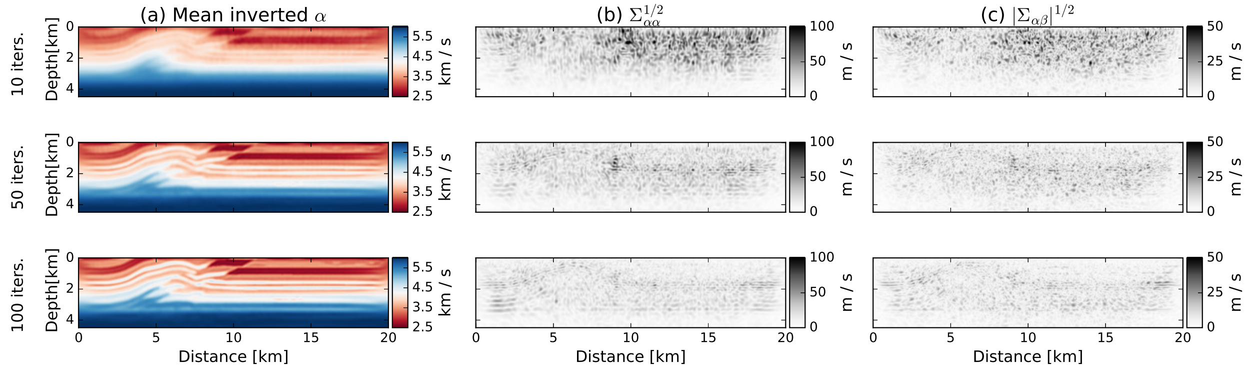


Figure 10.6: A comparison of the (a) mean inverted α model and (b-c) local covariances obtained after: (top row) 10, (middle row) 50, and (bottom row) 100 steepest descent iterations. The local standard deviation and covariances decrease in magnitude with an increasing number of iterations, implying that cross-talk artefacts are being attenuated.

of cross-talk noise in the gradient and is well established from prior studies (Moghaddam et al., 2013; Anagaw and Sacchi, 2014; Castellanos et al., 2015). Efficiency gain and further performance comparisons are evaluated at a target model error $\alpha^{err} = 0.65$. At the target model error, the largest efficiency gain is demonstrated by SD SEFWI ($\eta = 77.7$). NLCG and L-BFGS offer reduced efficiency gains of $\eta = 48.1$ and $\eta = 18.4$, respectively. The reduced effectiveness of NLCG/L-BFGS is due to an increased sensitivity of the algorithms to cross-talk noise. A similar observation was documented by Castellanos et al. (2015). Despite exhibiting reduced efficiency gains, NLCG and L-BFGS still outperform SD in SEFWI, requiring fewer iterations and simulations to reach the target model error (panel (b), figure 10.5). L-BFGS requires a greater number of iterations (76) than NLCG (48) to reach the desired model error; however, the more efficient line-search results in fewer overall simulations despite the disparity in iterations. Relative to our most efficient FWI implementation (L-BFGS + backtracking line search), the efficiency gain of SEFWI algorithms is more modest. This is an indication that the computational gain provided by SEFWI may be offset by the more sophisticated optimization algorithms available in FWI.

Figures 10.6 and 10.7 display mean inverted models and local covariances. As the inversions are devoid of noise and regularization, the local covariances may be interpreted as a proxy for the strength of the cross-talk artefacts. Figure 10.6 displays the evolution of $\tilde{\alpha}$, $\Sigma_{\alpha\alpha}^{1/2}$, and $|\Sigma_{\alpha\beta}|^{1/2}$ over the course of SD SEFWI iterations. The standard deviation of β ($\Sigma_{\beta\beta}^{1/2}$) is not included, but follows trends consistent with the other terms of the local covariance matrix. The local covariances reduce in magnitude at later iterations indicating that the cross-talk artefacts are being suppressed. Figure 10.7 displays a similar comparison, but compares $\tilde{\alpha}$, $\Sigma_{\alpha\alpha}^{1/2}$, and $|\Sigma_{\alpha\beta}|^{1/2}$ after 100 iterations

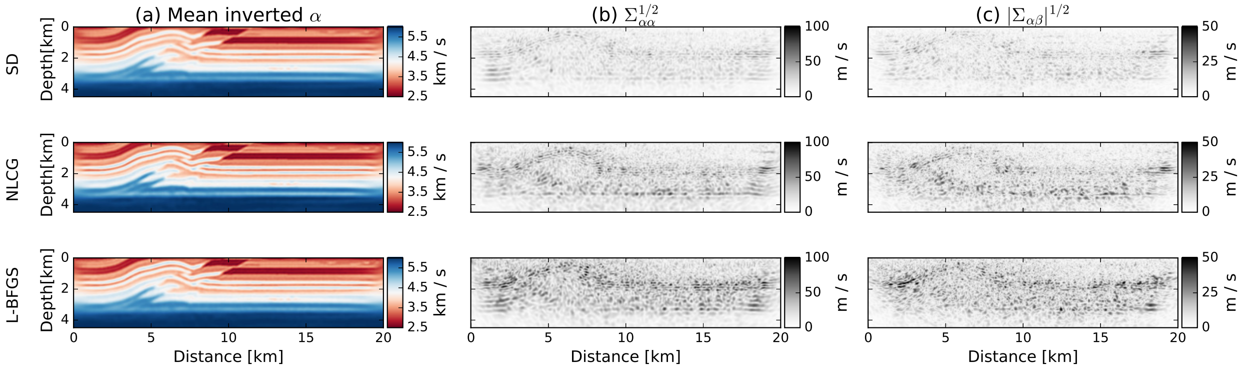


Figure 10.7: A comparison of the (a) mean inverted α model and (b-c) local covariances obtained after 100 iterations using: (top row) SD, (middle row) NLCG, and (bottom row) L-BFGS. SD randomizes the source-encoding at each iteration, whereas NLCG and L-BFGS randomize the source-encoding every 3 and 5 iterations, respectively. The strength of cross-talk artefacts is tied to the frequency with which the source-encoding is reset.

for the 3 different optimization methods. Figure 10.5 displays that NLCG and L-BFGS reach lower model errors than SD. While SD exhibits small local covariances, the models inverted using NLCG/L-BFGS are better resolved. The strength of the local covariances is directly related to the frequency with which the source-encoding is randomized within an algorithm. In these trials the restart intervals are 1, 3, and 5 for SD, NLCG, and L-BFGS, respectively. The restart version of NLCG appears to provide the best compromise between model resolution and mitigating cross-talk artefacts in the final model.

10.4.4 Parameter trade-off

The Marmousi II model is a fully elastic synthetic model with multiple hydrocarbon layers and complex faulting (Martin et al., 2006). Shallow shale layers in the original model exhibit low shear wave velocities (300-400 m/s) that require fine grid spacing to avoid dispersion related artefacts in the data. Reduced grid spacing increases the the computational cost forward/adjoint simulations due to a larger computational domain and considerations of numerical stability. To reduce the computational burden, S-wave velocities in the shale layers are replaced by $\beta = \alpha/\sqrt{3}$; density in these layers is rescaled via Gardener's relation. Adjusting the shale layers alone preserves heterogeneities exclusive to α or β. In this case, an exclusive heterogeneity refers to instances where one parameter demonstrates a significant perturbation from background, while the other parameter does not. Some examples are identified with white arrows in figure 10.8. Heterogeneities exclusive

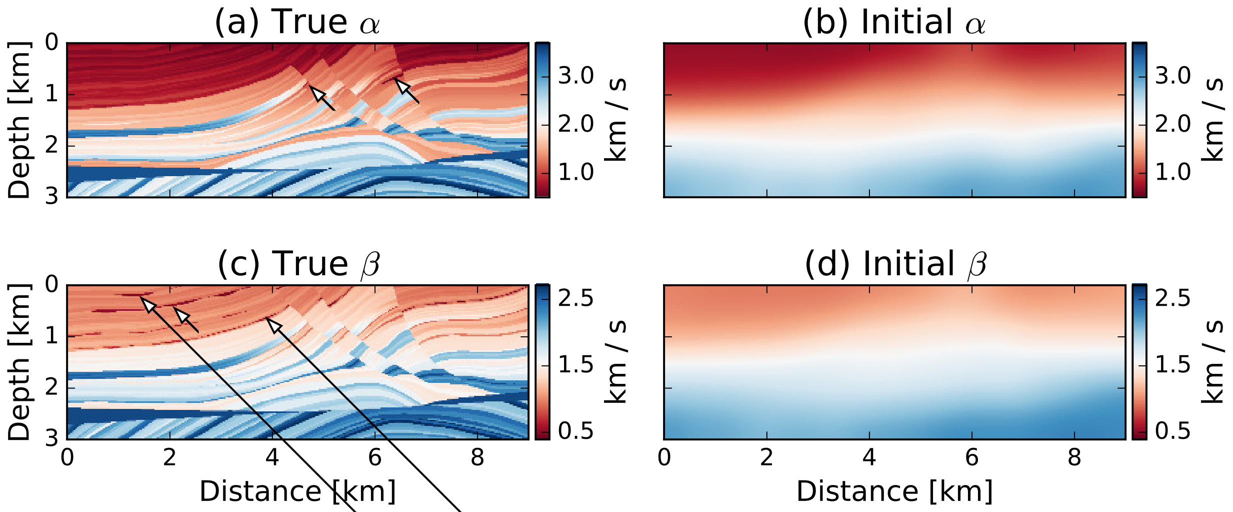


Figure 10.8: Modified Marmousi II model: (a, c) True and (b, d) initial α and β models. The original β model has been altered to increase β in the shale layers. A heterogeneous ρ model is used, but not displayed. Hydrocarbon reservoirs appear as distinct perturbations in α/β, some examples are indicated with white arrows.

to the α/β models serve as positional markers that are used to examine parameter trade-off. The water layer in the original model is removed to simulate land acquisition.

Initial models (fig. 10.8) are derived by convolving the true models with a Gaussian kernel ($\sigma_x = \sigma_z = 800$ m). The seismic experiment is composed of 112 explosive sources ($\Delta x_s = 80$ m, $z_s = 10$ m) and 296 receivers ($\Delta x_r = 30$ m, $z_r = 10$ m). 112 sources are reduced to $N_e = 2$ encoded sources, with each encoded source containing 56 individual sources. The source wavelet is a Ricker wavelet with a dominant frequency of 10 Hz. A 3 Hz highpass filter is applied to the data and source to remove some of the low frequency information. The starting model is sufficiently far from the true model that full-bandwidth FWI fails and converges to a local minima in the objective function. The multi-scale approach of Bunks et al. (1995) is implemented to circumvent cycle-skipping. The frequency bands used for inversion are informed by the selection criteria of Sirgue and Pratt (2004). Inversions are performed using low-pass cutoff frequencies of 3 Hz, 5 Hz and, 8 Hz. Multi-scale inversions are performed using only SD to allow for a more direct comparison. The source encoding is randomized at every iteration and inversions are terminated after 75 SD iterations at each scale. We include damping regularization in the form of eq. (10.27). For inversions on noise-free data $\gamma = 1 \times 10^{-4}$ and $\gamma = 1 \times 10^{-3}$ for the noisy data example.

The final inverted α and β models are displayed in figure 10.9. The SEFWI example corresponds to 1 of the 5 random trials conducted. Qualitatively, both inversion methods produce comparable results

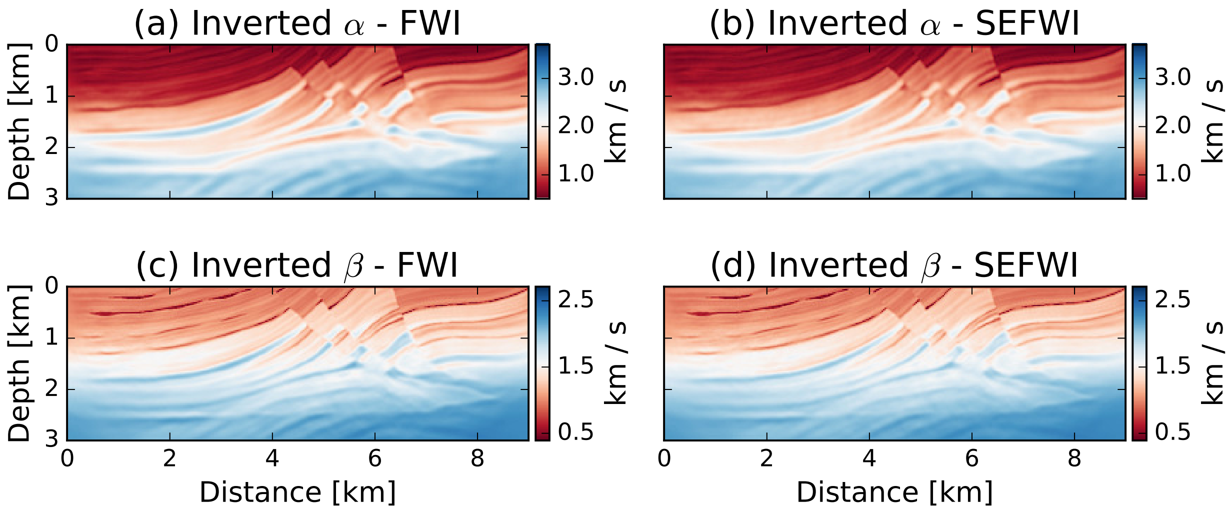


Figure 10.9: Inverted Marmousi II models: (a, c) Final α and (b, d) β models after multiscale inversion. SEFWI attains models with similar resolution to those from FWI. SEFWI models do not exhibit any discernible parameter trade-offs related to cross-talk

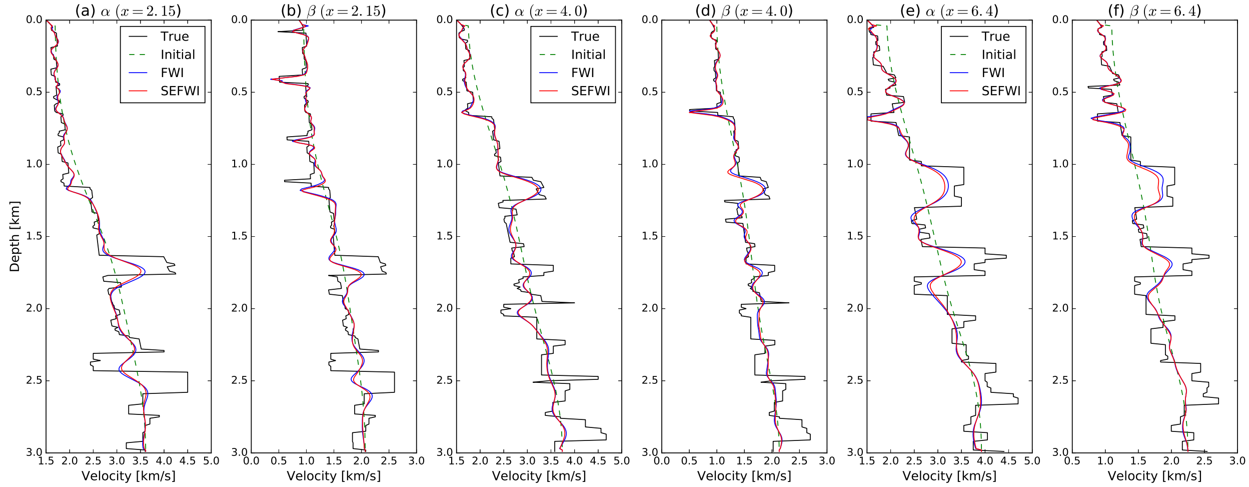


Figure 10.10: Pseudo well logs of α and β taken at (a-b) x = 2.5 km, (b-d) 4.0 km, and (e-f) 6.4 km. Models inverted with FWI display marginally better amplitude recovery at intermediate depths. Perturbations distinct to α or β do not appear to map into the other parameter, suggesting that the parameters are well resolved in both methods.

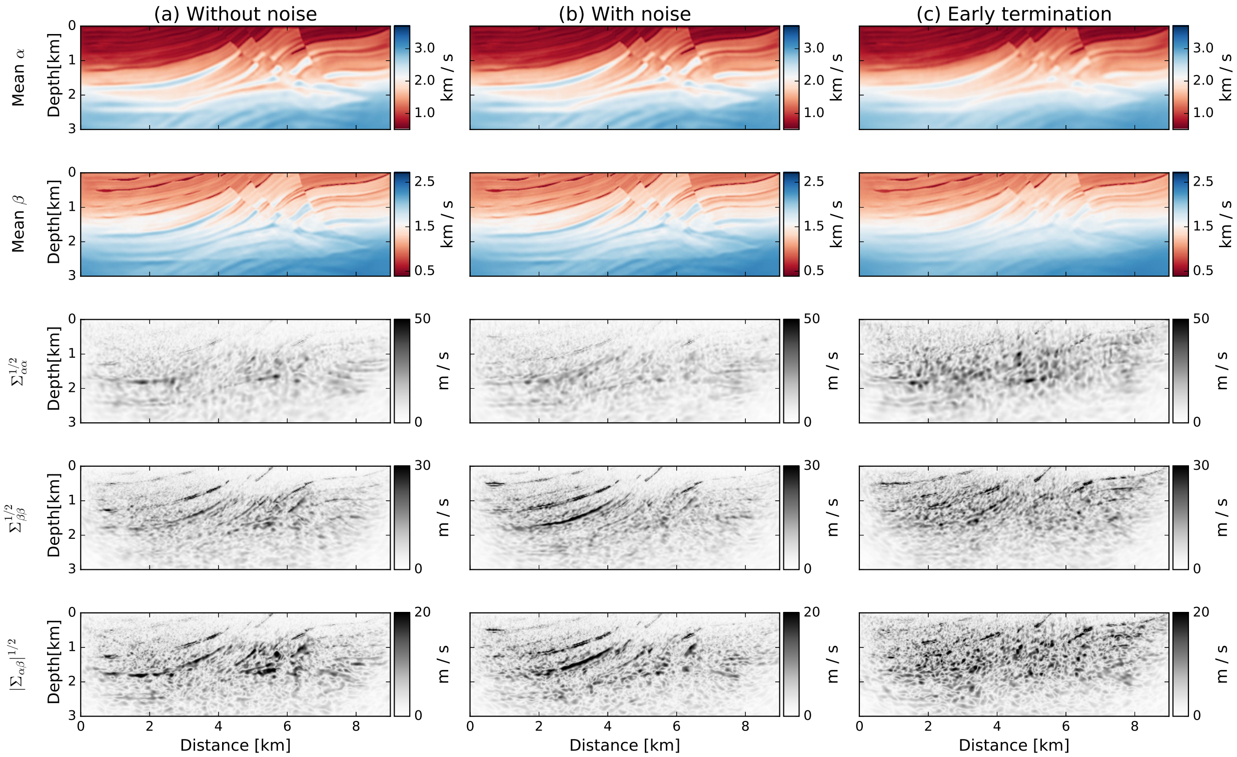


Figure 10.11: A comparison of mean α, mean β, and local covariances for SEFWI (a) with noiseless data, (b) noisy data (SNR=10 dB), and (c) early termination (noiseless data). (a) and (b) use 75 SD iterations whereas (c) undergoes early termination after 30 SD iterations.

and no cross-talk artefacts are noticeable in the SEFWI models. Hydrocarbon layers, indicated by arrows in figure 10.8, appear to be well resolved in α and β, with no perceptible trade-off between parameters. Further confirmation is provided by pseudo well-logs taken at different points in the model. For example, at $x = 2.5$ km the well intersects a hydrocarbon layer a distinct signature only in the β model. Conventional FWI provides a marginal improvement in the estimation of true model perturbations.

Sensitivity to random noise

To test the sensitivity of SEFWI to noise, we add random noise to the Marmousi II data. Gaussian white noise arrays are generated for each component of each shot record. The variance of the noise array is set by selecting a desired signal-to-noise ratio (SNR), defined as

$$\text{SNR (dB)} = 10\log_{10}\left(\frac{a_{rms}^2}{\sigma^2}\right), \tag{10.40}$$

and solving for the required variance σ^2. a_{rms}^2 is the root mean square amplitude of the shot record. For any given shot, the noise arrays of both components (x, z) have equal variance which results in different SNRs for the two components. We pick a_{rms}^2 from the z-component data and refer to the SNR of the z-component data in the text. The noisy data used for inversion has SNR = 10 dB. The inversion procedure follows the noise-free example with an increase to the regularization hyperparameter ($\gamma = 1 \times 10^{-3}$). This is done to damp high-frequency contributions from the noise to the model update.

Figure 10.11 (b) displays the mean inverted models and the local covariances. The inversion results are largely similar to the noise-free case. A small increase in the magnitudes of the local covariances is observed.

Sensitivity to early termination

Model updates that occur during the the later stages of FWI i.e. when the data misfit has largely flattened out, can still generate appreciable reductions in the model error despite producing only small reductions in the data misfit (e.g. figure 10.5, panels (a-c)). In realistic applications of 3D FWI, it may not be feasible to extend an inversion to a large number of iterations due to considerations of time or computational expense. In such cases, practitioners may terminate the inversion after a set number of iterations, before the optimization has truly converged. Early termination may also be prompted by strong noise in the data. In the presence of noise, the least-squares waveform misfit will converge to L_2 norm of the noise. Once the data misfit has flattened out, it is difficult to ascertain whether subsequent model updates are fitting the data or the noise. Early termination serves as a precautionary measure to prevent overfitting the data. In SEFWI, the later iterations are used to further reduce the imprint of cross-talk artefacts.

We perform a test to investigate the effect of early termination in SEFWI. The inversion procedure follows the earlier multi-scale inversion performed on noise-free Marmousi II data, but with the number of SD iterations reduced from 75 to 30 at each scale. The mean inverted models and the local covariances are depicted in figure 10.11 (c). Early termination does not appear to destabilize the inversion, rather it demonstrates two predictable results. Firstly, the mean inverted models are not as well resolved as equivalent inversions run to a greater number of iterations (figure 10.11 (a)). Secondly, early termination increases the magnitude of the local covariances, consistent with the expectation of increased cross-talk artefacts.

10.5 Conclusions

Source encoding has been applied to elastic isotropic full waveform inversion. The theory of source-encoded FWI was extended to the general multi-parameter case, with an emphasis placed on understanding the influence of source encoding on parameter trade-off. The behaviour was determined by analysing the source-encoded multi-parameter Hessian. The convergence of the expected source-encoded Hessian towards the conventional FWI Hessian, was verified via Hessian probing techniques. When cross-talk artefacts are suppressed, the properties of the source-encoded Hessian ensure that the parameter trade-off in SEFWI is comparable with FWI. Furthermore, SEFWI exhibits similar sensitivities to model parametrization and inversion schemes as FWI.

Numerical tests were conducted to evaluate the performance of SEFWI. Tests sought to assess: efficiency gain, parameter trade-off, and sensitivity to noise and early termination. For all optimization algorithms, SEFWI required significantly fewer overall simulations to reach a target model error. The efficiency gain was on the order of the number of individual sources in an experiment. We observed that a hybrid-CG algorithm outperformed SD or L-BFGS alternatives. While SD generated SEFWI models with the lowest variances, both CG and L-BFGS converged to a smaller model error in the same number of iterations. The increased model variance observed with hybrid CG/L-BFGS algorithms is due to less frequent randomization of the source encoding. A test on the Marmousi II model corroborated the claim that the parameter trade-off in SEFWI is comparable to that of FWI. Spatially inconsistent $P-$ and S-wave velocity models were well resolved in both methods. The presence of noise or early termination did not exacerbate the inversion results significantly. Early termination resulted in models that were less resolved and exhibited greater model variances.

Our results indicate that source encoding is feasible in multi-parameter FWI; however, there are concerns that hinder the use of source encoding in real data applications. The fixed-spread assumption impedes the use of source encoding in data-driven workflows that require extensive data pre-processing. Alternative misfit functions may be useful in such cases, as demonstrated by the application of SEFWI to streamer data (Routh et al., 2011; Choi and Alkhalifah, 2012). Ultimately, the applicability of source encoding on real data, multi-parameter FWI is entirely dependent on the dataset. Source encoding may also be a useful tool to access the multi-parameter Hessian in a more economical manner.

10.6 Acknowledgements

This research has been funded by the NSERC Alexander Graham Bell Canada graduate scholarship and the AITF scholarship. We would like to thank the sponsors of the SAIG group at the University of Alberta for their continued support. Inversions were automated using a modified version of the `SeisFlows` inversion framework.

References

Aki, K., and P. G. Richards, 2002, Quantitative seismology: University Science Books. Geology (University Science Books): Seismology.

Alkhalifah, T., and R. douard Plessix, 2014, A recipe for practical full-waveform inversion in anisotropic media: An analytical parameter resolution study: Geophysics, **79**, R91–R101.

Aminzadeh, F., J. Brac, and T. Kunz, 1997, SEG/EAGE 3-D salt and overthrust models: SEG/EAGE 3-D Modeling Series No. 1, SEG.

Anagaw, A. Y., and M. D. Sacchi, 2014, Comparison of multifrequency selection strategies for simultaneous-source full-waveform inversion: Geophysics, **79**, R165–R181.

Ben-Hadj-Ali, H., S. Operto, and J. Virieux, 2011, An efficient frequency-domain full waveform inversion method using simultaneous encoded sources: Geophysics, **76**, R109–R124.

Brossier, R., S. Operto, and J. Virieux, 2009, Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion: Geophysics, **74**, WC105–WC118.

Bunks, C., F. M. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: Geophysics, **60**, 1457–1473.

Capdeville, Y., Y. Gung, and B. Romanowicz, 2005, Towards global earth tomography using the spectral element method: a technique based on source stacking: Geophysical Journal International, **162**, 541–554.

Castellanos, C., L. Métivier, S. Operto, R. Brossier, and J. Virieux, 2015, Fast full waveform inversion with source encoding and second-order optimization methods: Geophysical Journal International, **200**, 718–742.

Chen, P., T. H. Jordan, and L. Zhao, 2007, Full three-dimensional tomography: a comparison between the scattering-integral and adjoint-wavefield methods: Geophysical Journal International, **170**, 175–181.

Choi, Y., and T. Alkhalifah, 2012, Application of multi-source waveform inversion to marine streamer data using the global correlation norm: Geophysical Prospecting, **60**, 748–758.

Dai, W., P. Fowler, and G. T. Schuster, 2012, Multi-source least-squares reverse time migration: Geophysical Prospecting, **60**, 681–695.

Epanomeritakis, I., V. Akelik, O. Ghattas, and J. Bielak, 2008, A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion: Inverse Problems, **24**, 034015.

Fichtner, A., H.-P. Bunge, and H. Igel, 2006, The adjoint method in seismology: Physics of the Earth and Planetary Interiors, **157**, 86 – 104.

Fichtner, A., B. L. N. Kennett, H. Igel, and H.-P. Bunge, 2009, Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods: Geophysical Journal International, **179**, 1703–1725.

Fichtner, A., and T. v. Leeuwen, 2015, Resolution analysis by random probing: Journal of Geophysical Research: Solid Earth, **120**, 5549–5573. (2015JB012106).

Fichtner, A., and J. Trampert, 2011a, Hessian kernels of seismic data functionals based upon adjoint techniques: Geophysical Journal International, **185**, 775–798.

———, 2011b, Resolution analysis in full waveform inversion: Geophysical Journal International, **187**, 1604–1624.

Gardner, G. H. F., L. W. Gardner, and A. R. Gregory, 1974, Formation velocity and density - the diagnostic basics for stratigraphic traps: Geophysics, **39**, 770–780.

Gholami, Y., R. Brossier, S. Operto, A. Ribodetti, and J. Virieux, 2013, Which parameterization is suitable for acoustic vertical transverse isotropic full waveform inversion? part 1: Sensitivity and trade-off analysis: GEOPHYSICS, **78**, R81–R105.

Haber, E., M. Chung, and F. Herrmann, 2012, An effective method for parameter estimation with pde constraints with multiple right-hand sides: SIAM Journal on Optimization, **22**, 739–757.

Huang, Y., and G. T. Schuster, 2012, Multisource least-squares migration of marine streamer and land data with frequency-division encoding: Geophysical Prospecting, **60**, 663–680.

Köhn, D., D. De Nil, A. Kurzmann, A. Przebindowska, and T. Bohlen, 2012, On the influence of model parametrization in elastic full waveform tomography: Geophysical Journal International, **191**, 325–345.

Komatitsch, D., and R. Martin, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: Geophysics, **72**, SM155–SM167.

Krebs, J. R., J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M.-D. Lacasse, 2009, Fast full-wavefield seismic inversion using encoded sources: Geophysics, **74**, WCC177–WCC188.

Lailly, P., 1983, The seismic inverse problem as a sequence of before stack migrations: J. B. Bednar,

R. Redner, E. Robinson, and A. Weglein, eds., Conference on inverse scattering: Theory and application: SIAM, 206–220.

Levander, A. R., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Martin, G. S., R. Wiley, and K. J. Marfurt, 2006, Marmousi2: An elastic upgrade for Marmousi: The Leading Edge, **25**, 156–166.

Métivier, L., F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux, 2014, Full waveform inversion and the truncated Newton method: quantitative imaging of complex subsurface structures: Geophysical Prospecting, **62**, 1353–1375.

Métivier, L., R. Brossier, J. Virieux, and S. Operto, 2013, Full Waveform Inversion and the Truncated Newton Method: SIAM Journal on Scientific Computing, **35**, B401–B437.

Modrak, R., and J. Tromp, 2016, Seismic waveform inversion best practices: regional, global and exploration test cases: Geophysical Journal International, **206**, 1864–1889.

Moghaddam, P. P., H. Keers, F. J. Herrmann, and W. A. Mulder, 2013, A new optimization approach for source-encoding full-waveform inversion: Geophysics, **78**, R125–R132.

Mora, P., 1987, Nonlinear twodimensional elastic inversion of multioffset seismic data: Geophysics, **52**, 1211–1228.

Nocedal, J., and S. J. Wright, 2006, Numerical optimization, 2nd ed.: Springer. Springer Series in Operations Research and Financial Engineering.

Operto, S., Y. Gholami, V. Prieux, A. Ribodetti, R. Brossier, L. Métivier, and J. Virieux, 2013, A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice: The Leading Edge, **32**, 1040–1054.

Pan, W., K. A. Innanen, G. F. Margrave, M. C. Fehler, X. Fang, and J. Li, 2016, Estimation of elastic constants for hti media using gauss-newton and full-newton multiparameter full-waveform inversion: Geophysics, **81**, R275–R291.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Plessix, R.-., and Q. Cao, 2011, A parametrization study for surface seismic full waveform inversion in an acoustic vertical transversely isotropic medium: Geophysical Journal International, **185**, 539–556.

Pratt, R. G., C. Shin, and G. J. Hick, 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Prieux, V., R. Brossier, S. Operto, and J. Virieux, 2013a, Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the Valhall field. Part 1: imaging compressional wave speed, density and attenuation: Geophysical Journal International, **194**, 1640–1664.

———, 2013b, Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the valhall field. part 2: imaging compressive-wave and shear-wave velocities: Geophysical Journal International, **194**, 1665–1681.

Romero, L. A., D. C. Ghiglia, C. C. Ober, and S. A. Morton, 2000, Phase encoding of shot records in prestack migration: Geophysics, **65**, 426–436.

Routh, P., J. Krebs, S. Lazaratos, A. Baumstein, S. Lee, Y. H. Cha, I. Chikichev, N. Downey, D. Hinkley, and J. Anderson, 2011, Encoded simultaneous source fullwavefield inversion for spectrally shaped marine streamer data: SEG Technical Program Expanded Abstracts 2011, 2433–2438.

Schuster, G. T., X. Wang, Y. Huang, W. Dai, and C. Boonyasiriwat, 2011, Theory of multisource crosstalk reduction by phase-encoded statics: Geophysical Journal International, **184**, 1289–1303.

Sears, T., S. Singh, and P. Barton, 2008, Elastic full waveform inversion of multi-component OBC seismic data: Geophysical Prospecting, **56**, 843–862.

Shin, C., S. Jang, and D.-J. Min, 2001, Improved amplitude preservation for prestack depth migration by inverse scattering theory: Geophysical Prospecting, **49**, 592–606.

Shipp, R. M., and S. C. Singh, 2002, Two-dimensional full wavefield inversion of wide-aperture marine seismic streamer data: Geophysical Journal International, **151**, 325–344.

Sirgue, L., and R. G. Pratt, 2004, Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies: Geophysics, **69**, 231–248.

Tang, Y., 2009, Target-oriented wave-equation least-squares migration/inversion with phase-encoded Hessian: Geophysics, **74**, WCA95–WCA107.

Tape, C., Q. Liu, A. Maggi, and J. Tromp, 2009, Adjoint Tomography of the Southern California Crust: Science, **325**, 988–992.

Tarantola, A., 1984, Linearized inversion of seismic reflection data: Geophysical Prospecting, **32**, 998–1015.

———, 1986, A strategy for nonlinear elastic inversion of seismic reflection data: Geophysics, **51**, 1893–1903.

Tromp, J., C. Tape, and Q. Liu, 2005, Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels: Geophys. J. Int., **160**, 195–216.

van Leeuwen, T., and F. J. Herrmann, 2013, Fast waveform inversion without source-encoding: Geophysical Prospecting, **61**, 10–19.

Vigh, D., and E. W. Starr, 2008, 3D prestack plane-wave, full-waveform inversion: Geophysics, **73**, VE135–VE144.

Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889–901.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1–WCC26.

Zhu, H., E. Bozdağ , and J. Tromp, 2015, Seismic structure of the European upper mantle based on adjoint tomography: Geophysical Journal International, **201**, 18–52.

Seismic.jl: status and examples

F. Carozzi[1]

Abstract

The Signal Analysis and Imaging Group (SAIG) released a package named Seismic.jl that provides tools to read, write and manipulate seismic data. The package is coded in julia, a new language that is being developed under the MIT license. For this reason, Seismic.jl has undergone several upgrades and modifications since its release in 2015.

Besides their constant evolution, julia and Seismic.jl are open-source codes and have proved to be useful tools for research and teaching.

This report presents a brief introduction to some outstanding functionalities of Seismic.jl. Through simple explanations and examples, it is intended as a help document for beginners as well as a leading point for discussion on future development.

11.1 Introduction

julia (Bezanson et al., 2017) is a modern dynamic programming language specially designed to fulfill the needs of high-performance data science and numerical analysis applications. It was originally released in 2012 under the MIT license, and it is still a developing language.

Among its numerous functionalities, the most outstanding ones include multiple dispatch function definitions, a dynamic type system, parallelism, cloud computing functionalities, and simple adaptability to C and Fortran functions.

[1] Email: carozzi@ualberta.ca

Since 2015, SAIG has been developing the `Seismic.jl` package (Stanton and Sacchi, 2016), written with `julia` language. Seismic.jl is an open source software consisting of more than ten modules that allow conventional seismic data manipulation. Its most unique characteristic resides in its approach towards multidimensional arrays and its capability to handle and process this type of data. The full package is periodically maintained and updated to the last stable version available for `julia`. Its latest version is always distributed through `https://github.com/SeismicJulia/Seismic.jl`.

`Seismic.jl` can be downloaded and installed on the `julia` command line by typing `Pkg.add("Seismic")`. As not always the last version is the one registered as a `julia` package, it is highly recommended that, after installation, the user performs `Pkg.checkout("Seismic")` which updates the installation to the master branch of the package. Any developer interested in contributing to the package is welcomed to do so. To that end, please follow the instructions on the tutorial `https://seismic.physics.ualberta.ca/docs/develop_SeismicJulia.pdf`.

The main features of `Seismic.jl` can be summarized as follows

- Reading & Writing
- Modelling
- Processing tools
- 5D binning and data manipulation
- Imaging
- Tools

being “Tools” a combination of basic functions, wavelets, and windows used to support the needs of other functions from the package. The aim of this report is to present working examples that lead to a friendly approach towards the basic usage of `Seismic.jl` and its dependencies.

11.2 Reading and Writing - SEIS Format

`Seismic.jl` presents a particular format where headers and data are kept in separate files. This characteristic allows a dynamic manipulation of the seismic data based only on headers values.

The package is capable of reading either SEGY or SU format and convert the file to the internal `Seismic.jl` format, `SEIS`. After such process data remains stored in three independent files.

The initial file, where the actual traces are stored, has the extension `@data@`. The second file holds the headers information with the extension `@headers@`. Both files are stored in binary format. Finally, a third file without extension is created. This file has text format and holds the `extent` information which is most important for the correct internal process of reading and writing seismic data. That is, `Seismic.jl` can work without the headers of the data but many of its core functions will not work properly if the extent information is missing. Furthermore, the extent file saves the location of the files containing the corresponding data and headers.

In addition, two custom types, `Header` and `Extent`, are defined allowing powerful imaging and processing workflows for 5D data. The Header type includes the following 31 fields

- `tracenum`: Trace number
- `o1`: Initial value for the first dimension (time/depth)
- `n1`: Number of samples in the first dimension
- `d1`: Sample interval for the first dimension
- `sx`, `sy`, `gx`, `gy`: source and receiver coordinates
- `mx`, `my`, `hx`, `hy`, `h`, `az`, `ang`: midpoint coordinates, offset, azimuth and angle
- `isx`, `isy`, `igx`, `igy`: source and receiver calculated grid nodes after binning
- `imx`, `imy`, `ihx`, `ihy`, `ih`, `iaz`, `iang`: midpoint, offset, azimuth and angle grid nodes after binning
- `selev`, `gelev`: receiver group and source elevation
- `sstat`, `gstat`: source and group static correction
- `trid`: Trace identification code, based on SU indexing.

Moreover, the Extent type includes

- `n1,n2,n3,n4,n5`: Number of samples in each dimension
- `o1,o2,o3,o4,o5`: Initial value for each dimension

- `d1`,`d2`,`d3`,`d4`,`d5`: Sample interval in each dimension
- 5 positions for labeling the information contained in the dimensions considered above
- 5 positions for indicating the units in which the information is provided.

Both composite types and the types corresponding to each field inside them are defined in `Seismic.jl/src/ReadWrite/Header.jl`.

The functionalities for reading, writing and format conversion included in `Seismic.jl` are `SegyToSeis`, `SeisRead`, `SeisReadHeaders`, `SeisHeaderInfo`, `SeisWrite`, `SeisWriteHeaders`, `SeisCopy`, `SeisRemove`, and `SeisToSegy`. A brief description of the capabilities of such functions is included below.

11.2.1 `SegyToSeis`

Converts SEGY or SU data to `SEIS` format. This function considers the endianness definition of the file through the keyword argument "swap_bytes". The endian-format of a file is defined by the order in which the bytes are saved. Big-endian format stores the byte with the most significant bit in the first place. In contrast, little-endian format stores it in the final position. In addition, it considers the floating point definition of the file (either "IBM" or "IEEE") (Meisinger, 2004).

Both considerations, endianness and format of a file can have a big impact in a proper or improper reading of the data. These are characteristics that should be considered before reading any file.

11.2.2 `SeisRead` and `SeisReadHeaders`

Once the data is converted to the `SEIS` format, if we need it to be stored in the memory we should read it using `SeisRead`. `SeisRead` offers two possibilities, either reading all the information or only traces and extent. In order to use the resources of the computer optimally, these two options should be considered before reading information.

Given the presence of the extent file, the data read by `SeisRead` will be shaped according to its proper dimension.

Furthermore, the option of reading only headers is also available through `SeisReadHeaders`. This is a faster, computationally less expensive alternative, that allows the user to calculate statistics, visualize geometry, and manipulate data among others.

11.2.3 SeisHeaderInfo

Without storing the information in the computer memory, Seismic.jl offers a tool to compute header statistics. This is done via SeisHeaderInfo. The output is a comparison of minimum, maximum and mean values for each of the field composing the Header type.

11.2.4 SeisWrite, SeisWriteHeaders, SeisCopy, SeisRemove and SeisToSegy

The remaining functionalities allow the user to write the calculated data, or just headers, to disk. This can also be done through SeisToSegy if SU or SEGY files are needed. Finally, if a SEIS file is to be removed, it is recommended that it is done using SeisRemove.

11.2.5 Reading & writing example

Several small files in SU format are loaded to http://seismic.physics.ualberta.ca/data for reading and format conversion testing. For any of these files, the steps needed to convert the data and read it in Seismic.jl are

```
1. using Seismic, PyPlot

2. download("http://seismic.physics.ualberta.ca/data/gom_cdp_nmo.su","gom_cdp_nmo.su");

3. SegyToSeis("gom_cdp_nmo.su","gom_cdp_nmo",format="su",input_type="ieee",swap_bytes=true)

4. d,h,ext=SeisRead("gom_cdp_nmo");

5. SeisPlot(d)
```

The first line loads two packages: Seismic and PyPlot. PyPlot is required by Seismic for plotting purposes.

The second line uses a julia function to download the data from a web link to our computer. It will be saved as gom_cdp_nmo.su in the current folder.

Following, SegyToSeis performs the conversion from SU to SEIS format. At this point, three files are generated with the names gom_cdp_nmo, gom_cdp_nmo@data@ and gom_cdp_nmo@headers@. The

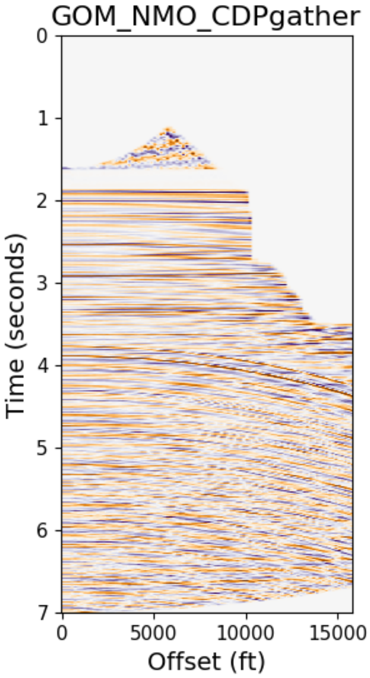


Figure 11.1: A read and plot example.

first one is the extent file, the second one includes the proper information in the traces and the third one incorporates the header information. Even though these are three different files, `Seismic.jl` will allow the user to refer to the information as `gom_cdp_nmo` without having to consider the partitioning of the data.

Next, the data is loaded into memory. Three variables are defined in order to save such information. `d` is an array of type `Float32` with the number of dimensions according to the extent information. `h` is an array of type `Seismic.Header`, the custom type described in the introduction to this section. `ext` is a constant of type `Seismic.Extent`, also defined in this section of the report. In order to check these types, the user can write `typeof(d)` in the `julia` prompt.

Finally, the data in `d` is ploted using `SeisPlot`, a `Seismic.jl` function for plotting needs (Figure 11.1).

11.3 Core packages

Once the data is available in the `SEIS` format, the user can work with the core modules of the package. A list of the functions in each module is available in Table 11.1. In addition, a collection of tests and examples are available in the following webpage
`https://github.com/SeismicJulia/Seismic.jl/tree/master/examples` .

Documentation and help on usage of each of the functions are available by writing in the `julia` prompt a question mark, followed by the name of the function. For example

```
julia > ?
help?  > SeisLinearEvents
```

will prompt the particular information to model seismic linear events. In addition, the webpage `http://SeismicJulia.github.io/Seismic.jl/` provides extra documentation on the package and modules.

Most `Seismic.jl` functions have two types of arguments, regular arguments and **keyword** arguments. Keyword arguments in `julia` are function arguments that are identified by name instead of position. This feature of `julia`, and by inheritance `Seismic.jl`, ease the use of complex interfaces.

Keyword arguments have default values, that is they need to be defined only when necessary. For example, continuing with `SeisLinearEvents`, the function can be called as

```
julia > d,ext = SeisLinearEvents();
```

which will create seismic linear events with default parameters.

In the case the user is interested in defining specific modelling parameters, the keyword arguments can be passed in the function call as

```
julia > d,ext = SeisLinearEvents(dt = 0.002, nt = 500, ox1 = 0, ox2 = 0, dx1 = 10,dx2 = 10,
dx3 = 10, dx4 = 10, nx1 = 20, nx2 = 20, nx3 = 20, nx4 = 20, tau = [1./4.,1./3.,0.7],
p2 = [−.001,0.002,0.0015], p1 = [0.0,0,0.0], p3 = [0.0,0,0.0], p4 = [0.0,0,0.00], amp = [1.0,−0.7,0.5],
f0 = 20.0 );
```

which creates a tailored volume that meets the specific needs. In the previous case, there are 4 dipping events. To plot the volume

```
julia > SeisPlot(d[:,10,:,10,10], style="wiggles", dy=0.002, xlabel="Trace number",
ylabel="Time (s)")
```

which plots the traces for a $2D$ slice of the previously created volume with proper axis; see Figure 11.2.

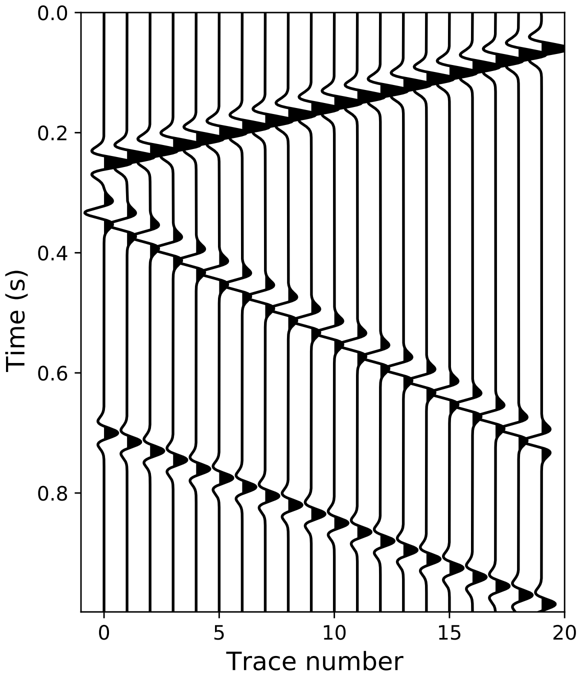


Figure 11.2: Seismic Traces created with function SeisLinearEvents.

Regular and keyword arguments with their default assignment are described in the `julia` prompt help and the documentation webpage.

11.4 Utils module - 5D data

A particular emphasis is needed on the `Utils` module, the section that deals with most of the functions that handle and process multilinear arrays. The functions that integrate such module are listed on Table 11.1, and the corresponding documentation and help are available as described in the previous section.

The capabilities of this module will be explained through a simple example using seismic prestack data. The script to the example consists of the following main commands

```
1. Download

2. SegyToSeis(‘‘file_su’’, "file_seis", format="su", input_type="ieee")

3. SeisGeometry(‘‘file_seis’’; param1...)

4. SeisBinHeaders(‘‘file_seis’’,’’file_bin’’; param2...);

5. SeisBinData(‘‘file_seis’’,’’file_bin’’; param2...);

6. patch_out,npatch=SeisPatch(‘‘file_bin’’,"patch"; param3...  );

7. SeisUnPatch(‘‘patch_out’’,’’file_final’’; nt=1000,param3...);
```

Following are the explanations to each line.

1. `SegyToSeis` is the function that converts formats from SU, SEGY or Madagascar to `SEIS`. In this case, the input file is `file_su` and the output file will be named `file_seis`. As explained in the Reading and Writing section of this report, `file_seis` is actually composed of three files, two with binary data and a text file containing the extent information. In this example, the input data is a SU file with IEEE format.

2. `SeisGeometry` updates the headers with geometry information. `param1` is a dictionary containing the parameters needed for the case. In the example provided in the link, the parameters considered are the stepsize for common midpoint coordinates, offset and azimuth.

3. SeisBinHeaders and SeisBinData are the functions calculating the new binned volume. The functions are independent in order to be able to quality control the parameters for the binning using the header information, prior to the calculation of the complete volume.

   The parameters needed for these steps should be common. In the provided example, they are the minimum and maximum node value for the binned volume plus the stepsizes used for SeisGeometry. In every case, the binning style should be defined between a provided selection.

   The termination of SeisBinHeaders will provide two files file_bin with the extent information and file_bin@headers@ with the headers.

4. SeisBinData effectively calculates the traces of the binned volume and saves them to the file file_bin@data@

5. SeisPatch generates the patches from the original volume. The outputs of this function are patch_out, a string array containing the name of the files where each patch is saved, and npatch a constant containing the number of generated patches. This constant can be used for different processes applied to each individual section of data.

   The parameters needed for SeisPatch include style, minimum and maximum grid nodes to consider, length of each dimension for the patches and overlapping traces.

6. SeisUnPatch reclusters the patched information into a unique volume. The parameters are common to those used for SeisPatch, including the number of time samples of the final volume.

A working example with a subsection of the data used in this report can be found in the following links https://tinyurl.com/ybwmn38c , and https://tinyurl.com/ycr37c6c . In addition, the function SeisPatchProcess performs a complete process with big volumes, that is, it patches the initial data, applies functions to the individual sections, and finally unpatches the seismic volume from the partial resulting sections.

11.5 Conclusions

Seismic.jl is the young, open sourced package developed by SAIG that provides tools to read, write and process seismic data. Coded in the new julia language, it fully benefiting from its capabilities specially designed for high-performance data management.

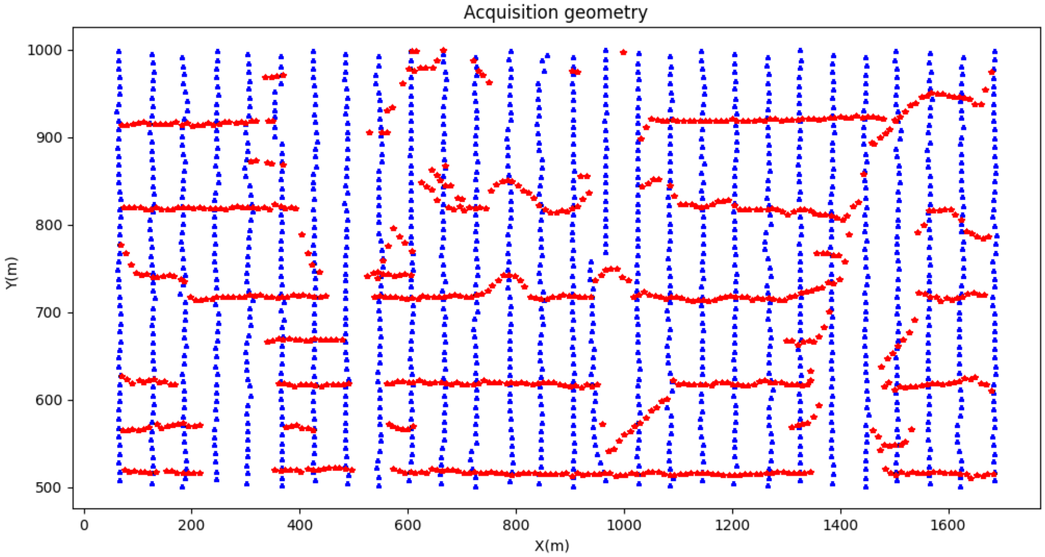


Figure 11.3: Data acquisition geometry.

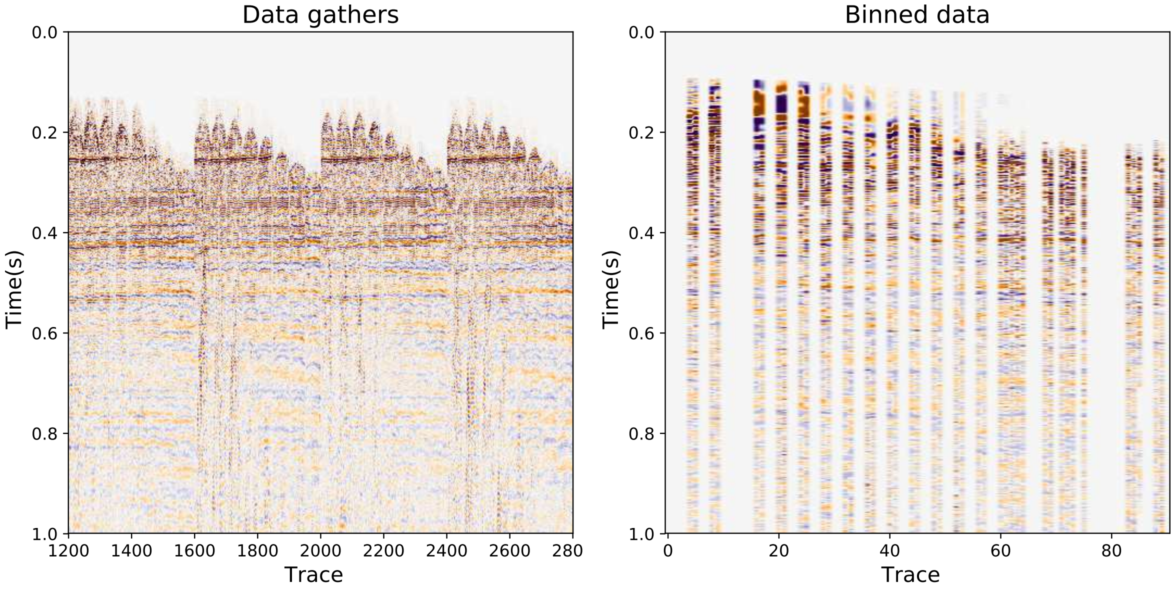


Figure 11.4: Original data gathers and one slice after binnig.

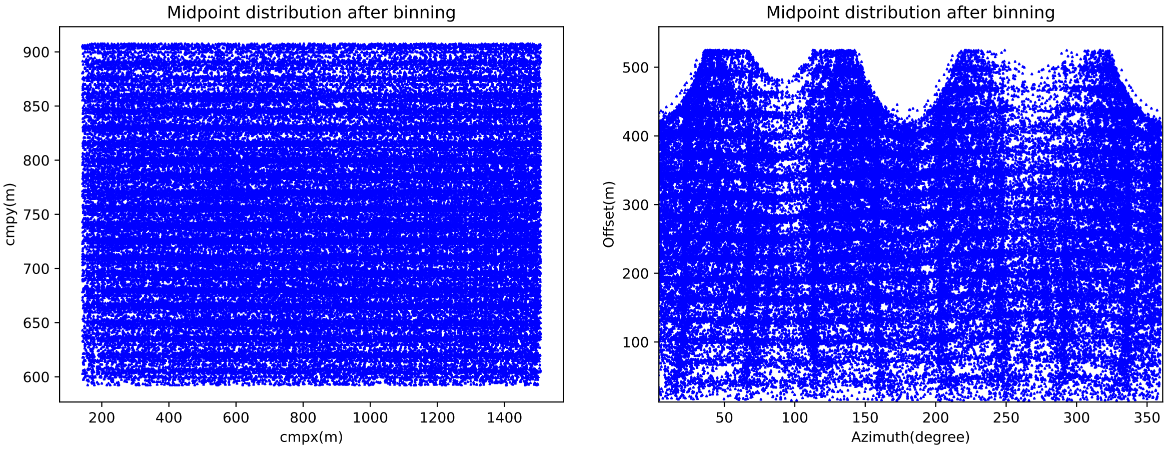


Figure 11.5: CMP and offset distribution after binning.

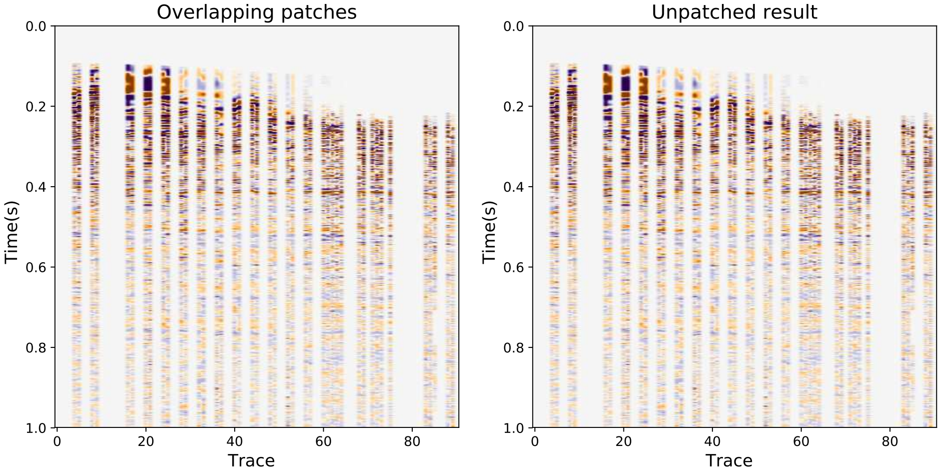


Figure 11.6: Overlapping patches and reclustered or unpatched data. This figure together with Figure 11.4 show that the patching and unpatching processes are not modifying the original data.

This report introduces the `Seismic.jl` most outstanding features as its efficiency in handling header information and multidimensional data. In addition, it provides a brief description of core modules and their documentation. Finally, an example processing 5D data is provided.

To conclude, this report is intended to be a document for inexperienced beginners as well as a benchmark for discussion on future developments.

11.6 Acknowledgements

We thank the creators of `julia`, `Seismic.jl`, and the sponsors of Signal Analysis and Imaging Group (SAIG) at the University of Alberta.

References

Bezanson, J., A. Edelman, S. Karpinski, and V. B. Shah, 2017, Julia, a fresh approach to numerical computing: SIAM Review, **59**, 65–98.

Meisinger, D., 2004, Segy floating point confusion: CSEG Recorder, **29**.

Stanton, A., and M. D. Sacchi, 2016, Efficient geophysical research in julia: CSEG GeoConvention, 1–3.

Modelling & Wavelets & Windows	Imaging	Processing	Utils	Tools, Solvers & Operations
SeisAcousticWave	PostStackWEM	SeisBandPass	SeisBinData	CalculateSampling
SeisAddNoise	SeisPSTM	SeisDecimate	SeisBinHeaders	Convmtx
SeisHypEvents	ShotProfileEWEM	SeisDelay	SeisGeometry	DotTest
SeisLinearEvents	ShotProfileLSEWEM	SeisDiff	SeisPatch	InnerProduct
SeisParabEvents	ShotProfileLSWEM	SeisEnvelope	SeisProcess	Integrate
Berlage	ShotProfileWEM	SeisFKFilter	SeisProcessHeaders	MeasureSNR
Ormsby	WaveSep	SeisFXDecon	SeisSort	Pad5D
Ricker	ComputeAngles	SeisGain	SeisUnPatch	PadFirstAxis
Hamming	GetShots	SeisKolmogoroff	SeisWindow	PhaseShift
		SeisMWNI	SeisWindowHeaders	TriangleFilter
		SeisMute		CGStep
		SeisNMO		ConjugateGradients
		SeisPOCS		IRLS
		SeisPWD		FFTOp
		SeisRadonFreqFor		LinearOperator
		SeisRadonFreqInv		MatrixMultiplyOp
		SeisSemblance		WeightingOp
		SeisSmooth1		
		SeisSmooth2		
		SeisSmoothGathers		
		SeisSmoothStructure		
		SeisStack		

Table 11.1: Core modules and functions available in Seismic.jl.