# Studies in multicomponent seismic data processing and Kronecker least-squares reverse time migration

by

Wenlei Gao

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

Doctor of Philosophy
in
Geophysics

Department of Physics
University of Alberta

# Abstract

Traditionally, seismic data processing considers data composed mainly of compressional waves. In recent years, we have witnessed the rapid development of multicomponent seismic exploration in the oil and gas industry. Multicomponent seismic data contain both compressional (P) and shear (S) wave modes permitting improving processes associated with lithology identification, fluid discrimination and fracture-stress characterization. Unfortunately, deficiencies of land multicomponent seismic data, such as the low-quality of horizontal components, inadequate spatial sampling of the elastic wavefield and misalignment of time-domain PP-wave and PS-wave images, pose unique challenges for conventional seismic processing and interpretation workflows. This thesis addresses several essential processing steps for multicomponent seismic data. Multilinear (tensor) algebra is proposed as a means to denoise and regularize onshore multicomponent data. For this purpose, I adopted the Candecomp/Parafac (CP) tensor decomposition to represent seismic prestack volumes. The decomposition is used for denoising and 5D tensor reconstruction. Randomization techniques are utilized to reduce the computational cost of the CP decomposition and making it an efficient noise attenuation tool for large multi-dimensional seismic datasets. The process of aligning PS-wave events to their corresponding PP-wave responses is called registration which is also studied in this thesis. Registration is posed as a non-linear constrained optimization problem. I also proposed a processing flow where 5D reconstruction via low-rank tensor completion is applied to prestack PP-wave and PS-wave data independently to enhance the data quality before registration.

In the second part of my thesis, I proposed a new method for least-squares reverse time

migration. Inspired by techniques associated to matrix and tensor completion, I approximate the Hessian matrix of the least-squares reverse time migration problem as the superposition of Kronecker products. The latter leads to an efficient least-squares reverse time migration algorithm that operates in the image domain.

# Preface

A version of the work in chapter 3 of this thesis has been published in a journal paper: Gao. W., and M. D. Sacchi, 2019, Random noise attenuation via the randomized CP decomposition, Geophysical Prospecting, accepted.

A version of the work in chapter 5 of this thesis has been published in a journal paper: Gao. W., and M. D. Sacchi, 2017, Multicomponent seismic data registration by nonlinear optimization, Geophysics, 83, V1-V10.

A version of the work in chapter 6 of this thesis has been published in a journal paper: Gao. W., Gian. M. and M. D. Sacchi, 2019, Fast least-squares reverse-time migration via a superposition of Kronecker products, Geophysics, accepted.

In these publications, I was responsible for designing and programming the algorithms, preparing data examples and writing the manuscripts. Dr. Sacchi was the supervisory author and was involved in concept formulation and manuscript editing.

# Dedication

*To my wife, Yue Wang*

# Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Mauricio Sacchi, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been fortunate to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I want to thank everyone in the Signal Analysis and Imaging Group (SAIG) for those enjoyable conversations and the fruitful discussions. Furthermore, I appreciate the sponsors of SAIG for their financial support and constructive feedback from the annual meetings. I also thank TGS for providing field seismic data for testing algorithms proposed in this research. Also, I am grateful to Prof. Guofa Li, Jianjun Gao, and Dr. Mengyao Sun for their guidance and fruitful discussion during my Ph.D. study. I appreciate my friends Zhenhua Li, Siwen Wei, Jiang Li, Ruijia Wang and Yuefeng Chen for their company during my stay at the University of Alberta. I want to express my special thanks to my wife Yue Wang, for her continued support and encouragement. I sincerely thank my parents, Ping Gao and Mei Zhao, for their unconditional trust and endless patience.

# Contents

# List of Tables

# List of Figures

# List of abbreviations and symbols

| | |
|---|---|
| $f$-$\mathbf{x}$ | Frequency-space domain |
| $fold_n$ | mode-$n$ tensor folding |
| $t$-$\mathbf{x}$ | Time-space domain |
| $unfold_n$ | mode-$n$ tensor unfolding |
| $V_p/V_s$ | $V_p/V_s$ ratio |
| ALS | Alternative least-squares method |
| AVO | Amplitude variation with offset |
| CG | Conjugate gradient method |
| CIGs | Common image gathers |
| CP | Canonical polyadic decomposition |
| HO-SVD | High order singular value decomposition |
| K-SVD | K-means singular value decomposition |
| LSM | least-squares migration |
| LSRTM | least-squares reverse-time migration |
| MSSA | Multi-channel singular value analysis |
| OBS | Ocean bottom survey |
| PMF | Parallel matrix factorization |
| POCS | Projection onto convex sets |
| RTM | Reverse-time migration |
| SVD | Singular value decomposition |

| | |
|---|---|
| $\circ$ | Vector outer product |
| $\circledast$ | Hadamard product |
| $\Delta t$ | Time sampling interval |
| $\gamma(t,x)$ | $V_p/V_s$ velocity ratio |
| $\iint$ | Integration |
| $\mathbb{R}$ | Real data set |
| $\boldsymbol{\mathcal{D}}$ | Tensor |
| $\mathbf{D}$ | Matrix |
| $\mathbf{d}$ | Vector |
| $\mathbf{H}$ | Hessian matrix |
| $\mathbf{L}$ | Born approximation operator |
| $\mu$ | Trade-off parameter |
| $\odot$ | Khatru-Rao product |
| $\otimes$ | Kronecker product |
| $\times_i$ | Mode-$i$ tensor multiplication |
| $^\dagger$ | Moore-Penrose pseudo-inverse |
| $d$ | Scalar |
| $D[i_1,\ldots,i_N]$ | One element of tensor $\boldsymbol{\mathcal{D}}$ |
| $e_P$ | Envelope of PP-wave data |
| $e_S$ | Envelope of PS-wave data |
| $P_\Omega$ | Projection operator |
| $w(t,x)$ | Warping function |
| $\mathbf{D}_{(n)}$ | Mode-$n$ unfolding of tensor $\boldsymbol{\mathcal{D}}$ |

# CHAPTER 1

# Introduction

Seismic surveys play a critical role in oil and gas exploration and development. The general principle of the seismic exploration method is to send waves, usually generated by a surface source such as a dynamite explosion or a mechanical device, into the Earth. The layer interfaces of the Earth reflect a portion of the down-going wave energy to the surface. These reflected waves are recorded by receivers, which can detect the ground motion and convert this motion into an electrical signal. After a series of seismic data processing steps, we can produce detailed images of the subsurface by mapping the surface-acquired reflected wave energy to the location of interfaces. These images assist in determining the location and size of oil and gas reservoirs.

The real Earth is too complicated to be precisely described by any mathematical model. Traditionally, an acoustic Earth model is an approximation that assumes the Earth behaves as a fluid and only supports the propagation of compressional (P) waves. This approximation simplifies the seismic data processing workflow significantly (Yilmaz, 2001).

Exploration seismology adopts the physics of wave propagation and statistical signal processing methods to estimate images of the subsurface. The main steps in exploration seismology involve data acquisition, data preconditioning, imaging, inversion, and interpretation. My work focuses on preconditioning multicomponent seismic data with a particular interest in defining new methods, and data processing flow to improve the quality of seismic images acquired via PP and PS seismic reflections. Specific challenges in exploration seismology are those associated with multi-channel seismic data denoising, reconstruction, and data registration. A goal of my research is to obtain high-quality subsurface images of PP and PS reflectivity estimates that can facilitate subsurface parameter estimation and geological interpretation. As part of my research, I also delved into seismic imaging by adopting tools of multilinear algebra that I have developed to process converted-wave data.

Exploration seismology centers on the utilization of compressional waves (P-waves). However, recent acquisition multicomponent seismic recording systems can also record shear wave modes (S-wave). Techniques developed for subsurface imaging via P waves have dominated exploration seismology. For the last 50 years, efforts in signal processing have focused on adopting scalar wavefield seismology methods to denoise and enhance the quality of seismic records before imaging (Yilmaz, 2001). A challenge today is to formulate new techniques, and processing flows that are suitable for vector field seismic data processing and imaging. In other words, we want to fully exploit the richness of vector field measurements (P and S-waves) to improve the detection and characterization of reservoirs of gas and oil (Stewart et al., 2002, 2003; Farfour and Yoon, 2016).

Processing and interpreting converted wave data is challenging (Granli et al., 1999; Gaiser, 1999; Stewart et al., 2002, 2003). In the classical PP and PS processing flow, one often needs to run registration algorithms to transform PS data to its equivalent image in PP-wave traveltime (Gaiser, 1996; Fomel, 2007; Compton and Hale, 2014). The latter allows us to compare 3D volumes of PP data with their associated PS volumes. One difficulty is the difference in the quality of pre-stack PP and PS volumes. In my research, I also investigate a 5D reconstruction flow that enables us to correctly preconditioning pre-stack PP and PS volumes before registration (Liu and Sacchi, 2004; Zwartjes and Sacchi, 2006).

## 1.1 Review of seismic data denoising

Random noise attenuation is a critical step in the field of seismic data processing. We can broadly categorize denoising methods into five groups:

- Methods that adopt linear prediction theory, also occasionally referred to as autoregressive modelling techniques
- Transform-based methods
- Reduced-rank matrix filtering techniques
- Methods based on multilinear (tensor) algebra
- Approaches that adopt machine learning techniques.

#### Methods that adopt linear prediction theory

Traditional methods inspired by linear prediction theory can be applied in the frequency-space ($f$-$\mathbf{x}$) domain (Canales, 1984; Gülünay, 1986; Chase, 1992; Gülünay, 2000) or in the

time-space ($t$-$\mathbf{x}$) domain (Abma and Claerbout, 1995). These methods utilize the auto-regressive (AR) (Akaike, 1969) signal model to estimate, from noisy observations, a filter capable of predicting the signal embedded in the seismic record. Prediction filters could introduce artifacts such as signal leakage into the estimated noise. These artifacts are a consequence of adopting the AR formulation, which is unable to model the additive noise term correctly. The noise term in AR modelling is an innovation rather than additive noise (Sacchi and Kuehl, 2001). Several authors have addressed this inconsistency. For instance, Soubaras (1994, 1995) not only discussed the inconsistency mentioned above but also proposed to alleviate signal leakage via $f$-$x$ projection filtering.

**Methods that adopt transforms and sparsity promoting techniques**

Transform-based techniques often used in conjunction with sparsity constraints have become a popular option for seismic data denoising and reconstruction. First, one estimates the coefficients that model the clean signal by procedures that promote sparsity. Then, the calculated coefficients can synthesize the denoised signal. Linear transforms, which define basis functions for this type of methods, include the Fourier transform (Sacchi et al., 1998; Liu and Sacchi, 2004), the Radon transform (Sacchi and Ulrych, 1995; Trad et al., 2002; Wang et al., 2010; Ibrahim and Sacchi, 2013) and the curvelet transform (Starck et al., 2002; Herrmann et al., 2007).

**Methods based on rank-reduction of matrices**

Incoherent noise removal can also be formulated as a matrix rank-reduction problem (Trickett, 2003) in the $f$-$\mathbf{x}$ domain. This method is called $f$-$xy$ eigenimage filtering. Similarly, random noise reduction and seismic data interpolation can be formulated via rank-reduction methods that operate on block Hankel matrices in $f$-$\mathbf{x}$ domain (Trickett et al., 2010a; Oropeza and Sacchi, 2011a; Gao et al., 2011).

**Methods based on multi-linear algebra**

There is no ambiguity on the definition of the rank of a matrix. Conversely, low-rank tensor approximations require attention. First of all, there is no unique form in which a tensor can decompose into smaller structures because the definition of tensor rank is non-unique (Kolda and Bader, 2009a). A tensor can be decomposed via the CP decomposition (Rokhlin and Tygert, 2008; Comon et al., 2009; Wang et al., 2015a; Battaglino et al., 2018). In this particular decomposition, a rank-one tensor is a volume computed via the multi-dimensional outer product of vectors. Similarly, one can decompose a tensor in terms of the

superposition of matrices and a small core tensor by adopting the high-order SVD (HO-SVD) . Kreimer and Sacchi (2011, 2012a) adopted the HO-SVD for seismic data denoising and 5D seismic data reconstruction. Alternatively, one can utilize the Parallel Matrix Factorization (PMF) method to represent a tensor with low-rank unfoldings (Xu et al., 2015; Sacchi et al., 2015). Sidiropoulos et al. (2017) provided a thorough overview of the applications of tensor techniques to signal processing, statistical data analysis, data mining, and machine learning.

#### Methods that adopt Machine Learning technqiues

Machine learning techniques have recently been proposed to remove additive Gaussian noise (Zhang et al., 2017). Machine learning methods are capable of achieving excellent results when applied for natural image denoising. Similar ideas can apply to seismic noise attenuation and the removal of interferences in simultaneous source seismic acquisition (Wang and Nealon, 2019; Wang and Chen, 2019; Richardson and Feller, 2019).

## 1.2 Review of seismic data regularization

Multicomponent seismic data reconstruction can be achieved via vector-field signal processing techniques. For instance, Stanton and Sacchi (2013) integrated quaternion Fourier transform into a Projection onto Convex Sets (POCS) algorithm to allow for the reconstruction of radial and transverse recordings in the offset-azimuth domain. These authors demonstrated an improvement in reconstruction quality by employing vector field techniques. Hou et al. (2016) proposed a method based on K-SVD to reconstruct multicomponent data of ocean bottom survey (OBS) simultaneously. Given the large travel time difference between PP- and PS-wave events, the proposed flow independently reconstructs PP and PS-wave 5D pre-stack volumes via a rank-reduction tensor completion method (Gandy et al., 2011; Liu et al., 2013; Xu et al., 2015).

Reconstruction methods based on reduced-rank techniques are divided into two categories. In the first category, the multi-dimensional data are rearranged into a multi-level block Hankel matrix, and a rank reduction algorithm is used to improve the $SNR$ and to reconstruct the data. Methods in this category are often named Cadzow reconstruction methods (Trickett et al., 2010b) or multichannel singular spectrum analysis (MSSA) reconstruction (Oropeza and Sacchi, 2011b; Gao et al., 2013). The second category of rank reduction methods encompasses techniques based on dimensionality reduction of multilinear arrays or tensors. Multi-dimensional seismic data are viewed as multilinear arrays, and dimensionality reduction techniques are directly applied to the multilinear array (Kreimer and Sacchi,

2012b; Kreimer et al., 2013; Gao et al., 2015). My work builds on recent efforts on simultaneous seismic reconstruction and denoising via tensor completion techniques. Specifically, I have adopted a tensor Higher-Order SVD rank-reduction in conjunction with an imputation algorithm (Kreimer and Sacchi, 2012b) to reconstruct and denoise PP and PS-wave 5D volumes. After denoising and regularizing the data via 5D tensor completion, the data are ready for signal registration and the subsequent estimation of subsurface parameters.

## 1.3 Review of multicomponent seismic data registration

Multicomponent seismic data registration is an important step before quantitative seismic data interpretation and joint AVO analysis (Lu et al., 2015). Registration involves mapping PS seismic reflections to the position of the corresponding PP reflections. It is well-known that the $V_p/V_s$ ratio can be estimated as a by-product of the registration process (Fomel et al., 2005). The $V_p/V_s$ ratio is a critical indicator for lithology and fluid properties (Stewart et al., 2003). In addition, the $V_p/V_s$ ratio has been also used for improving the characterization of unconventional reservoirs (Lines et al., 2005; Guliyev and Davis, 2007; Zuleta and Lawton, 2012).

Multicomponent seismic data registration is typically performed by methods based on cross-correlation functions. For instance, Gaiser (1996) introduced a correlation-based method to determine the long-wavelength components of the $V_p/V_s$ ratio. Geis et al. (1990) used both P- 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 and Kristiansen (2003) discussed different tools for obtaining and refining the $V_p/V_s$ ratio. These authors also studied methods to include seismic data registration in processing workflows to improve the quality of seismic images. Fomel (2007) defined an attribute named local correlation and applied it to multicomponent seismic image registration of a nine-component land survey. In the context of time-lapse seismic, Hale (2009) described a registration method to calculate the local correlation between two data sets. 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. Similarly, Yuan et al. (2008) adopted a global optimization approach via Simulated Annealing to minimize the normalized cross-correlation between PP-wave data and warped PS-wave data. Yuan et al. (2008) also accounted for frequency-domain differences by introducing time-variant spectrum whitening. Last, we also mention the work of Compton and Hale (2014) who proposed to use dynamic warping to align PS-wave traces to its corresponding PP-wave traces.

Registration algorithms map PS-wave data to its equivalent in PP-wave travel time. The

latter assists seismic data interpreters in correlating 3D volumes of PP-wave data with their associated PS volumes. Joint PP-, PS-wave pre-stack inversion can be applied to predict reservoir parameters, such as lithology and presence of fluids (Avseth et al., 2010).

## 1.4 Review of least-squares reverse-time migration

Reverse-time migration (RTM) is a two-way wave-equation based imaging technique first introduced for post-stack data migration (McMechan, 1983; Baysal et al., 1983; Whitmore, 1983). The technique was also adopted for imaging vertical seismic profiling (VSP) data using the excitation-time imaging condition (Chang and McMechan, 1986). Various numerical and real data examples demonstrated that RTM is well suited for imaging complex structures because it imposes no limitations on the dip angle of reflectors (Buur and Kühnel, 2008).

Conventional RTM is susceptible to strong migration artifacts (Etgen et al., 2009; Zhang and Sun, 2009). Linearized seismic imaging can be adopted to suppress migration artifacts and enhance the quality of seismic images. Lailly and Bednar (1983) and Tarantola (1984b) formulated seismic imaging as a linearized, least-squares inverse problem to estimate velocity perturbations from seismic data. Similarly, Bourgeois et al. (1989) explored the conditions for which the linearization is justifiable and observes that linearized inversion yields substantially sharper images than pre-stack migration. Lambaré et al. (1992) demonstrated that a linearized least-squares inversion, implemented via iterative methods, is capable of improving the spatial resolution of seismic images. Migration artifacts (footprint) may arise from incomplete data with limited recording aperture, coarse sampling and acquisition gaps in the data. Nemeth et al. (1999) coined the name least-squares migration (LSM) to identify imaging methods that adopt demigration operators to model the data. Nemeth et al. (1999) demonstrated the ability of LSM to attenuate acquisition footprint noise. Kühl and Sacchi (2003) introduced an imaging principle that extracts common image gathers (CIGs) in the ray-parameter domain and provide an LSM migration algorithm based on one-way wave-equation operators. Clapp (2005) used a regularized form of LSM to compensate amplitudes in inadequately illuminated regions of the seismic image. Kaplan et al. (2010) derived the forward and adjoint operators for shot-profile, one-way wave-equation LSM and also accounted for lateral migration velocity variations via the split-step approximation.

Contrary to the methods mentioned thus far, least-squares reverse-time migration (LSRTM) utilizes the two-way, rather than the one-way, wave-equation to construct the forward and adjoint operators used for imaging. By iteratively minimizing the difference between the observed and synthetic data, LSRTM delivers a "true amplitude" image and sharpens the reflectivity of subsurface reflectors (Dong et al., 2012). Østmo et al. (2002) developed a

frequency-domain formulation for LSRTM that solves the constant-density acoustic wave-equation using finite-difference methods. The large computational overhead of LSRTM hinders its implementation in 3D seismic exploration. To alleviate the computational burden, researchers have explored data compression algorithms centred around simultaneous-source data and regularized inversion (Dai et al., 2012; Xue et al., 2015; Cheng et al., 2016). Regularized LSRTM has also been used to attenuate crosstalk artifacts that arise when surface-related multiples are incorporated into the migration process (Zhang and Schuster, 2013; Wong et al., 2015; Tu and Herrmann, 2015). As an alternative to the standard least-squares data misfit, Zhang et al. (2014) proposed a generalized LSRTM method that maximizes the cross-correlation between the simulated and observed seismic data.

The LSM problem can be formulated in the image domain. Chavent and Plessix (1999) studied the Hessian matrix associated with image domain LSM and show that an optimal matrix of weights can be designed to restore the correct amplitudes of migrated events. Hu et al. (2001) applied a deblurring filter to the migrated image to reduce migration artifacts and to improve spatial resolution. Sjoeberg et al. (2003) adopted a similar approach through a 2D deconvolution technique. By assuming a 1D layered velocity model, Yu et al. (2006) introduced pre-stack migration deconvolution to approximate the inverse of the Hessian matrix. Guitton (2004) approximated the inverse of the Hessian matrix with a bank of non-stationary matching filters that partially captures the effects of least-squares inversion. Rickett (2003) compensated for uneven illumination in subsurface images through a normalization scheme that is appropriate for wave-equation migration algorithms. The scheme estimates weights from the ratio between the synthetic migration result and an initial reference model. Plessix and Mulder (2004) proposed amplitude-preserving migration weights based on the Born approximation of the acoustic wave equation. An advantage of LSM in the image domain, compared to the data domain, is that the problem can be posed in a target-oriented manner. Valenciano et al. (2006) introduced a target-oriented strategy that estimates a wave-equation least-squares inverse image. Target-oriented approaches are particularly useful when targeting localized reservoir targets under complex overburdens (Fletcher et al., 2016).

## 1.5 Contributions of this thesis

The main contributions of this thesis are summarized as follows:

- My research developed an algorithm where the randomized CP decomposition is used to attenuate incoherent noise of multicomponent seismic data in the pre-stack domain. The over-determined least-squares subproblem associated with CP decomposition can

be solved efficiently using a random sketching technique. The improved efficiency stems from the reduced size of the subproblems. The proposed algorithm shows superiorities in handling seismic data with static shifts and multi-dimensional seismic data when compared to traditional algorithms.

- I propose a new registration method to match PS-wave images to PP-wave images by minimizing the difference between the PP-wave and the warped PS-wave envelope. I adopt a parameterization of the $V_p/V_s$ ratio to reduce the number of unknowns, and guarantee is positiveness. This reparameterization represents $V_p/V_s$ ratio using cubic spline basis functions. Furthermore, I extend the conventional single-trace matching process to multi-channel registration using the Kronecker product.

- Techniques in the field of matrix and tensor completion that I developed for processing converted wave has inspired my work in the field of imaging. I propose an efficient algorithm for the least-squares reverse time migration (LSRTM) in the image domain. The Hessian matrix is approximated as the superposition of Kronecker products, and the Kronecker factors are estimated from preferential samples of the Hessian matrix. The efficiency of LSRTM can be improved significantly as the inversion process only involves operations of matrices of small size.

Overall my work contributes to the field of exploration seismology and to the subfields of multicomponent seismic data processing and seismic imaging. My thesis contributes to multicomponent seismic data processing by proposing denoising, regularization and registration methods for prestack PP and PS-wave dataset. My thesis contributes to seismic imaging by developing a new fast iterative method where the Hessian of the least-squares migration problem is approximated by the sum of Kronecker products.

## 1.6 Thesis overview

In **Chapter 2**, I first provide notations used for tensor algebra and introduce the definitions of several linear operators, such as Hadamard product, Kronecker product and Khatru-Rao product. Secondly, by drawing the connection to the rank of a matrix, the definition of the rank of a tensor is discussed. I also introduce tensor decomposition and three low-rank tensor models commonly used for seismic data processing, specifically, CP decomposition, high-order SVD and parallel matrix factorization.

In **Chapter 3**, I investigate the application of the CP decomposition to random noise attenuation in the $t$-$\mathbf{x}$ domain for pre-stack multi-dimensional seismic volumes. First, I discuss the alternating least squares method for CP decomposition (CP-ALS). I observe that

the CP-ALS method can be efficiently solved using a randomized technique which uniformly samples the rows of matrices tied to an over-determined subproblem. The solution is a reasonable approximation to the original problem. The latter leads to the fast randomized CP decomposition (Rand-CP-ALS) algorithm. I apply the proposed algorithm to synthetic and field seismic data volumes and compare their results against the ones obtained via traditional denoising methods. My analysis also considers random noise attenuation in the presence of statics, where the CP decomposition offers better noise attenuation performance than traditional $f$-$\mathbf{x}$ domain methods.

In **Chapter 4**, I propose to attenuate noise of pre-stack seismic data via 5D interpolation/reconstruction. I adopted a tensor completion method to reconstruct and enhance pre-stack PP and PS-wave data independently. Missing traces are successfully recovered and random noise are attenuated. To demonstrate the effectiveness of the proposed workflow, we test it on a 3D land multicomponent seismic data acquired in Central Alberta, Canada.

In **Chapter 5**, I formulate multicomponent seismic data registration as a non-linear optimization problem. I expand previous contributions in the field of seismic data registration in several ways. First, I parameterize the model of $V_p/V_s$ ratio in terms of splines. This reparameterization can reduce the number of unknowns and alleviate the nonlinearity of the optimization problem. Secondly, by adopting Kronecker products, the proposed algorithm extends the standard single-trace matching to multi-channel registration. It enforces smoothness constraints on the model of $V_p/V_s$ ratio along the spatial direction. The misfit function (the difference between PP-data and PS-data after warping) is minimized with respect to spline coefficients that represent the model of $V_p/V_s$ ratio. I solve the non-linear optimization problem using the Gauss-Newton method. Tests on synthetic and field data sets demonstrate the effectiveness and robustness of the proposed algorithms.

In **Chapter 6**, I propose a new and efficient LSRTM method formulated in the image domain. In the first part, I outline the problem of LSRTM based on the theory of Born approximation, which linearizes the seismic imaging problem. I observe that the Hessian matrix associated with LSRTM can be properly approximated as a superposition of Kronecker products. I describe two procedures for performing the Kronecker-based factorization: (1) a direct decomposition method when the full Hessian is known (2) by solving a low-rank matrix completion problem from a sparse set of elements from the Hessian. Once the Kronecker factors are obtained, the original LSRTM problem can be readily solved with the conjugate gradient (CG) method. Operations involving small, compact matrices replace expensive migration and demigration operations resulting in fast CG iterations. The increased computational efficiency stems from approximating the Hessian matrix as a superposition of Kronecker products. In the final section, I present numerical examples to evaluate the per-

formance of our proposed method against conventional LSRTM. I observe that our method is able to produce migration images comparable to LSRTM, but at a drastically reduced computational cost.

**Chapter 7** includes the conclusion of this thesis. I summarize the main content of this thesis and, more importantly, point out the limitations of the proposed method and discuss potential research directions to overcome these limitations.

# CHAPTER 2

# Multi-dimensional tensor algebra

## 2.1 Introduction

Tensor decomposition can be regarded as a generalization of matrix decomposition to tensors, which has been widely applied in statistics, signal processing, psychometrics and chemometrics in the last decades and shown advantages in the analysis of large datasets. A tensor is a multidimensional or N-way array. Recently, the application of tensor algebra has become a subject of study for seismic data processing. Excellent results have been achieved in the field of random noise attenuation and 5D seismic data interpolation. To facilitate the discussion of the next chapter, I provide the notation of tensor algebra, definitions of several common tensor operations and review three low-rank tensor decomposition algorithms that have been applied in the seismic data processing.

## 2.2 Notations

The following notations and definitions are from Kolda and Bader (2009a). See also Kreimer and Sacchi (2012a) where the same notation is adopted in a seismic processing context. Scalars, vectors and matrices are represented by lower-case, lower-case bold, upper-case bold letters, e.g., $d$ , $\mathbf{d}$ and $\mathbf{D}$ , respectively. We denote the $N$-dimensional seismic data as a tensor $\boldsymbol{\mathcal{D}}$ with element $D[i_1, \ldots, i_N]$ . The dimension of a multi-dimensional array is referred as the *order* denoted by $N$. The first dimension represents time and the remaining dimensions indicate the spatial coordinates. For instance, in a post-stack 3D seismic volume, the spatial dimensions could represent in-line and cross-line number. Dimension sizes are indicated by the set $\{I_1, \cdots, I_N\}$. A *mode-n* fiber of a tensor is defined by fixing all the

indices of the tensor except the $n^{th}$ index. Fibers are the high-order analogue of columns and rows of a matrix. For example, a *mode*-2 fiber of a third order tensor is designated as $\mathbf{D}[i_1, :, i_3]$.

## 2.3 Definitions

### 2.3.1 Tensor folding and unfolding

The *mode-n unfolding* or *matricization* of a tensor aligns the *mode-n* fibers as the columns of an $I_n \times \prod_{m \neq n} I_m$ matrix $\mathbf{D}_{(n)}$ where $n = 1, 2, \cdots, N$ and the tensor entry $D[i_1, \ldots, i_N]$ maps to the entry $(i_n, j)$ of the matrix $\mathbf{D}_{(n)}$ via the relation:

$$j = 1 + \sum_{k=1, k\neq n}^{N} (i_k - 1) J_k \quad where \quad J_k = \prod_{m=1, m\neq n}^{k-1} I_m \,. \tag{2.1}$$

The reverse operation of *unfolding* or *matricization* is called *folding* or *unmatricization*. Those two operations along *mode-n* are designated as follows

$$\begin{aligned} \mathbf{D}_{(n)} &= unfold_n(\boldsymbol{\mathcal{D}}), \\ \boldsymbol{\mathcal{D}} &= fold_n(\mathbf{D}_{(n)}), \quad n = 1, \ldots, N \,. \end{aligned} \tag{2.2}$$

Figure 2.1 shows the schematic representation of unfolding and folding for a third-order tensor (Kolda and Bader, 2009a; Sacchi et al., 2015).

### 2.3.2 Tensor multiplication

We can perform the multiplication between the $n$-mode tensor unfolding and a matrix. The $n$-mode product of a tensor $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{M} \in \mathbb{R}^{J \times I_n}$ is denoted as $\boldsymbol{\mathcal{D}} \times_n \mathbf{M}$ is of size $I_1 \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots I_N$. The element-wise representation is

$$(\boldsymbol{\mathcal{D}} \times_n \mathbf{M})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} d_{i_1 i_2 \cdots i_N} m_{j i_n}. \tag{2.3}$$

It is equivalent to each mode-$n$ fiber is multiplied by the matrix $\mathbf{M}$. So the idea can also expressed with unfolded tensors as

$$\mathbf{Y}_{(n)} = \mathbf{M}\mathbf{D}_{(n)}, \tag{2.4}$$

where $\mathbf{Y}_{(n)}$ is the $n$th-mode unfolding of the resultant tensor $\boldsymbol{\mathcal{Y}}$.

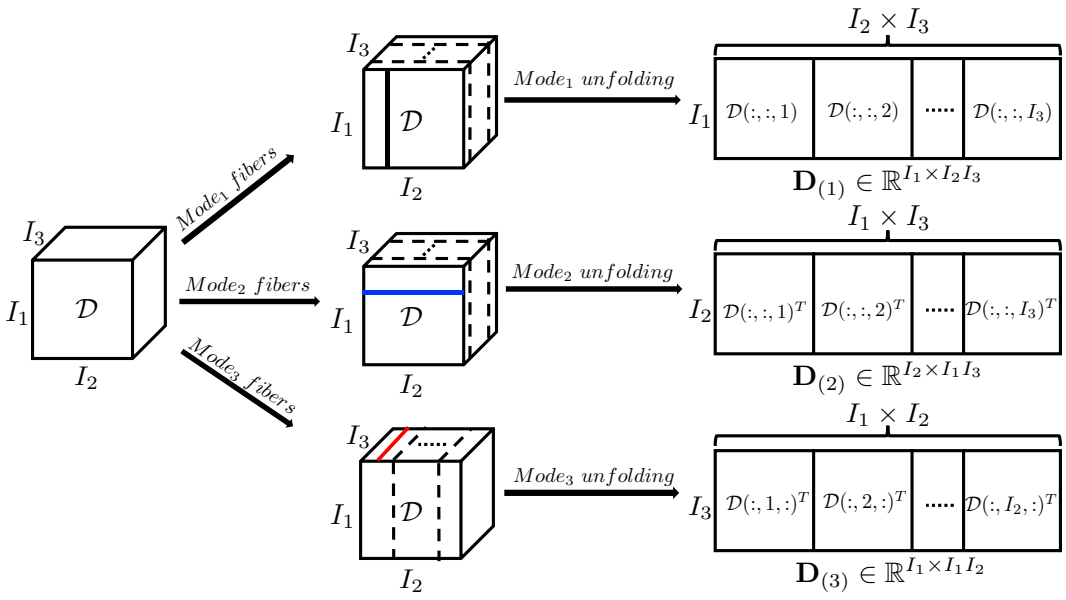


Figure 2.1: Schematic representation of unfolding and folding for a third-order tensor $\boldsymbol{\mathcal{D}}$ with size of $I_1 \times I_2 \times I_3$.

### 2.3.3 Hadamard product

The *Hadamard product* is defined as the element-wise product of two matrices of the same size. Given $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$, their *Hadamard product* is denoted by $\mathbf{A} \circledast \mathbf{B} \in \mathbb{R}^{I \times J}$. This operation can be extended to $N$ matrices, and we simplify the *Hadamard product* via the following expression

$$\underset{n}{\circledast} \mathbf{A}^{(\mathbf{n})} = \mathbf{A}^{(1)} \circledast \mathbf{A}^{(2)} \circledast \cdots \circledast \mathbf{A}^{(N)}. \tag{2.5}$$

### 2.3.4 Kronecker product

The *Kronecker product* of the matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ and the matrix $\mathbf{B} \in \mathbb{R}^{K \times L}$ is a block matrix size of $IK \times JL$, denoted by $\mathbf{A} \otimes \mathbf{B}$. Each block is obtained by multiplying the matrix $\mathbf{B}$ by one element of the matrix $\mathbf{A}$.

### 2.3.5 Khatru-Rao product

The *Khatri-Rao product* of matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$ is a matrix of size $I\,J \times K$, defined as the matching column-wise Kronecker product and denoted by $\mathbf{A} \odot \mathbf{B}$. Mathematically it is defined via the following expression

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_k \otimes \mathbf{b}_k \end{bmatrix},$$

where $\otimes$ represents the *Kronnecker product*. Similar to the *Hadamard product*, the *Khatri-Rao product* of a set of matrices can be simply denoted as

$$\bigodot_{n} \mathbf{A}^{(n)} = \mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \cdots \odot \mathbf{A}^{(1)}. \tag{2.6}$$

## 2.4 Tensor rank and decompositions

Tensor decomposition was first proposed by Hitchcock (1927) and later by Cattell (1944). The early applications of tensor decompositions appeared in psychometrics and chemometrics. Since then, tensor algebra has become extremely popular in these fields (Harshman, 1970; Carroll and Chang, 1970; Henrion, 1993; Smilde et al., 1994). In the last decade, we have seen interests in tensor decomposition expanded to other fields include signal processing (De Lathauwer and De Moor, 1998; Sidiropoulos et al., 2000; De Lathauwer and Vandewalle, 2004), computer vision (Vasilescu and Terzopoulos, 2002, 2003) and data mining (Liu et al., 2005; Sun et al., 2005, 2006) and more. In the remaining sections of this chapter, I review the CANDECOMP /PARAFAC (CP) (Carroll and Chang, 1970; Harshman, 1970) and Tucker (Tucker, 1966) tensor decompositions, which can be considered as a higher-order generalization of the matrix singular value decomposition (SVD). Another decomposition algorithm called Parallel matrix factorization (Xu et al., 2015; Gao et al., 2015) is also commonly implemented for seismic data reconstruction will be briefly reviewed in this chapter for the completeness of discussion.

### 2.4.1 Canonical polyadic decomposition

The CP decomposition represents a tensor into a superposition of rank-one tensors, which can be written as the outer product of vectors. For example, a 3rd order rank-one tensor

$$\boldsymbol{\mathcal{D}} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \mathbf{a}^{(3)}. \tag{2.7}$$

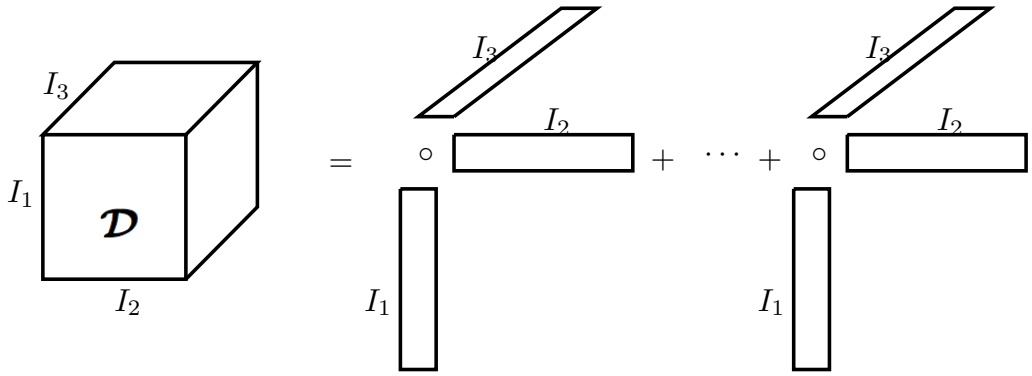


Figure 2.2: Schematic diagram of Candecomp / Parafac decomposition.

The symbol "$\circ$" indicate the vector outer product. One element of the tensor can be obtained as the product of the corresponding vector elements:

$$d_{i_1 i_2 i_3} = a^{(1)}_{i_1} a^{(3)}_{i_2} a^{(3)}_{i_3}. \tag{2.8}$$

With the definition of a rank-one tensor, the CP decomposition approximate a 3rd order tensor as

$$\boldsymbol{\mathcal{D}} = \sum_{r=1}^{R} \mathbf{a}^{(1)}_r \circ \mathbf{a}^{(2)}_r \circ \mathbf{a}^{(3)}_r, \tag{2.9}$$

Where $R$ is the target rank. Figure 2.2 show the schematic representation of the CP decomposition of 3rd order tensor. Different to the rank of matrix, there is no unique concept of tensor rank. For example, the rank of a tensor can be different over real or complex domains or sometimes the rank cannot even be determined (Kolda and Bader, 2009b). Once we know the value of the target rank $R$, many algorithms estimate factors alternatively to find the ones that fits the tensor best. The alternating least squares (ALS) method is one of the most commonly used and reliable algorithms (Carroll and Chang, 1970; Harshman, 1970). We refer readers to section 3.2 in Chapter 3 for more details about this method.

### 2.4.2 Tucker decomposition

The Tucker decomposition is a form of a high-order singular value decomposition (SVD). A tensor is represented as a core tensor multiplied by a matrix along each mode (Tucker, 1966).

We still take a 3rd-order tensor $\boldsymbol{\mathcal{D}}$ as an example. A 3rd-order tensor can be factorized as

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} c_{ijk} \mathbf{a}_i^{(1)} \circ \mathbf{a}_j^{(2)} \circ \mathbf{a}_k^{(3)}, \tag{2.10}$$

where $R_i$ is the target rank along the $i$th mode, $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is the core tensor, $\mathbf{A}^{(i)} \in \mathbb{R}^{I_i \times R_i}$ is the factor matrix along the $i$th mode and $\mathbf{a}_j^{(i)}$ is the $j$th column of $\mathbf{A}^{(i)}$. Element-wise, the Tucker decomposition in equation (2.10) can be expanded as

$$d_{pqr} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} c_{ijk} a_{pi}^{(1)} a_{qj}^{(2)} a_{rk}^{(3)}. \tag{2.11}$$

CP decomposition can be considered as a special case of the Tucker decomposition when $R_1 = R_2 = R_3$ and the core tensor $\boldsymbol{\mathcal{C}}$ is superdiagonal (Kolda and Bader, 2009a). I introduce the definition of Tucker decomposition in the context of a 3rd order tensor for simplification, the Tucker model can be generalized to an $N$th order tensor

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}. \tag{2.12}$$

Tucker (1966) introduced the first method for computing the Tucker decomposition and this method is known as the higher-order SVD (HOSVD), which is considered a generalization of the matrix SVD (De Lathauwer et al., 2000). However, HOSVD is not optimal in terms of minimizing the fitting error. Kroonenberg and De Leeuw (1980) developed an ALS algorithm to refine the decompositions obtained by HOSVD for 3rd order tensors. The algorithms initially designed for 3rd order tensor can be extended to $N$th order tensors straightforwardly.

### 2.4.3 Parallel matrix factorization

Xu et al. (2015) proposed a model to reconstruct a low-rank tensor by simultaneously applying low-rank matrix factorization to the matricization of a tensor along each mode. They aim to recovering a low-rank tensor $\boldsymbol{\mathcal{M}}$ from its partial observations $\boldsymbol{\mathcal{D}} = P_\Omega(\boldsymbol{\mathcal{M}})$. Where $\Omega$ is the set of observed elements. They solved the following optimization problem to recover $\boldsymbol{\mathcal{M}}$

$$\underset{\mathbf{X}_{(n)}, \mathbf{Y}_{(n)}, \boldsymbol{\mathcal{Z}}}{\text{minimize}} \sum_{n=1}^{N} \alpha_n ||\mathbf{X}_{(n)} \mathbf{Y}_{(n)}^T - \mathbf{Z}_{(n)}||_F^2, \text{ subject to } P_\Omega(\boldsymbol{\mathcal{Z}}) = \boldsymbol{\mathcal{D}} \tag{2.13}$$

where $\mathbf{X}_{(n)}, \mathbf{Y}_{(n)}$ are matrix factors along the $n$th mode, $\boldsymbol{\mathcal{Z}}$ is a common variable introduced to relate these matrix factorizations. The parameters $\alpha_n$ are weights placed on each mode

and satisfying the condition $\sum_{n=1}^{N} \alpha_n = 1$. The cost function represented in equation 2.13 can be solved via an alternating minimization method. In this case, one updates $\mathbf{X}_{(n)}$, $\mathbf{Y}_{(n)}$ and $\boldsymbol{\mathcal{Z}}$ alternatively using the following expression

$$\begin{aligned}
\mathbf{X}_{(n)}^{k+1} &= \mathbf{Z}_{(n)}^{k} \left(\mathbf{Y}_{(n)}^{k}\right)^{T} \left(\mathbf{Y}_{(n)}^{k} \left(\mathbf{Y}_{(n)}^{k}\right)^{T}\right)^{\dagger} \ , n = 1, \cdots, N \\
\mathbf{Y}_{(n)}^{k+1} &= \left(\left(\mathbf{X}_{(n)}^{k}\right)^{T} \mathbf{X}_{(n)}^{k}\right)^{\dagger} \left(\mathbf{Y}_{(n)}^{k}\right)^{T} \mathbf{Z}_{(n)}^{k} \ , n = 1, \cdots, N \\
\boldsymbol{\mathcal{Z}}^{k+1} &= P_{\Omega^c} \left(\sum_{n=1}^{N} \alpha_n fold_n \left(\mathbf{X}_{(n)}^{k+1} (\mathbf{Y}_{(n)}^{k+1})^{T}\right)\right) + \boldsymbol{\mathcal{D}}
\end{aligned} \tag{2.14}$$

where the symbol "$^{\dagger}$" is the Moore-Penrose pseudo-inverse, $\Omega^c$ is the complement set of $\Omega$. The final recovered low-rank tensor can be obtained from the estimated matrix factorizations using

$$\boldsymbol{\mathcal{M}} = \sum_{n=1}^{N} \alpha_n fold_n \left(\mathbf{X}_{(n)}^{k+1} (\mathbf{Y}_{(n)}^{k+1})^{T}\right) \tag{2.15}$$

Topics in the area of tensor algebra discussed in this chapter have received significant attention from the seismic data processing community as they enable us exploiting the high-dimensional structure of seismic data. For instance, regularly sampled noise-free seismic data can be represented by a low-rank tensor. Missing traces and random noise increase the rank of the tensor. Hence we can formulate multi-dimensional seismic data denoising and reconstruction as a low-rank tensor decomposition or completion problem.

# CHAPTER 3

# Random noise attenuation via the CP decomposition[1]

## 3.1 Introduction

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.

Traditional methods for random noise attenuation adopt prediction filters in $f-\mathbf{x}$ domain (Canales, 1984; Gülünay, 1986) or in $t-\mathbf{x}$ domain (Abma and Claerbout, 1995). Incoherent noise removal can also be formulated as a matrix rank-reduction problem (Trickett, 2003) in $f-x-y$ 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., 2010a; Oropeza and Sacchi, 2011a). There is no ambiguity in the definition of the rank of a matrix (Kolda and Bader, 2009a). 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 tensor of lower rank because there is no unique definition of tensor rank (Kolda and Bader, 2009a). 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, 2012a). Similarly, one can adopt Parallel Matrix Factorization (PMF) to represent a tensor for one with low-rank unfolding (Xu et al., 2015; Sacchi et al., 2015). Similarly, a tensor can be decomposed via the CP decomposition (Hitchcock, 1927; Harshman, 1970; Carroll and Chang, 1970). In

[1]A version of this chapter is published in Gao. W., and M. D. Sacchi, 2019, Random noise attenuation via the randomized CP decomposition, Geophysical Prospecting, early view.

this particular decomposition, a rank-one tensor is a volume computed by multi-dimensional outer product of vectors.

My work investigates the application of the CP decomposition to random noise attenuation in the $t$-$\mathbf{x}$ domain. First, I discuss the CP decomposition via the alternating least squares (CP-ALS) method. The CP-ALS can be efficiently solved by a randomized method that uniformly samples the rows of matrices associated with a least-squares subproblem (Rokhlin and Tygert, 2008; Wang et al., 2015a; Battaglino et al., 2018). The solution is a reasonable approximation to the original problem. The latter leads to the fast randomized CP decomposition (Rand-CP-ALS) algorithm. Finally, I apply the proposed algorithm to synthetic and field data volumes and compare their results against the ones obtained via traditional denoising methods. My analysis also considers random noise attenuation in the presence of statics, where the CP decomposition offers better noise attenuation performance than traditional $f$-$\mathbf{x}$ domain methods. The effectiveness of the proposed algorithm is also tested on a 3D multicomponent seismic dataset from the Western Canadian Basin.

## 3.2 Theory

### 3.2.1 Candecomp / Parafac Tensor Decomposition

The CP decomposition aims to approximate a $N^{th}$-order tensor as the sum of $R$ rank-one tensors

$$\boldsymbol{\mathcal{D}} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}\,, \tag{3.1}$$

where $R$ is called tensor rank. The factor vector along the $n^{th}$ mode is given by the vector $\mathbf{a}_r^{(n)}$ of length $I_n$. The symbol $\circ$ represents multi-dimensional *outer product*. Figure 2 shows the CP decomposition for a third-order tensor ($N = 3$). We can collect the factor vectors for a given mode and organize them into a factor matrix

$$\mathbf{A}^{(n)} = \begin{bmatrix} \mathbf{a}_1^{(n)} & \mathbf{a}_2^{(n)} & \cdots & \mathbf{a}_R^{(n)} \end{bmatrix} \in \mathbb{R}^{I_n \times R}, \tag{3.2}$$

where $\mathbf{A}^{(n)}$ is called the factor matrix for *mode-n*. The *mode-n* unfolding of the tensor $\mathbf{D}_{(n)}$ are related to the factor matrices via the relationship

$$\mathbf{D}_{(n)} = \mathbf{A}^{(n)} \mathbf{Z}^{(n)^T}, \tag{3.3}$$

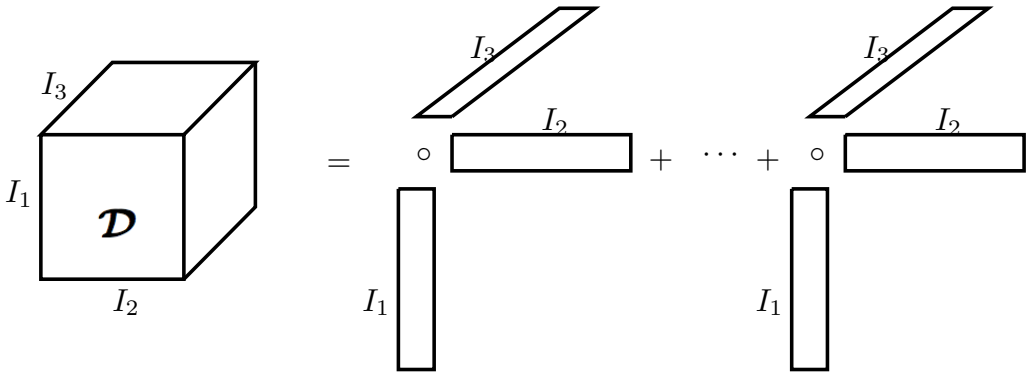


Figure 3.1: Schematic diagram of Candecomp / Parafac decomposition.

where $\mathbf{D}_{(n)}$ is the *mode-n* unfolding of the tensor $\boldsymbol{\mathcal{D}}$. The matrix $\mathbf{Z}^{(n)}$ is computed via the *Khatri-Rao product* of all factor matrices after excluding $\mathbf{A}^{(n)}$

$$\mathbf{Z}^{(n)} = \bigodot_{k \neq n} \mathbf{A}^{(k)} = \mathbf{A}^{(N)} \odot \cdots \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \cdots \odot \mathbf{A}^{(1)} \,. \tag{3.4}$$

The standard algorithm for finding each factor matrix $\mathbf{A}^{(n)}$ adopts the alternating least squares (CP-ALS) method (Harshman, 1970; Kolda and Bader, 2009a; Comon et al., 2009). The method entails a loop over all modes, fixing every factor matrix but $\mathbf{A}^{(n)}$. According to equation (3.3), the factor matrix $\mathbf{A}^{(n)}$ can be updated by minimizing the following cost function

$$\mathbf{A}^{(n)} = \underset{\mathbf{A}^{(n)}}{arg\,min} \, ||\mathbf{D}_{(n)} - \mathbf{A}^{(n)} \mathbf{Z}^{(n)^T}||_F^2 \,, \tag{3.5}$$

where $||\quad||_F$ is the *Frobenius* norm of a matrix. The system of normal equations (3.5) is given by

$$\mathbf{A}^{(n)} \left( \mathbf{Z}^{(n)^T} \mathbf{Z}^{(n)} \right) = \mathbf{D}_{(n)} \mathbf{Z}^{(n)}, \tag{3.6}$$

where the $\mathbf{A}^{(n)}$ is an $I_n \times R$ matrix. The size of $\mathbf{Z}^{(n)}$ is $\prod_{m \neq n} I_m \times R$. Therefore, the size of $\mathbf{Z}^{(n)T} \mathbf{Z}^{(n)}$ is $R \times R$. The right hand side of equation (3.6), $\mathbf{D}_{(n)}$, is a $I_n \times \prod_{m \neq n} I_m$ matrix and, consequently, the size of $\mathbf{D}_{(n)} \mathbf{Z}^{(n)}$ is $I_n \times R$. The factor matrices can be updated alternatively via the following expression

$$\mathbf{A}^{(n)} = \mathbf{D}_{(n)} \mathbf{Z}^{(n)} \;/\; \mathbf{Z}^{(n)^T} \mathbf{Z}^{(n)} = \mathbf{D}_{(n)} \mathbf{Z}^{(n)} \;\times\; inv(\mathbf{Z}^{(n)^T} \mathbf{Z}^{(n)}), \tag{3.7}$$

where the operator *inv* denotes matrix inversion. By taking advantage of the properties of

*Khatri-Rao product*, one can write

$$\left(\mathbf{A}^{(i)} \odot \mathbf{A}^{(j)}\right)^T \left(\mathbf{A}^{(i)} \odot \mathbf{A}^{(j)}\right) = \mathbf{A}^{(i)^T}\mathbf{A}^{(i)} \circledast \mathbf{A}^{(j)^T}\mathbf{A}^{(j)} \,. \tag{3.8}$$

which leads to a fast way to compute the term $\mathbf{Z}^{(n)^T}\mathbf{Z}^{(n)}$ in equation (3.7)

$$\mathbf{Z}^{(n)^T}\mathbf{Z}^{(n)} = \underset{m\neq n}{\circledast} \mathbf{A}^{(m)^T}\mathbf{A}^{(m)} \,. \tag{3.9}$$

Algorithm 1 summarizes the CP-ALS technique.

**Algorithm 1** CP-ALS Algorithm

**Inputs:**

full tensor $\boldsymbol{\mathcal{D}}$, target rank $R$, number of iterations $N$.

**Initialize:**

factor matrices $\mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}$.

**while** termination criteria is not satisfied **do**

**for** n=1 : N **do**

$\mathbf{X} = \underset{m\neq n}{\circledast} \mathbf{A}^{(m)^T}\mathbf{A}^{(m)}$

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

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

$\mathbf{A}^{(n)} = \mathbf{Y}/\mathbf{X}$

**end for**

**end while**

**return** factor matrices $\{\mathbf{A}^{(n)}\}$.

The input to Algorithm 1 is the observed data volume $\boldsymbol{\mathcal{D}}$ and the target rank $R$. The factor matrices of the CP-ALS method are typically initialized with random matrices. Alternatively, we can chose $\mathbf{A}^{(n)}$ as the $R$ leading left singular vectors of mode-n unfolding $\mathbf{D}_{(n)}$. The latter is named the higher-order SVD (HO-SVD) initialization. None of the initialization methods can guarantee convergence to a global minimum or even to a stationary point of the cost function. The iteration converges to a solution where the objective function ceases to decrease. In general, there is often little advantage of using leading singular vectors as initial factor matrices because it might lead to a local minimum (Kolda and Bader, 2009a). Global convergence of the CP decomposition cannot be guaranteed. A common practice to appraise convergence entails running the algorithm multiple times (with different random initializations) and evaluate if it leads to the same solution (Phan et al., 2015). My experiments did not encounter problems associated with convergence. In this article, I set the maximum number of iterations to 50. I also include a stopping criterion

that evaluates the difference of the cost function between consecutive iterations.

The expression in Line 7 of Algorithm 1 can be computationally demanding because it is associated with products of unfolded tensors and Khatri-Rao products of all factor matrices except one. These products account for most of the computation time of the CP-ALS algorithm. The direct computation of tensor unfolding is relatively slow as it involves the permutation of the elements of a tensor. Phan et al. (2013) proposed an ingenious method to reorganize the computation of Line 7; their technique only needs to perform one tensor unfolding along a specific dimension to obtain $\mathbf{Y}$ for each mode.

In contrast, Battaglino et al. (2018) take advantage of the over-determined nature of the problem expressed by equation 9 to propose an efficient CP-ALS solver. The new algorithm samples a specified number of rows from $\mathbf{Z}^{(n)}$ and the corresponding rows from $\mathbf{D}_{(n)}^{T}$ to obtain a new linear system of reduced size. Furthermore, the technique avoids tensor unfolding and Khatri-Rao products of factor matrices. The next section discusses this algorithm, which I designate as randomized CP-ALS (RAND-CP-ALS) decomposition.

### 3.2.2 An efficient CP Decomposition by Random Sketching: Randomized CP-ALS Decomposition

The intrinsic highly over-determined character of the cost function in equation (3.5) becomes more clear when it is expressed in its transposed form

$$\mathbf{A}^{(n)} = \underset{\mathbf{A}^{(n)}}{arg\,min} \, ||\mathbf{Z}^{(n)}\mathbf{A}^{(n)^T} - \mathbf{D}_{(n)}^{T}||_F^2. \tag{3.10}$$

The system is illustrated schematically in Figure 3.2, $\mathbf{Z}^{(n)}$ is denoted by the tall rectangle on the left, then the factor matrix $\mathbf{A}^{(n)^T}$, on right side is $\mathbf{D}_{(n)}^{T}$. In general, the number of rows $\prod_{m \neq n} I_m$ of $\mathbf{Z}^{(n)}$ is much larger than the tensor rank $R$, in other words, the number of constraints is more than the number of unknowns. The task is to reduce the workload of the CP-ALS algorithm drastically without sacrificing its accuracy. To achieve this purpose, I introduce a randomized CP-ALS algorithm, which uniformly samples a specific number of rows from $\mathbf{Z}^{(n)}$ and the corresponding rows from $\mathbf{D}_{(n)}^{T}$ to make a new linear system with much smaller size. Sketching is a technique for solving linear algebra problems where one constructs a smaller problem with a solution that is a reasonable approximation to the solution of the original problem (Rokhlin and Tygert, 2008).

The sampled rows can be formed without explicitly forming $\mathbf{Z}^{(n)}$ and unfolding the tensor $\boldsymbol{\mathcal{D}}$ along the $n^{th}$ dimension. Clearly, the latter does not involve products and sums, only permutations of a $N$-$D$ tensor. We should not dismiss the cost of the permutations. In fact,

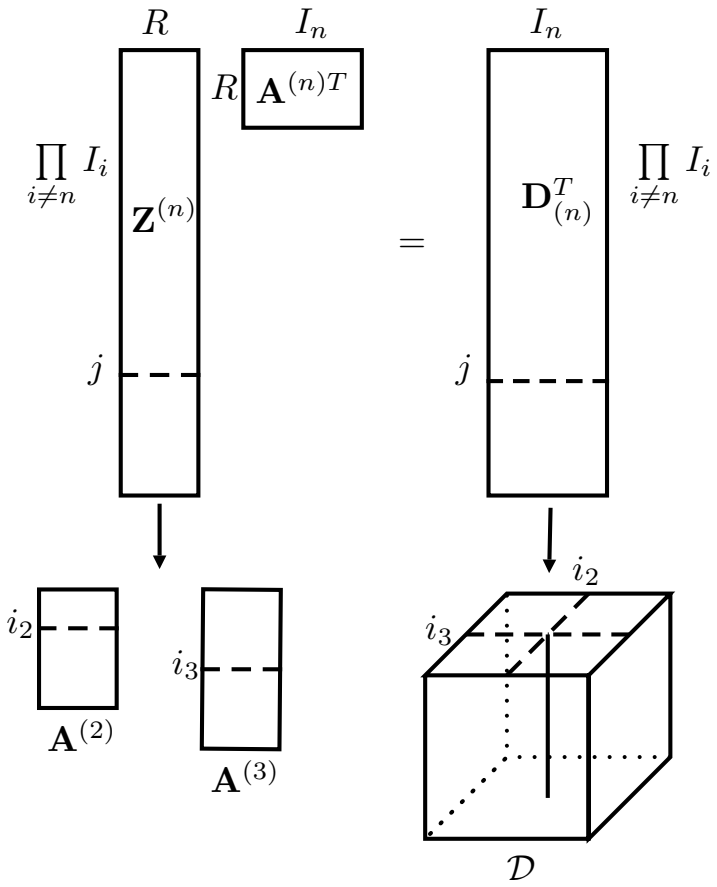


Figure 3.2: The over-determined linear equations of the alternating least squares method. I take the first mode of a *third* order tensor as an example. The sampled rows of $\mathbf{Z}^{(1)}$ correspond to the *Hadamard product* of the corresponding rows of the factor matrices (indicated by dashed lines on factor matrices). Similarly, sampled rows of $\mathbf{D}^T_{(1)}$ correspond to *mode*-1 fibers of $\boldsymbol{\mathcal{D}}$ (solid line inside cube), this sampling strategy can avoid explicitly forming $\mathbf{Z}^{(1)}$ and tensor unfolding operations.

the unfolding and folding steps usually take more than half of the total computation time of the CP-ALS algorithm for a moderate size third-order tensor (Phan et al., 2013). The $j^{th}$ row of $\mathbf{Z}^{(n)}$ is the *Hadamard* product of the corresponding rows of the factor matrices. Then, the $j^{th}$ row can be computed as follow

$$\mathbf{Z}^{(n)}[j,:] = \mathbf{A}^{(N)}[i_N,:] \circledast \cdots \circledast \mathbf{A}^{(n+1)}[i_{n+1},:] \circledast \mathbf{A}^{(n-1)}[i_{n-1},:] \circledast \cdots \circledast \mathbf{A}^{(1)}[i_1,:], \quad (3.11)$$

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

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

Similarly, the fiber $\boldsymbol{\mathcal{D}}[i_1, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N]$ corresponds to the $j^{th}$ row of $\mathbf{D}_{(n)}^T$. Figure 3.2 schematically shows the updating of the *mode*-1 factor matrix of a third-order tensor. The $j^{th}$ row corresponding to the *Hadamard* product of the $i_2^{th}$ row of $\mathbf{A}^{(2)}$ and the $i_3^{th}$ row of $\mathbf{A}^{(3)}$. The $j^{th}$ row of $\mathbf{D}_{(1)}^T$ is the *mode*-1 fiber $\mathbf{D}[:, i_2, i_3]$. I assumed that the desired number of sampled rows is $n_s$, which is 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. (2018) pointed out that a sample size $n_s = 10\, R \log(R)$ is sufficient for most applications. The randomized version of the CP-ALS algorithm is shown in Algorithm 2 and named Rand-CP-ALS. *Line* 2 determines the number of sampled rows from the target tensor rank and *line* 6 specifies the selected set, which is denoted by $\mathbb{S}$. *Line* 7 and *line* 8 represent forming the sampled rows of $\mathbf{Z}^{(n)}$ and $\mathbf{D}_{(n)}^T$ from the corresponding factor matrices $\{\mathbf{A}^{(n)}\}$ and the observed multi-dimensional array $\boldsymbol{\mathcal{D}}$, respectively.

**Algorithm 2** Rand-CP-ALS Algorithm

**Inputs:**

full tensor $\boldsymbol{\mathcal{D}}$, target rank $R$, number of iterations $N$.

**Initialize:**

factor matrices $\mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}$.

$n_s = 10\, R\, log_{10}(R)$

**while** termination the criteria is not satisfied **do**

**for** n=1 : N **do**

Define sample set $\mathbb{S} \in \{1, \ldots, \prod_{i \neq n} I_i\}$

$\mathbf{Z}_s \leftarrow Sketch(\mathbb{S}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, \cdots, \mathbf{A}^{(N)})$

$\mathbf{D}_s^T \leftarrow Sketch(\mathbb{S}, \boldsymbol{\mathcal{D}})$

$\mathbf{A}^{(n)} = \underset{\mathbf{A}^{(n)}}{arg\,min} \, ||\mathbf{Z}_s \mathbf{A}^{(n)T} - \mathbf{D}_s^T||_F^2$

**end for**

**end while**

**return** factor matrices $\{\mathbf{A}^{(n)}\}$.

The benefits provided by the random sketching technique are two-fold. Firstly, the algorithm avoids memory-intensive operations required by tensor unfolding. I achieve this goal by extracting the corresponding *mode-n* fibres directly from the *N-D* tensor. Secondly, the size of each linear least-squares subproblem is shrunk significantly by random sampling. The computational complexity for updating each factor matrix also reduces.

I compare the efficiency of CP-ALS and Rand-CP-ALS by applying them to two synthetic tensors designed to have rank 20. The tensors are contaminated with 50% random noise; one is a third-order tensor size of $500 \times 300 \times 300$. The second example is a fourth-order tensor size of $200 \times 80 \times 80 \times 80$. I conducted numerical experiments with an algorithm written in Julia 1.1 and run on an Intel i5 3.2 GHz machine with 8GB memory. Julia is a high-level dynamic programming language devised for numerical analysis and computational science (Lubin and Dunning, 2015). The target rank is also chosen to be 20. Figure 3.3 shows the convergence rate versus computation time for CP-ALS and Rand-CP-ALS. The vertical axis indicates the signal-to-noise ratio (SNR) is defined as

$$SNR = 10 \cdot log_{10} \left( \frac{||\boldsymbol{\mathcal{D}}||_F^2}{||\tilde{\boldsymbol{\mathcal{D}}} - \boldsymbol{\mathcal{D}}||_F^2} \right), \tag{3.12}$$

where $\boldsymbol{\mathcal{D}}$ is the clean tensor and $\tilde{\boldsymbol{\mathcal{D}}}$ indicates the recovered tensor obtained by applying low-rank CP decomposition to the contaminated tensor.

In Figure 3.3, stars and triangles indicate 5 iterations of CP-ALS and Rand-CP-ALS, respec-

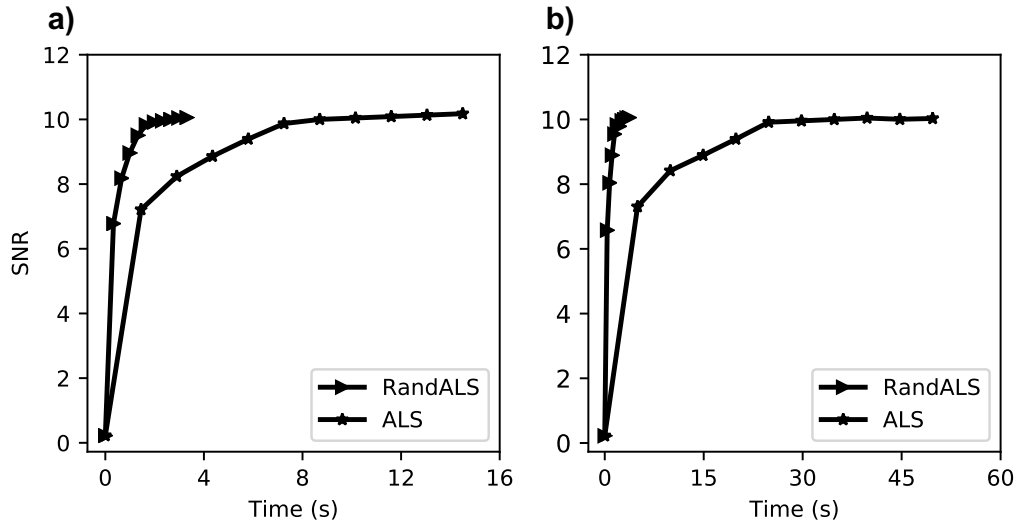


Figure 3.3: Convergence rate versus computation time. (a) Third-order tensor size of $500 \times 300 \times 300$. (4) Fourth-order tensor size of $200 \times 80 \times 80 \times 80$.

tively. In this comparison, I set the maximum number of iterations for the two algorithms to be 50. Both of them reach a SNR around 10 dB. One can observe that Rand-CP-ALS is 5 times faster than the CP-ALS algorithm for the third-order tensor. For the fourth-order tensor, the speed-up is even more significant (about 15 times). It is worth noting that the computation time of Rand-CP-ALS for both third-order and fourth-order tensors are almost the same. This is because I uniformly sampled the same number of rows for both cases. Therefore, the total computation workload for the two tensors is nearly identical.

### 3.2.3 Relation to other low-rank tensor decompositions

Higher-order SVD, which is also called Tucker decomposition, decomposes a tensor as a core tensor multiplied by a matrix along each mode, it can be expressed as

$$\boldsymbol{\mathcal{D}} \approx \boldsymbol{\mathcal{C}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} = \sum_{r_N=1}^{R_N} \cdots \sum_{r_2=1}^{R_2} \sum_{r_1=1}^{R_1} c_{r_1 r_2 \cdots r_N} \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \cdots \circ \mathbf{a}_{r_N}^{(N)}, \quad (3.13)$$

where $\boldsymbol{\mathcal{C}}$ is the core tensor, the symbol $\times_n$ represent tensor-matrix multiplication along $n^{th}$ mode. The matrix $\mathbf{A}^{(n)}$ is the factor matrix in the $n^{th}$ mode. The CP decomposition can be considered as a special case of the Tucker decomposition when the core tensor $\boldsymbol{\mathcal{C}}$ is a diagonal tensor with elements equal to 1 (Kolda and Bader, 2009a).

The Parallel Matrix Factorization (PMF) method decomposes a $N^{th}$-order tensor via the following sum

$$\boldsymbol{\mathcal{D}} \approx \sum_{n=1}^{N} \mathbf{fold}_n \left( \mathbf{U}^{(n)} \mathbf{V}^{(n)^T} \right) , \tag{3.14}$$

where the symbol $\mathbf{fold}_n$ represents matrix folding into a tensor along the $n^{th}$ mode. Tensor folding includes two steps: reshaping a matrix into a multi-dimensional array, then followed by permutation of entries of this array. The matrices $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are factor matrices and usually tall matrices (more rows than columns).

## 3.3 Examples

### 3.3.1 Synthetic data examples

To illustrate the performance of the proposed algorithms, I generate a synthetic $3D$ volume ($t$-$xy$) that includes 5 seismic events affected by static shifts. Figure 3.4a shows the uncorrupted data. The 5 events were first designed to have linear, parabolic, hyperbolic trajectories. Then, I applied smooth random temporal shifts to simulate long wavelength statics. Smoothness in static shifts was introduced by convolving random shifts with a 2D Gaussian kernel with a width of 5 traces (standard deviation).

I corrupted the clean data with additive band-limited (3-100 Hz) Gaussian noise (Figure 3.4b) with a SNR equal to $-2.4$. The size of the data volume is $500 \times 200 \times 200$, and the time sampling rate is 4 ms. To assist the visualization of the synthetic data cube, Figures 3.4c-e display clean slices extracted from the data cube in Figure 3.4a. Yellow dash lines in Figure 3.4a indicate the extraction location of the slices mentioned above. Figures 3.4f-h display the corresponding noisy sections. I also compare the proposed method against three other commonly used denoising techniques. Specifically, they are $f$-$xy$ prediction filtering (Chase, 1992), $f$-$xy$ eigenimage filtering (Trickett, 2003) and MSSA (Oropeza and Sacchi, 2011a). I adopted windowing strategy for all denoising tests. The three comparison methods operate in the frequency-space domain. Therefore, I partitioned the data cube in its two spatial dimensions with a sliding window of size $250 \times 32 \times 32$, the overlap between neighbouring windows is a quarter of the window width (8 traces). The working frequency band ranges from 1 to 50 Hz. The proposed method (Rand-CP-ALS) operates in the time domain. I partitioned the data cube both in time and space. The window size is $32 \times 32 \times 32$. Once I defined the window size and window overlap, only one parameter requires selection for each technique. They are the length of prediction filters ($f$-$xy$ prediction filtering), the number of eigenimages ($f$-$xy$ eigenimage filtering), the target rank of the block Hankel matrix (MSSA) and the target rank for the CP decomposition. These parameters are chosen to give the

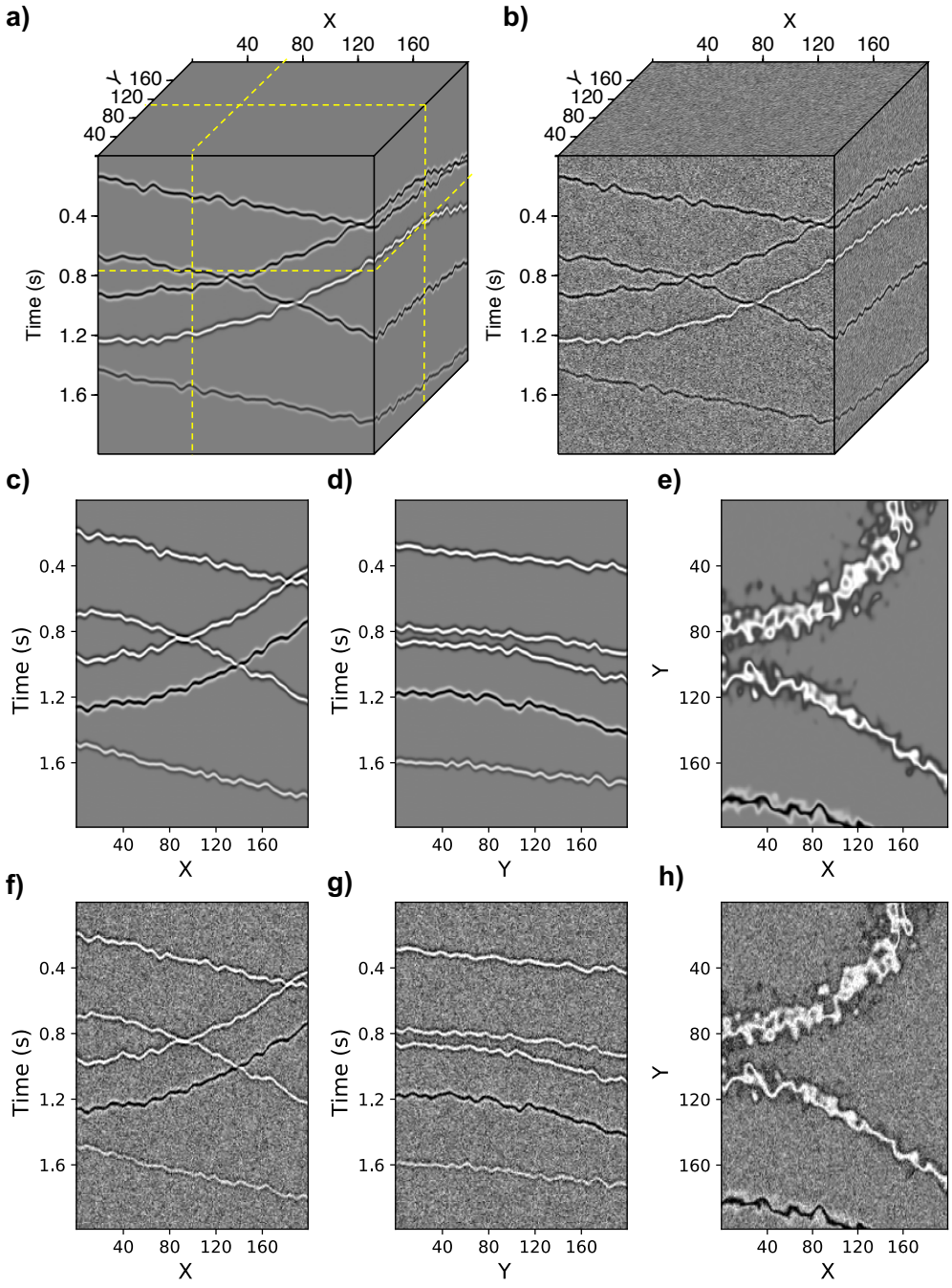


Figure 3.4: (a) Synthetic clean seismic cube. (b) Noisy data volume contaminated by band-limited Gaussian noise. (c-e) Frontal, lateral and time slices of the clean seismic cube. (f-h) The corresponding noisy slices.

Table 3.1: Comparison of different denoising methods, the first column indicates the method, the second column show the parameter selected for each algorithm. The third column indicates the SNR of processed results. The fourth column is the computation time for one window.

| Algorithm | parameter | SNR | CPU Time (s) |
|---|---|---|---|
| $f$-$xy$ eigenimage | 3 | 3.82 | 1.02 |
| MSSA | 12 | 7.08 | 3.03 |
| $f$-$xy$ prediction | 3 | 8.71 | 0.83 |
| Rand-CP-ALS | 11 | 8.93 | 0.26 |

best denoising performance in terms of SNR improvement. Table 1 summarizes the selected parameters, the SNR of the denoised data and the computational time for each method. Frontal sections of denoised data are in Figure 3.5. The first row shows the clean data (Figure 3.5a), the contaminated data (Figure 3.5b) and the additive noise (Figure 3.5c). The remaining rows display the denoised data and estimated noise in conjunction with tags indicating the algorithm. By comparison, $f$-$xy$ eigenimage is the least capable of removing the band-limited Gaussian noise with statics, then followed by MSSA. Rand-CP-ALS yields a slightly better SNR over $f$-$xy$ prediction filtering. The MSSA algorithm performance deteriorates by the presence of statics because the technique assumes data composed of a finite superposition of constant dip events (Oropeza and Sacchi, 2011a). To examine my results in more detail, I magnified the part indicated by the yellow frames in Figure 3.5a and plotted them in wiggles in Figure 3.6. I arranged the amplified parts using the layout of Figure 3.5. A careful comparison of my results (panels in the second column) indicates that Rand-CP-ALS generate the best denoising result, and this observation is consistent with the SNR reported in Table 1. Through this synthetic example, I observe that the CP decomposition performs better than the other 3 methods at preserving events with curvature and static time shifts.

Another advantage of the low-rank tensor filtering is its efficiency. I summarize the selected parameters, the SNR of the denoised cube and the computational time of the methods tested by my experiment in Table 1. The first column indicates the implemented algorithm; the second column shows the selected parameter, the third column is the SNR of the results, and the fourth column is the computational time in seconds. The *Rand-CP-ALS* method takes the least computational time. The SNR of the result obtained via the CP decomposition is higher than that of the three comparing methods.

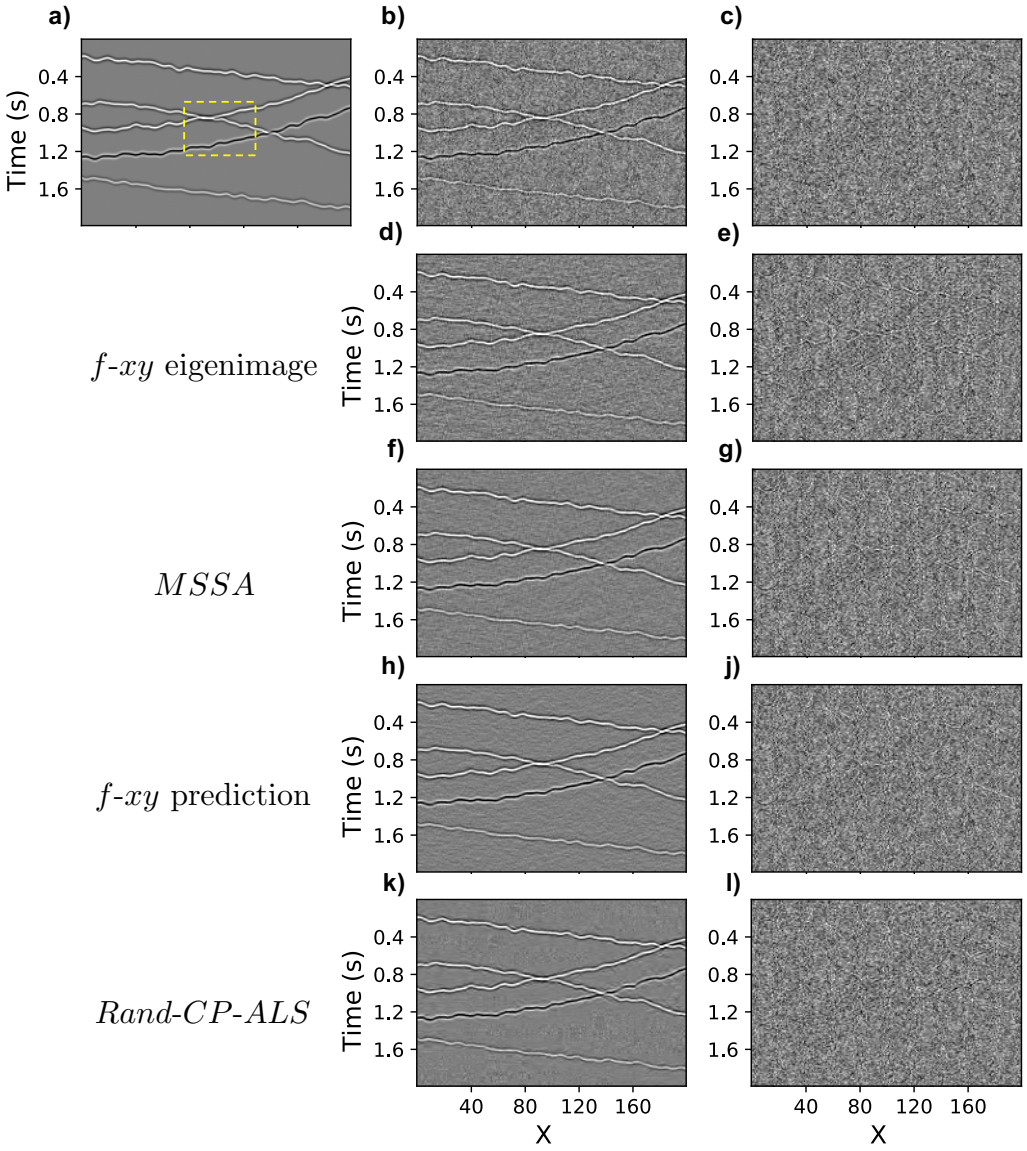


Figure 3.5: Denoising results of one frontal slice. (a) Clean synthetic data. (b) Noisy data. (c) Band-limited Gaussian noise. (d-e) The results of $f$-$xy$ eigenimage and its removed noise. (f-g) The result of MSSA and its removed noise. (h-j) The results of $f$-$xy$ prediction and its removed noise. (k-l) The results of Rand-CP-ALS and its removed noise.

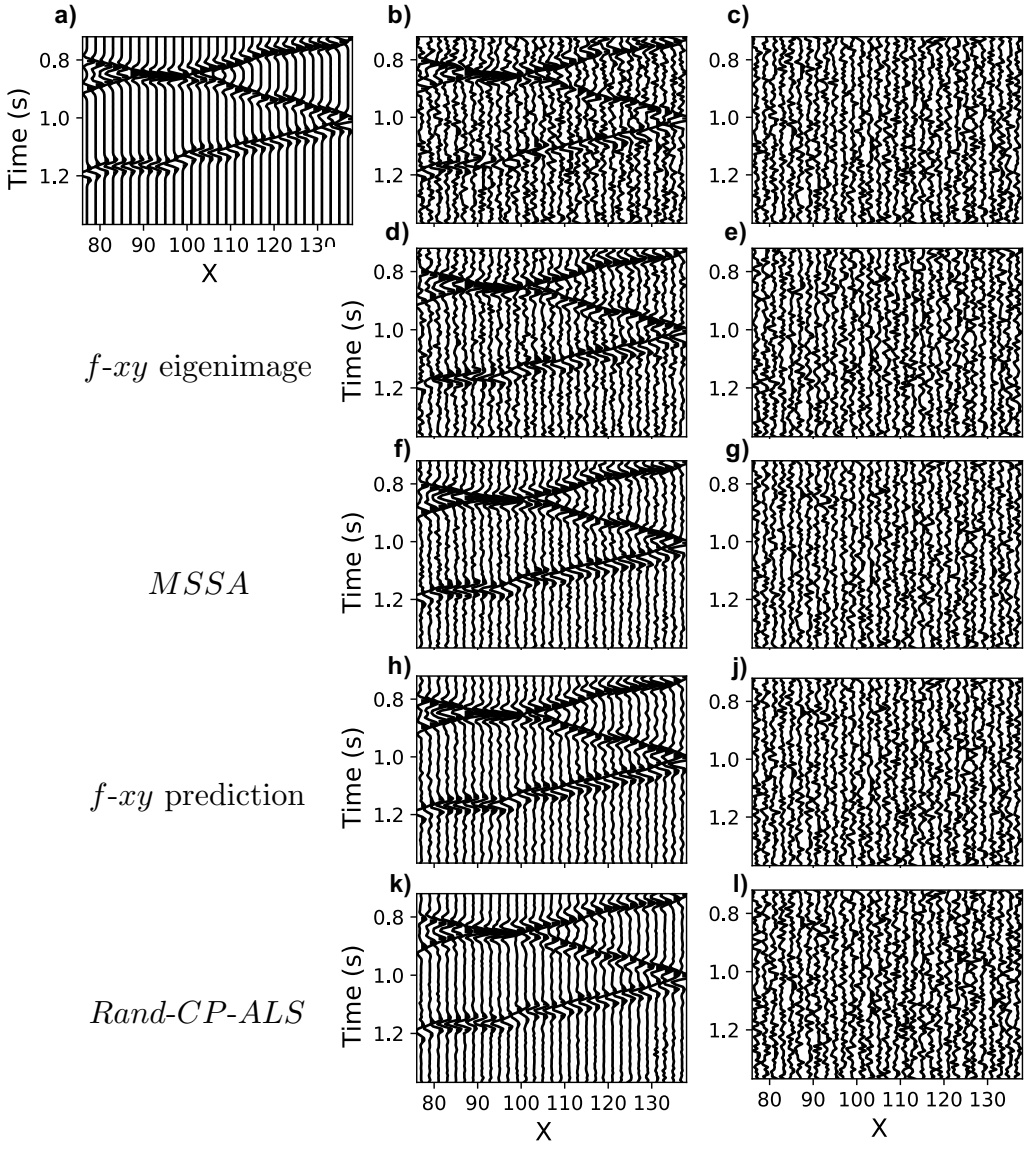


Figure 3.6: Magnified part of the frontal slice. (a) Clean synthetic data. (b) Noisy data. (c) Band-limited Gaussian noise. (d-e) The results of $f$-$xy$ eigenimage and its removed noise. (f-g) The result of MSSA and its removed noise. (h-j) The results of $f$-$xy$ prediction and its removed noise. (k-l) The results of Rand-CP-ALS and its removed noise.

### 3.3.2 Field data examples

I also test the proposed method on a land 3D post-stack data cube and on a prestack multicomponent pre-stack 4D volume.

#### 3D Seismic volume from the Western Canadian Basin

The data cube corresponds to a seismic survey in Central Alberta (Figure 3.7a). The size of the data cube is $750 \times 180 \times 120$ samples and the time sampling rate is 4 ms. To test the performance of the proposed method on random noise attenuation, I contaminated the data with band-limited Gaussian noise. Figure 3.7b shows the manufactured noisy data cube. The original sections and their slices are displayed in the second row (Figures 3.7c-e), and the noisy counterparts are shown in the third row (Figures 3.7f-h).

The technique $f$-$xy$ eigenimage filtering is omitted from the comparison because its performance was weak compared to MSSA and $f$-$xy$ prediction filtering. Similar to the synthetic data example, a windowing strategy was implemented. The length of the prediction filter is 7. I choose the target rank $R = 6$ and $R = 15$ for the MSSA method and the CP decomposition, respectively. I show the denoised sections in Figures 3.8a-c and the removed noise in Figures 3.8d-f. From the left side, the results of MSSA, $f$-$xy$ prediction and Rand-CP-ALS are displayed sequentially. These 3 algorithms achieve similar results because the geological structures of the exploration area are relatively flat. The seismic events present low spatial complexity, and hence, they can be handled equally well by the 3 methods. This dataset includes paleo-channels and their preservation after denoising could become a challenge for noise attenuation methods. Time slices, which are indicated by the yellow dash lines in Figure 3.7, are shown in Figure 3.9. Figures 3.9 a-c display the results after processing, and the corresponding removed noise is plotted in the second row (Figure 3.9d-f). I found that MSSA and $f$-$xy$ prediction filtering tend to over-smooth the features of the paleochannel. The $CP$ decomposition can better preserve the boundaries of the paleochannel. My argument is also supported by examining the noise sections (Figures 3.9d-f). Signal leakage associated with the paleochannel, which is marked by a yellow arrow, is observable in Figures 3.9d and e.

#### Denoising of a pre-stack volume

The previous two examples demonstrate the noise attenuation capability of the CP decomposition on 3D seismic cubes. Rand-CP-ALS are also capable of processing higher dimension seismic data. My last example is a 3D pre-stack multicomponent dataset acquired in Central Alberta, Canada. This dataset was acquired over a mature oil field in order to better

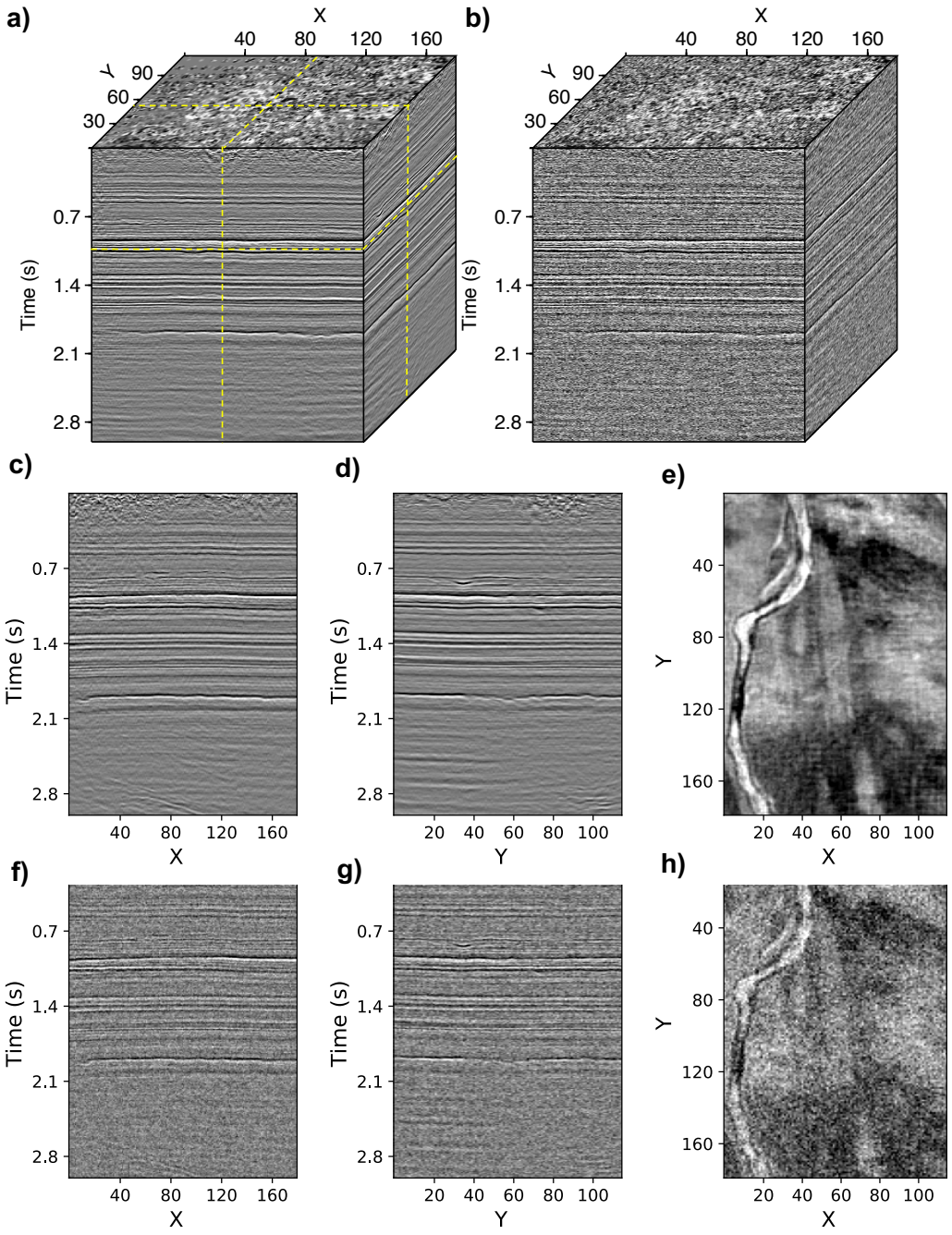


Figure 3.7: 3D land seismic volume. (a) Original seismic cube with a small amount of noise. (b) Synthetically contaminated volume by band-limited Gaussian noise. (c-e) Frontal, lateral and time slices of the original data. (f-h) The corresponding noisy slices.

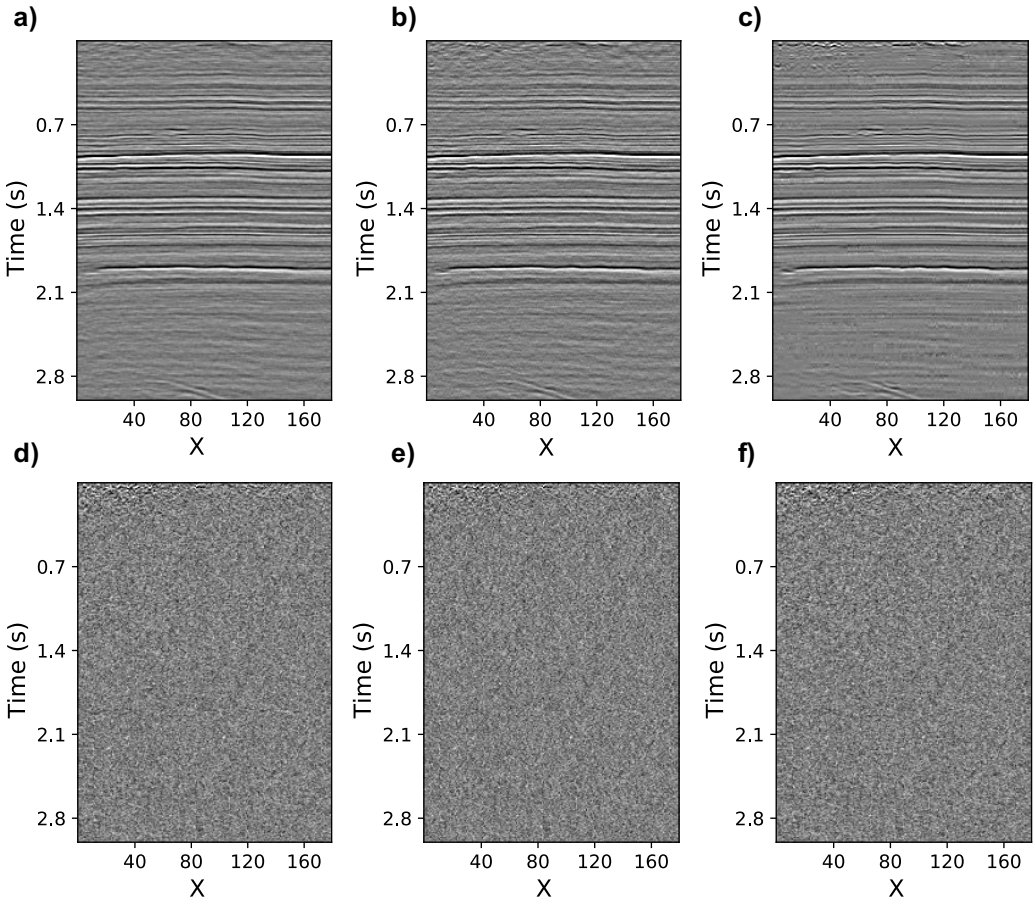


Figure 3.8: Denoising results through different approaches. (a) $MSSA$, (b) $f$-$xy$ prediction filtering (c) Rand-CP-ALS. (d-f) The noise sections removed by each method.

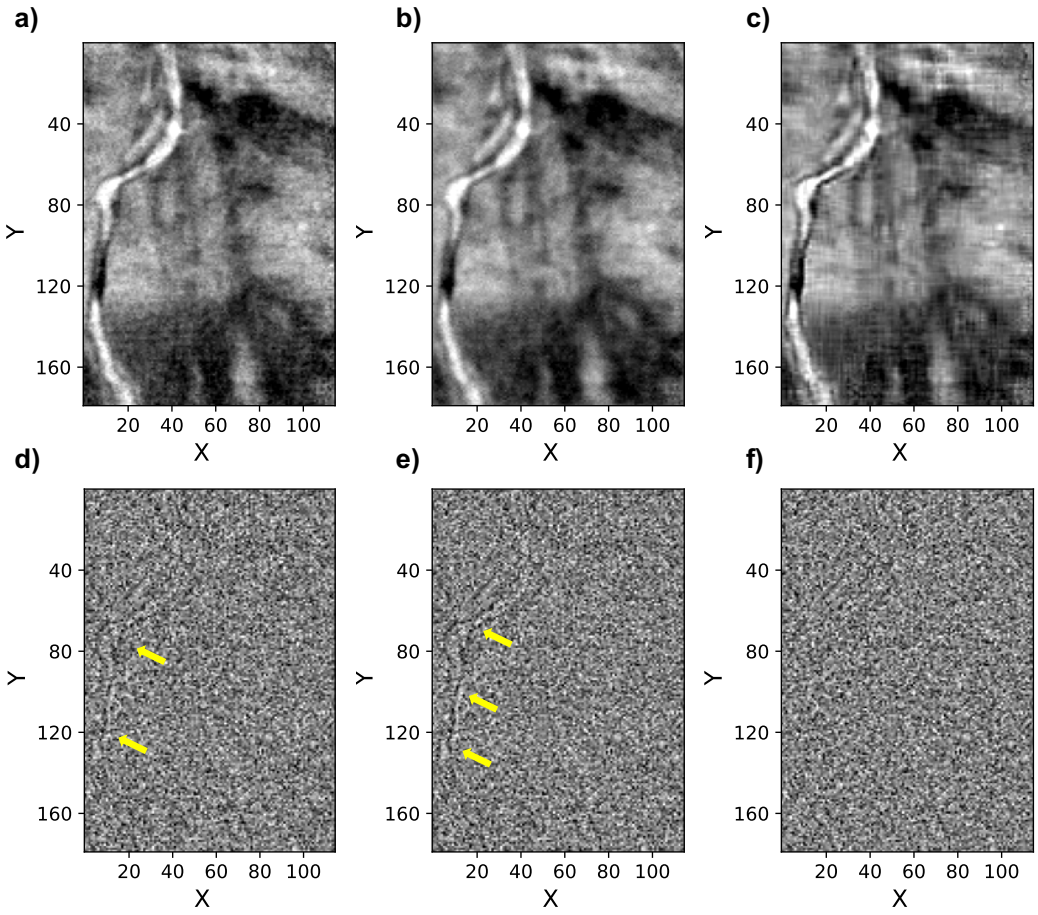


Figure 3.9: Denoising results through different approaches. (a) $MSSA$, (b) $f$-$xy$ prediction filtering (c) Rand-CP-ALS. (d-f) The noise sections removed by each method.

understand the characteristics of a waterflood operation (Perz et al., 2016). I cut out part of the dataset for testing the proposed algorithm. The time sample interval is 4 ms. The time length of PP-wave and PS-wave records is 2.4 s and 4.4 s, respectively. The reason for choosing a different record length for PP-wave and PS-wave data is that PS-wave reflections arrive at the surface later than PP-wave events. The data are binned on a $20 \times 20$m CDP grid; each PP-wave CDP gather includes 56 traces which are arranged from near to far offset. On the other hand, the PS-wave data are sorted into common conversion point (CCP) gathers, and each gather includes 36 traces. Seismic events were flattened via the NMO correction. The pre-stack PP-wave data is a fourth-order tensor of size $600 \times 56 \times 150 \times 260$. The PS-wave data is represented by a tensor of size $1100 \times 36 \times 150 \times 260$. The dimensions correspond to time, offset, cross-line number and in-line number. The proposed method is also implemented with a windowing strategy, and each window includes all the traces with different offset. In other words, I only partition the fourth-order tensor along time, cross-line number and in-line number directions. The window length is 36 for the three dimensions. I choose target ranks of 16 and 12 for the PP-wave and PS-wave data, respectively. The relatively smaller rank for PS-wave data is selected because the converted-wave data is noisier than the corresponding PP-wave data.

Figure 3.10 shows the result of PP-wave data. The post-stack cube before denoising (Figure 3.10a) and after denoising (Figure 3.10b) are obtained by stacking the offset dimension. To facilitate the examination of the results, I display the sections before and after processing in the second and third row of Figure 3.10, respectively. The locations of these sections are indicated by the yellow dash-lines in Figure 3.10a. The random noise is attenuated after processing, especially for the portion of the data between 1.0 s and 1.5 . The flat events are barely seen from Figures 3.10c-d and the strong noise is successfully removed after processing (Figures 3.10f-g).

Figure 3.11 shows the processing results of two PP-wave CDP gathers. The (in-line, cross-line) number of these two CDP gathers are $(30, 30)$ and $(120, 200)$, respectively. Start from left side, the panels in the first column display the traces before processing, panels in the second column show the gathers after processing and panels in the third column show the removed noise.

Figure 3.12 shows the result of PS-wave data. The post-stack cube before denoising (Figure 3.12a) and after denoising (Figure 3.12b) are obtained by stacking over the offset dimension. Similar to the PP-wave data, I display the sections before and after processing in the second and third row of Figure 3.12, respectively. The yellow dash-lines indicate the locations of these sections in Figure 3.12a. Comparing the panels in the second and third rows, I observe that seismic events appear much cleaner after denoising. The proposed algorithm can attenuate high amplitude noise at the deeper part of the record ($> 3.0$s) while preserving

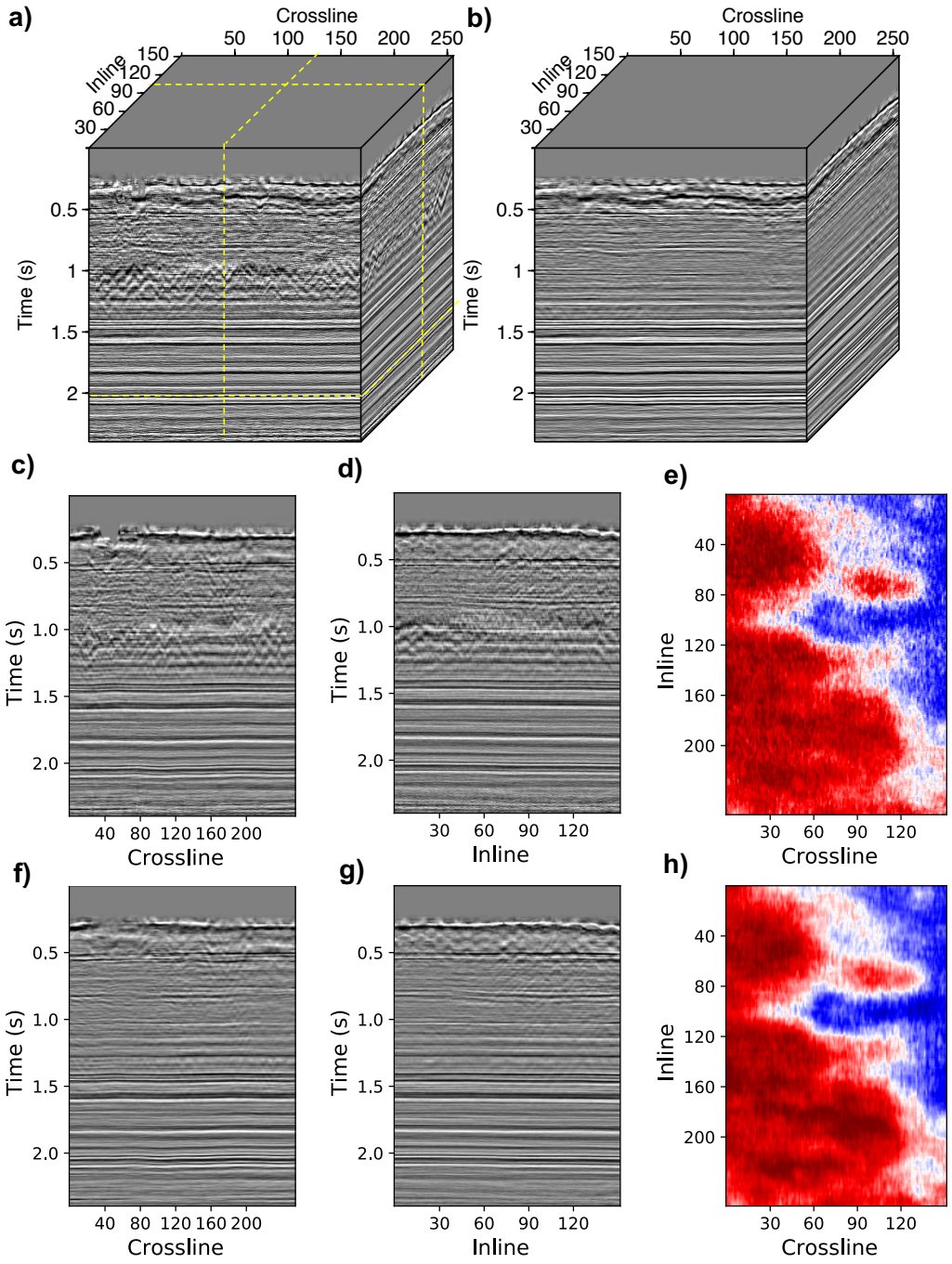


Figure 3.10: 3D PP-wave land data volume. (a) Post-stack volume before denoising. (b) Post-stack volume after denoising. (c-e) Inline, crossline sections and time slice of the post-stack PP-wave volume before denoising. (f-h) Inline, crossline sections and time slice of the post-stack PP-wave volume after denoising.

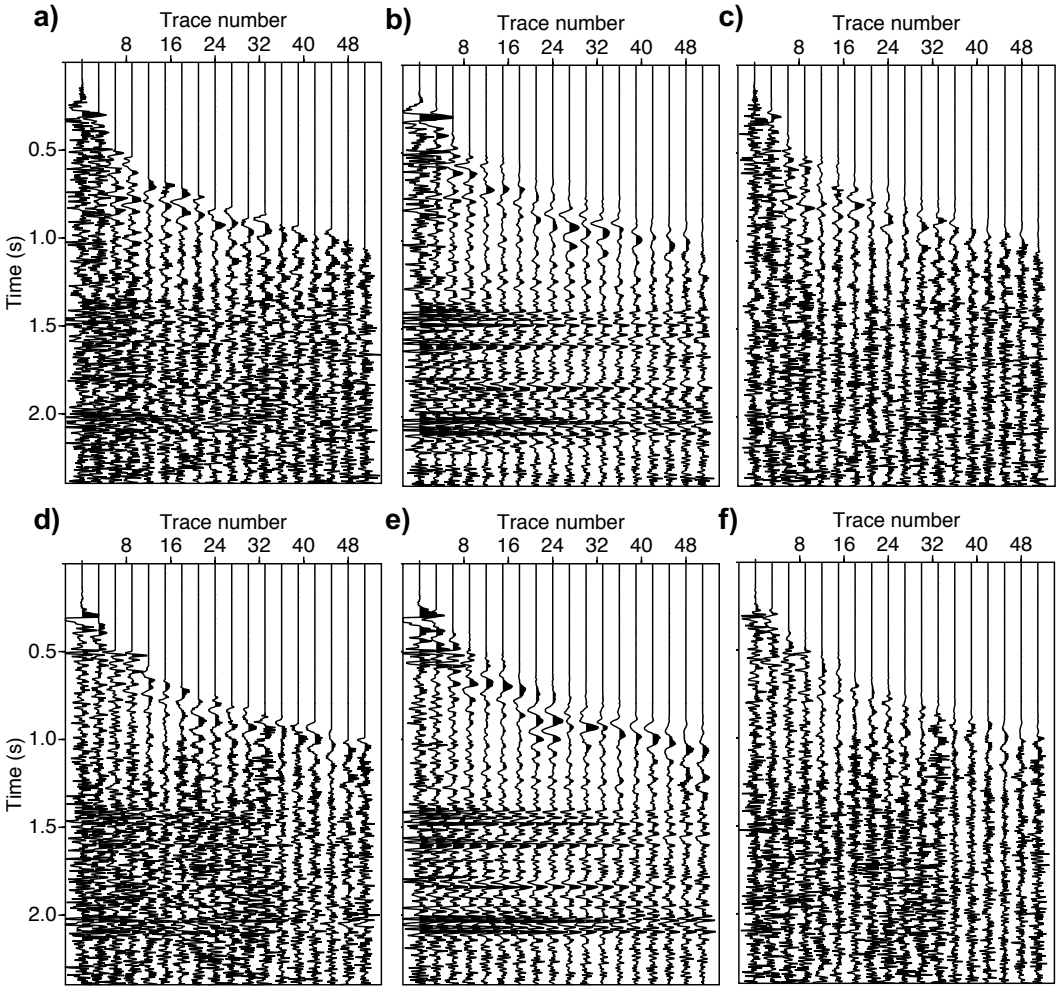


Figure 3.11: CDP gathers of NMO-corrected PP-wave data. (a) PP-wave CDP gather before denoising. (b) The same PP-wave CDP gather after denoising. (c) The removed noise. (d-f) The denoising result of another PP-wave CDP gather.

weak signals. Also, the amplitude map becomes more continuous after processing (Figure 3.12h).

Figure 3.13 shows the processing results of two PS-wave CCP (common conversion point) gathers. The in-line and cross-line numbers of these two CCP gathers are also $(30, 30)$ and $(120, 200)$, respectively. One important feature of converted-wave data is that near-offset reflections are weaker than mid and far offset reflections. This feature is observable from the CCP gathers after processing in Figures 3.13 b and e.

## 3.4 Discussion

Other tensor decomposition algorithms, such as HO-SVD and PMF, have been applied for seismic data processing in the $f-\mathbf{x}$ domain. In essence, 4D spatial tensors are extracted for each frequency as discussed by (Kreimer and Sacchi, 2012a; Gao et al., 2015). The current method based on CP decomposition operates in time domain and I stress that was developed as a means to perform SNR enhancement rather than seismic data reconstruction.

One disadvantage of tensor-based algorithms is the potentially high computational cost, which mainly originates from tensor unfolding and folding operations. To avoid this issue, the algorithm Rand-CP-ALS randomly selects a small number of rows from the original large over-determined linear system, and in this way, it can reduce the computational cost significantly and make CP decomposition suitable for seismic noise attenuation problems. One shortcoming of the proposed method is that the random sketching technique used by Rand-CP-ALS are unsuitable for the seismic data reconstruction problem, which formulates seismic data as incomplete tensors with missing values. In this case, the least-squares problems solved for updating factor matrices are no longer highly over-determined. Other tensor decomposition methods which are free of random sampling can be used for seismic data reconstruction.

My work addresses random noise attenuation. However, one must realize that problematic seismic noise is never entirely random. During processing, the coherency of seismic noise can be broken up by sorting the seismic data into a suitable domain. Then random noise attenuation techniques can be implemented. For example, interferences introduced by simultaneous source seismic acquisition are coherent in common shot domain but spatially incoherent in common-receiver and common-offset domains (Berkhout, 2008). Another example is marine swell noise attenuation and cavitation noise attenuation. This type of noise typically has strong amplitude and can contaminate a large number of neighbouring traces of a given shot gather. Swell and cavitation noise can also be made incoherent by sorting the data into common-offset or common-mid point domains (Elboth et al., 2010). These ex-

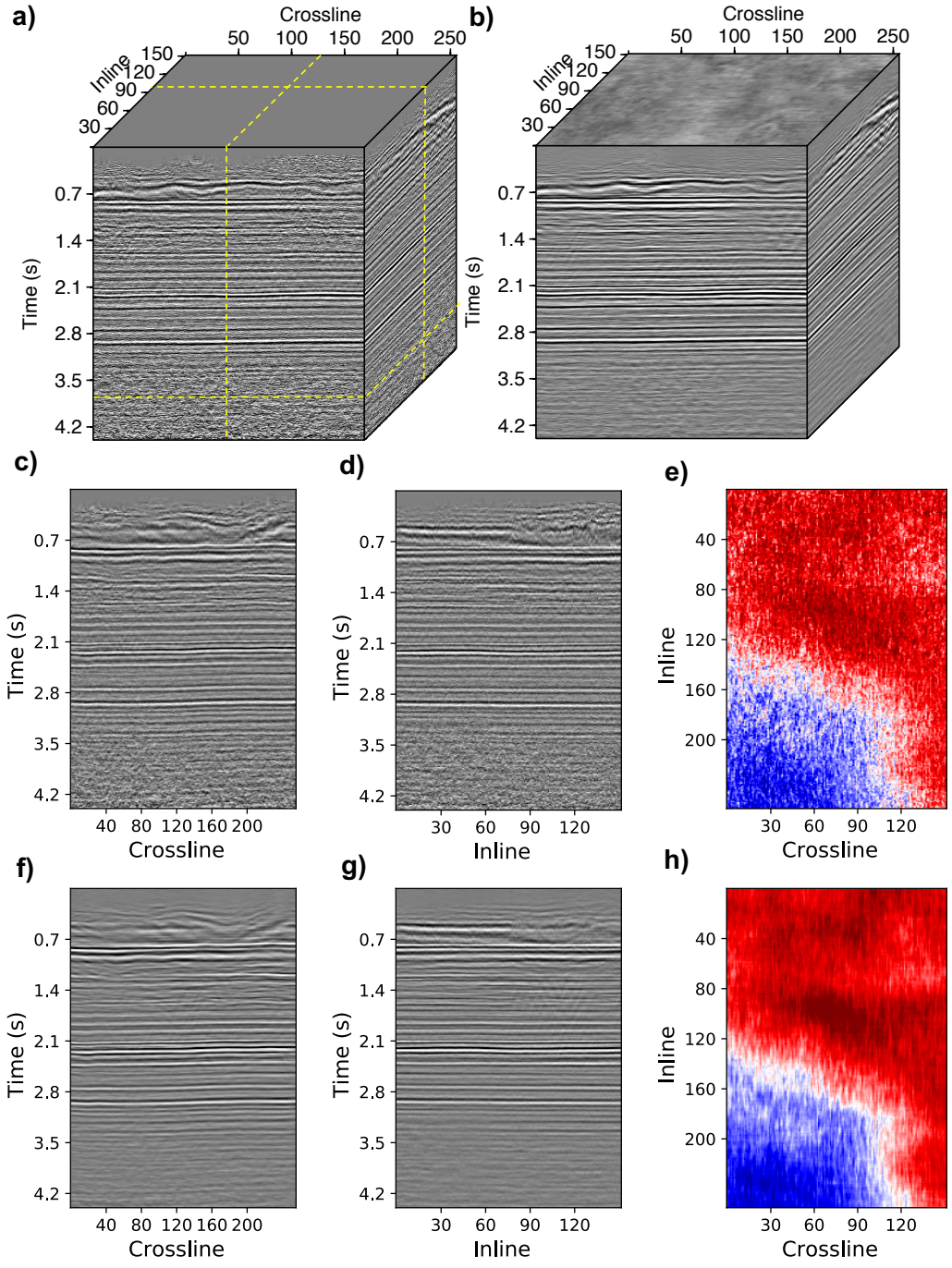


Figure 3.12: 3D PS-wave land volume. (a) The post-stack volume before denoising. (b) Post-stack volume after denoising. (c-e) Inline, crossline sections and time slice of the post-stack PS-wave volume before denoising. (f-h) Inline, crossline sections and time slice of the post-stack PS-wave volume after denoising.

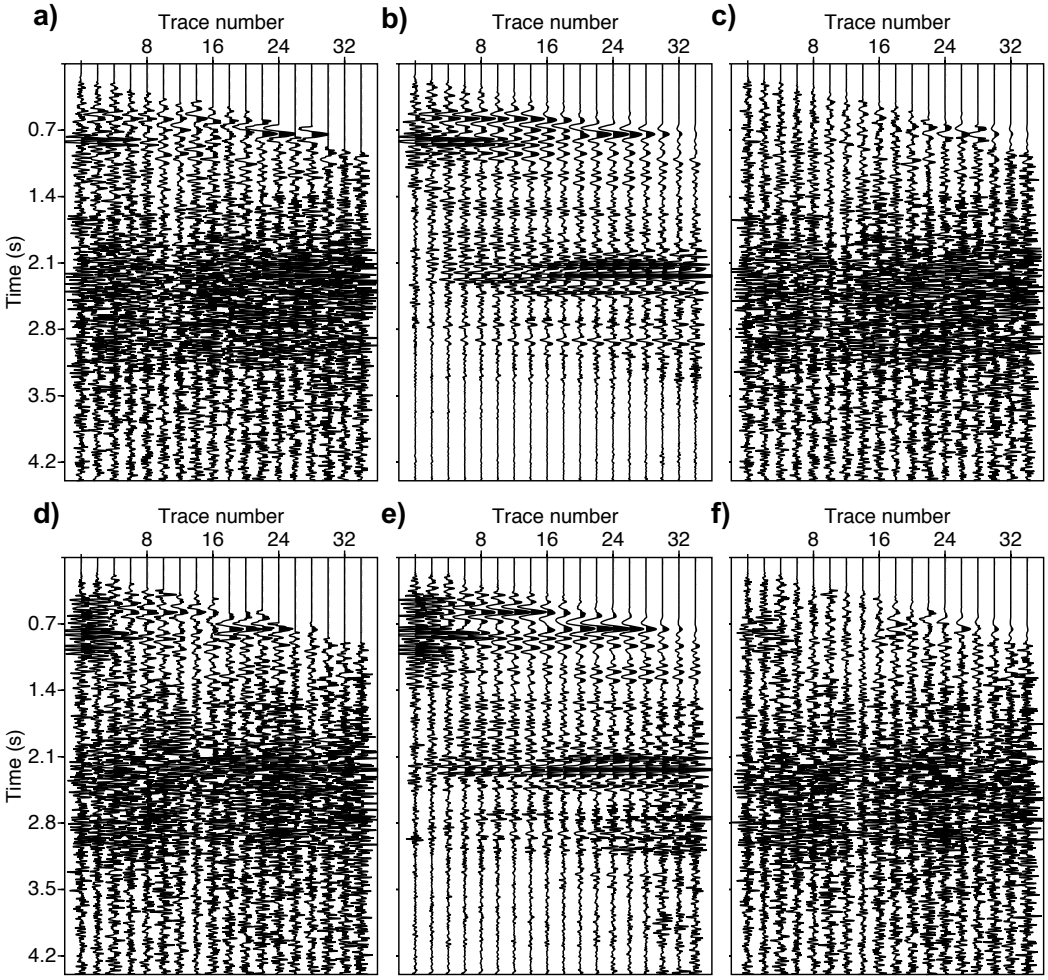


Figure 3.13: CCP gathers of PS-wave data. (a) PS-wave CCP gather before denoising. (b) The same PS-wave CCP gather after denoising. (c) The removed noise. (d-f) The denoising result of another PS-wave CCP gather.

amples provide scenarios where random noise attenuation techniques such as Rand-CP-ALS can be utilized to reduce coherent noise.

## 3.5 Conclusions

In this chapter, I have presented a random noise attenuation method for multi-dimensional seismic data via low-rank CP decomposition, a randomized version of the alternating least squares algorithm is also introduced to speed up the algorithm without sacrificing quality. For random noise problems, I demonstrate the performance of the proposed algorithms via experiments on synthetic and field $3D$ data volumes. My examples pertain 3D post-stack and pre-stack cases. I have also compared the method with traditional algorithms like MSSA and $f$-$\mathbf{x}$ prediction filtering. My research indicates that CP decomposition techniques implemented via randomized sketching algorithms are a viable fast alternative to conventional methods.

# CHAPTER 4

# Regularization of multicomponent seismic data using low-rank tensor completion

## 4.1 Introduction

Mapping post-stack PS-wave data to the PP-wave time domain is a critical step before joint interpretation and inversion. 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. However, one possible limitation of the applicability of registration algorithms on field multicomponent seismic data is the low signal-to-noise-ratio ($SNR$) of the PS-wave data. To deal with this issue, I propose to attenuate the noise of the pre-stack seismic data via 5D interpolation/reconstruction. In our processing flow, we recommend adopting a tensor completion method to reconstruct and enhance pre-stack PP and PS-wave data.

The proposed flow independently reconstructs PP and PS-wave 5D pre-stack volumes via a rank-reduction tensor completion method. Reconstruction methods based on reduced-rank techniques are divided into two categories. In the first category, the multidimensional data are rearranged into a block Hankel matrix, and a rank reduction algorithm is used to improve the $SNR$ and to reconstruct the data. Methods in this category are often named Cadzow reconstruction methods (Trickett et al., 2010b) or multichannel singular spectrum analysis (MSSA) reconstruction (Oropeza and Sacchi, 2011b; Gao et al., 2013). The second category of rank reduction methods encompasses techniques based on dimensionality reduction of multilinear arrays or tensors. Multidimensional seismic data are viewed as multilinear arrays, and dimensionality reduction techniques are directly applied to the multilinear array

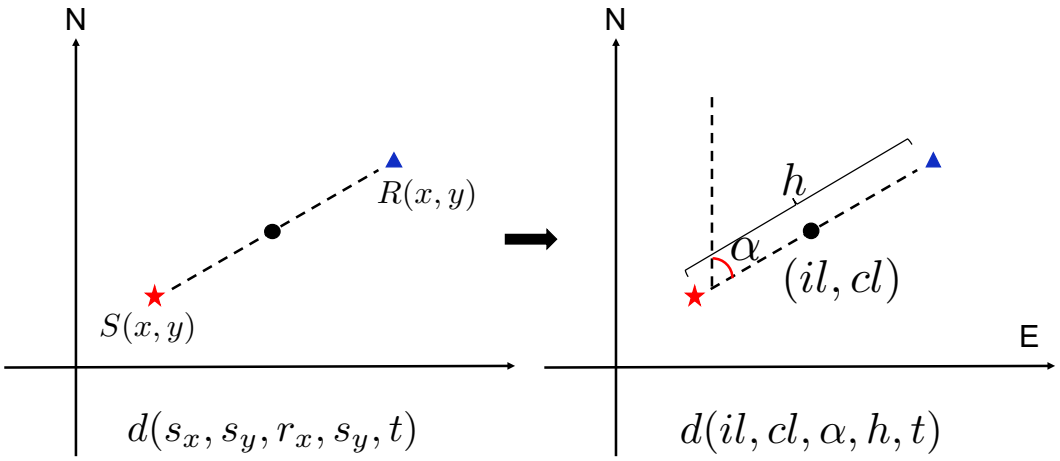


Figure 4.1: Schematic diagram of 3D seismic data acquisition. (a) One trace represented in source-receiver domain. (b) The trace expressed in CMP-offset-azimuth domain.

(Kreimer and Sacchi, 2012b; Kreimer et al., 2013; Gao et al., 2015). Our work builds on recent efforts on simultaneous seismic reconstruction and denoising via tensor completion techniques. Specifically, we have adopted a tensor Higher-Order SVD rank-reduction in conjunction with an imputation algorithm (Kreimer and Sacchi, 2012b) to reconstruct and denoise PP and PS-wave 5D volumes. To demonstrate the effectiveness of the proposed workflow, I test it on a 3D land multicomponent seismic data acquired in Central Alberta, Canada.

## 4.2 Theory

5D seismic data reconstruction can be posed as a low-rank tensor completion problem (Kreimer and Sacchi, 2012b; Kreimer et al., 2013; Gao et al., 2015). Figure 4.1a shows a schematic diagram of a $3D$ seismic data acquisition. The red star denotes a source with coordinates $s_x, s_y$ and blue triangle is a receiver with spatial coordinates $r_x, r_y$. A single trace can be represented by $d(s_x, s_y, r_x, r_y, t)$. After binning and sorting the data in midpoint-azimuth-offset domain, the seismic trace can be represented with the following coordinates: inline ($il$) and crossline number ($cl$) of the $CDP$ gather, azimuth ($\alpha$), offset ($h$) and time ($t$) (Figure 4.1b).

In my case, the input PP data was sorted in common-middle-point ($CMP$) gathers and normal moveout ($NMO$) corrected. Similarly, the PS pre-stack volume was sorted into

common-conversion-point ($CCP$) gathers and then NMO corrected (Chung and Corrigan, 1985; Stewart et al., 2002).

The pre-stack volume in inline-crossline-azimuth-offset-time domain is transformed to the frequency domain and, the tensor completion problem is posed as a fourth-order tensor completion problem for each temporal frequency. The problem is described as

$$\Phi = ||\boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{D}}||_F^2 + \mu||\boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \times_4 \mathbf{A}^{(4)} - \boldsymbol{\mathcal{Z}}||_F^2, \tag{4.1}$$

where $\boldsymbol{\mathcal{S}}$ denotes the sampling operator, $\boldsymbol{\mathcal{Z}}$ is the desired complete low-rank tensor, $\boldsymbol{\mathcal{D}}$ is the observed data, the symbol $\circ$ is element-wise multiplication. The symbol $||\ ||_F^2$ represents Frobenius norm for tensors. The second term is the low-rank constraint. In this case, we represent the low-rank tensor via the Tucker model also called the High-order SVD. The small tensor $\boldsymbol{\mathcal{G}}$ is called the core tensor and $\mathbf{A}^{(i)}$ are orthogonal factor matrices along each dimension. The symbol $\times_n$ represents tensor-matrix multiplication with $n = 1, \cdots, 4$ (Kolda and Bader, 2009a). The minimization of the cost function 4.1 leads to the iterative algorithm described in (Kreimer and Sacchi, 2012b). The algorithm requires one trade-off parameter $\mu$ that controls the reinsertion of data and the size of the core tensor. The size of the core tensor controls the smoothness of the reconstruction.

Algorithm 3 summarizes the HO-SVD technique. The input to this algorithm includes observed data $\boldsymbol{\mathcal{D}}$, sampling operator $\boldsymbol{\mathcal{S}}$, target rank for each mode $r_1, r_2, r_3, r_4$ and an add-back parameter $\alpha$ which controls the amount of reinsertion of the original data into the final solution. The parameter $\alpha$ is related to the trade-off parameter $\mu$ in equation 4.1 given as $\alpha = \frac{\mu}{N+\mu}$. The stopping criteria include two conditions: maximum number of iterations $maxiter$ and tolerance $tol$ of relative change $rel$ between two successive iterations. The relative change used in this chapter is defined as

$$rel = \frac{||\boldsymbol{\mathcal{Z}}^{k+1} - \boldsymbol{\mathcal{Z}}^{k}||_F^2}{||\boldsymbol{\mathcal{Z}}^{k}||_F^2} \tag{4.2}$$

The final output of this algorithm is the reconstructed low-rank tensor $\boldsymbol{\mathcal{Z}}^{k+1}$. The initial guess of factor matrix $\mathbf{A}^{(i)}$ can be obtained as the first $r_i$ left singular vectors of tensor unfolding $\mathbf{D}_{(i)}$. $\boldsymbol{\mathcal{I}}$ is a 4D tensor has same size as observed data. The symbol $\circ$ represents element-wise multiplication.

**Algorithm 3** HO-SVD Algorithm

**Inputs:**

Observations $\boldsymbol{\mathcal{D}}$, Sampling operator $\boldsymbol{\mathcal{S}}$, target rank $r_1, r_2, r_3, r_4$,
tolerance $tol$, imputation parameter $\alpha$.

**Initialize:**

factor matrices $\mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \mathbf{A}^{(4)}$ as orthogonal matrix, $\boldsymbol{\mathcal{Z}}^0 = \boldsymbol{\mathcal{D}}$

**for** k = 0 : maxiter **do**

$\mathbf{U}^{(1)} = \boldsymbol{\mathcal{Z}}^k \times_2 \mathbf{A}^{(2)\mathrm{H}} \times_3 \mathbf{A}^{(3)\mathrm{H}} \times_4 \mathbf{A}^{(4)\mathrm{H}}$

update $\mathbf{A}^{(1)}$ with the $r_1$ left singular vectors of $\mathbf{U}^{(1)}$

$\mathbf{U}^{(2)} = \boldsymbol{\mathcal{Z}}^k \times_1 \mathbf{A}^{(1)\mathrm{H}} \times_3 \mathbf{A}^{(3)\mathrm{H}} \times_4 \mathbf{A}^{(4)\mathrm{H}}$

update $\mathbf{A}^{(2)}$ with the $r_2$ left singular vectors of $\mathbf{U}^{(2)}$

$\mathbf{U}^{(3)} = \boldsymbol{\mathcal{Z}}^k \times_1 \mathbf{A}^{(1)\mathrm{H}} \times_2 \mathbf{A}^{(2)\mathrm{H}} \times_4 \mathbf{A}^{(4)\mathrm{H}}$

update $\mathbf{A}^{(3)}$ with the $r_3$ left singular vectors of $\mathbf{U}^{(3)}$

$\mathbf{U}^{(4)} = \boldsymbol{\mathcal{Z}}^k \times_1 \mathbf{A}^{(1)\mathrm{H}} \times_2 \mathbf{A}^{(2)\mathrm{H}} \times_3 \mathbf{A}^{(3)\mathrm{H}}$

update $\mathbf{A}^{(4)}$ with the $r_4$ left singular vectors of $\mathbf{U}^{(4)}$

compute core tensor $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{Z}}^k \times_1 \mathbf{A}^{(1)\mathrm{H}} \times_2 \mathbf{A}^{(2)\mathrm{H}} \times_3 \mathbf{A}^{(3)\mathrm{H}} \times_4 \mathbf{A}^{(4)\mathrm{H}}$

update low-rank tensor $\boldsymbol{\mathcal{Z}}^{k+1} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \times_4 \mathbf{A}^{(4)}$

data add-back insertion: $\boldsymbol{\mathcal{Z}}^{k+1} = \alpha \boldsymbol{\mathcal{Z}}^{k+1} + (\boldsymbol{\mathcal{I}} - \alpha \boldsymbol{\mathcal{S}}) \circ \boldsymbol{\mathcal{D}}$

**if** rel $\leq$ tol **then**

break

**end if**

**end for**

**return** low-rank tensor $\boldsymbol{\mathcal{Z}}^{k+1}$.

## 4.3 Examples

We investigate the effectiveness of the proposed workflow by implementing the algorithms on a 3D land multicomponent seismic data acquired in Central Alberta, Canada. The data corresponds to the Washout Creek 3D survey (He et al., 2015; Perz et al., 2016). The acquisition geometry is shown in Figure 4.2a, where orange lines denote receiver lines and blue lines are source lines. The data are binned on $20 \times 20$m CDP grid, and a $45° \times 180$m azimuth-offset grid prior to 5D reconstruction.The fold-map of binned PP-wave and PS-wave seismic data are shown in Figure 4.2c,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, I 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.

Seismic events were flattened via the NMO correction. The $5D$ interpolation algorithm is implemented in overlapping sliding windows of size of $16 \times 16 \times 8 \times 15$. The latter corresponds to patches of $16 \times 16$ midpoints, 8 azimuths and 15 offset sectors. For each temporal frequency, the target rank (size of the core tensor) is $(8, 8, 4, 8)$ for the PP-wave data. Figure 4.3 shows the results of the PP-wave interpolation for one CDP gather, the raw PP-wave data with missing traces are shown in Figure 4.3a, and the reconstructed result is shown in Figure 4.3b. Each section is divided into 8 panels corresponding to the azimuth index, for one azimuth segment, the traces are displayed with the increasing of offset from left to right.

Similarly, For the PS-wave data, we use the same binning but a core tensor of size $(5, 5, 4, 5)$. My experiments indicate that smaller core tensors are required for the PS-wave. The reconstruction result for one PS-wave gather is shown in Figure 4.4. The 5D interpolation algorithm preserves the weak reflection amplitudes of the converted waves at near-offset (Figure 4.4b).

Figures 4.5a and b show the raw and reconstructed stacks of PP-wave data. The PS-wave counterparts are shown in Figures 4.5c and d. The $SNR$ of the post-stack cube is improved significantly via 5D interpolation.

## 4.4 Conclusions

5D interpolation formulated as a tensor completion problem is efficient in attenuating random noise in pre-stack land multicomponent seismic data. I have formulated a new registration flow where we first run 5D reconstruction on PP and PS-wave data to produce stacked cubes with enhanced $SNR$. 5D interpolation allows us to denoise pre-stack volumes, and consequently, it yields optimal input data for subsequent processes such as PP/PS registration.

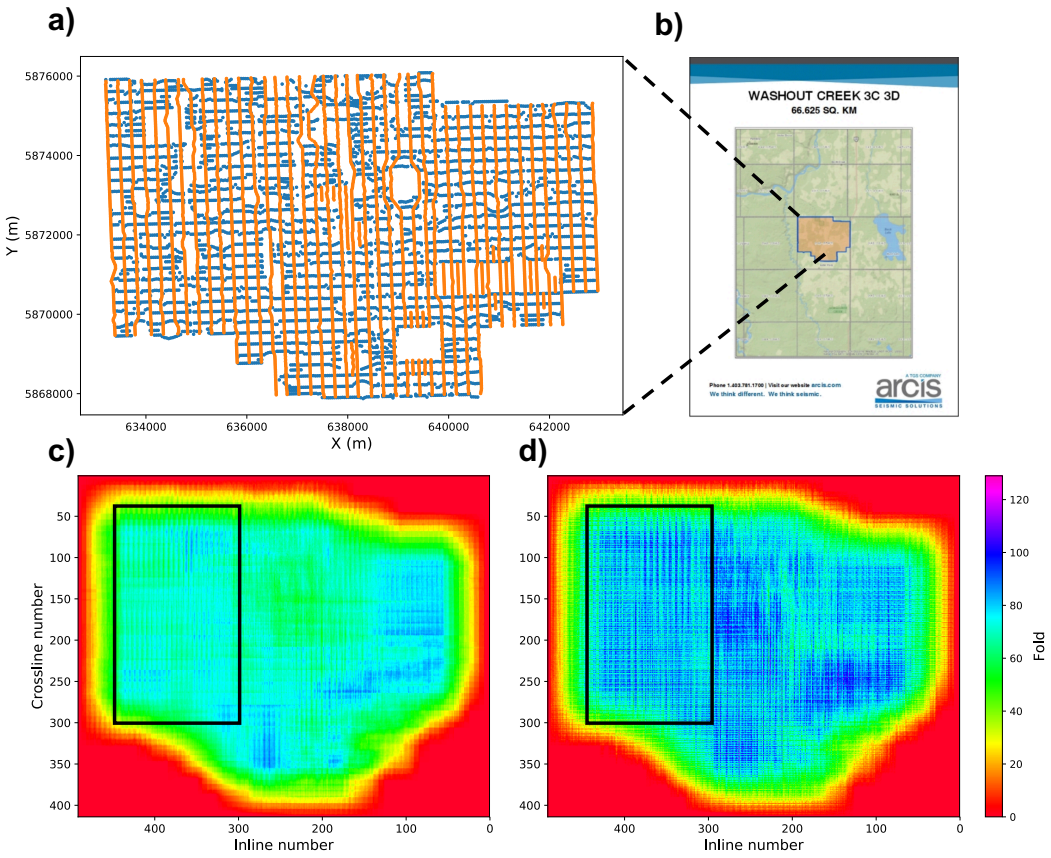


Figure 4.2: Washout-creek seismic data. (a) Acquisition geometry of the 3D seismic data set, orange lines denote receiver lines and blue lines indicate source lines. (b) Location of the seismic survey. (c) The fold-map for PP-wave data, (4) the fold-map for PS-wave data. The black box indicates the area of the data used for testing the proposed algorithm.

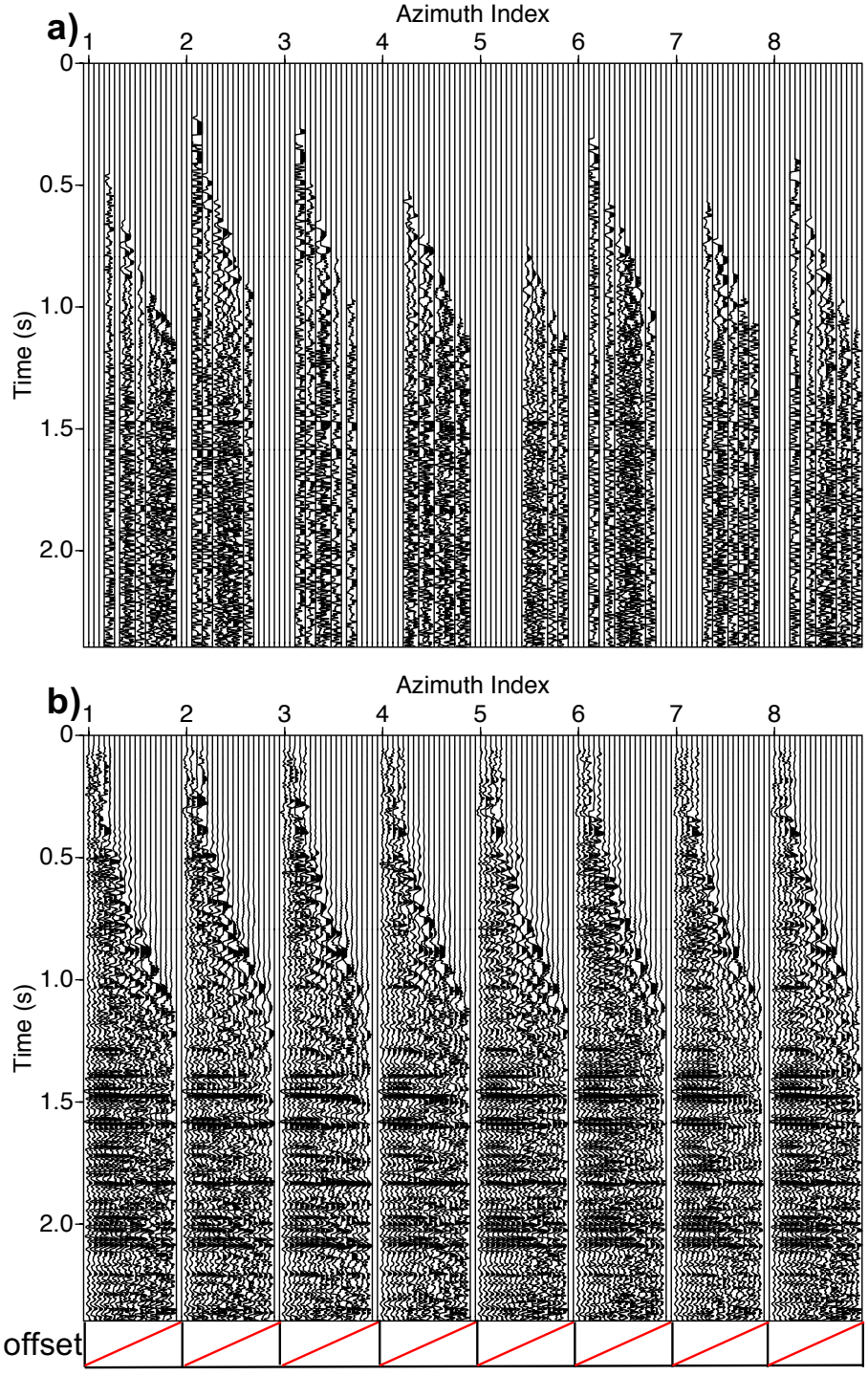


Figure 4.3: One PP-wave gather before (a) and after (b) reconstruction.

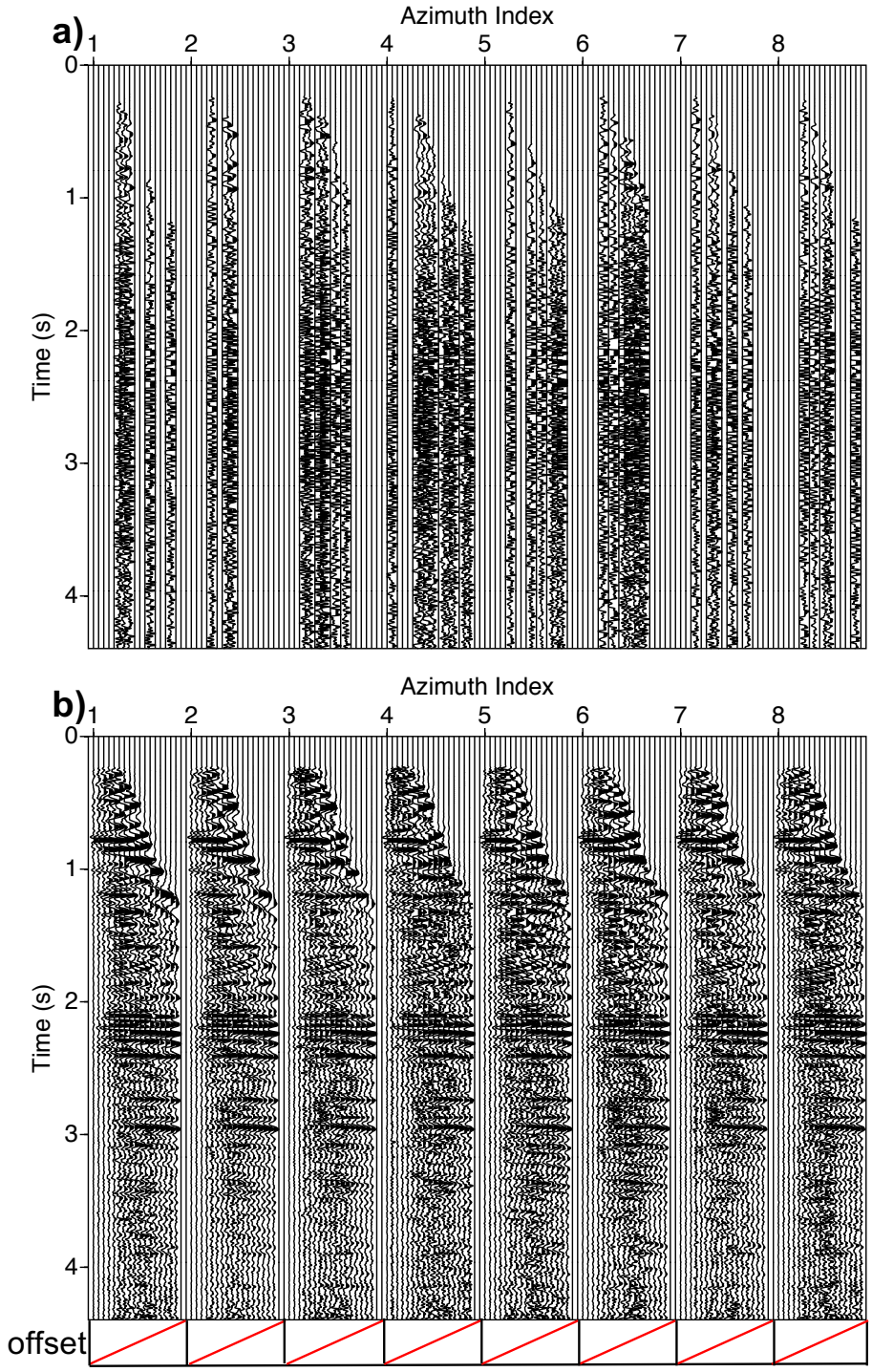


Figure 4.4: One PS-wave gather before (a) and after (b) reconstruction.

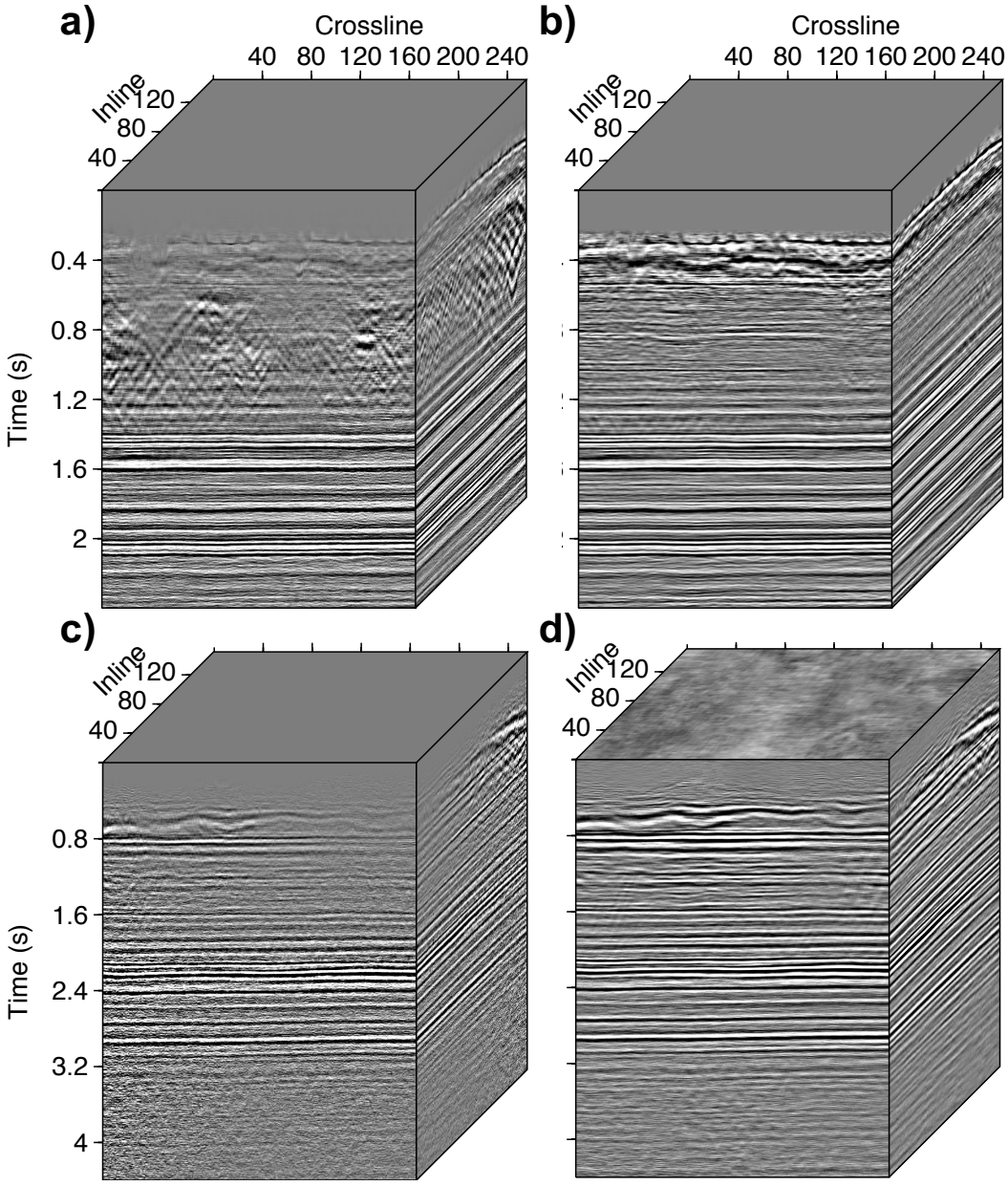


Figure 4.5: (a) and (c) are the PP and PS-wave cubes before interpolation. (b) and (d) cubes after interpolation.

# CHAPTER 5

# Multicomponent seismic data registration[1]

## 5.1 Introduction

Mapping PS-wave data to PP-wave time domain is a critical step before joint interpretation and pre-stack inversion. Multi-component seismic data registration is usually performed with provided $V_p/V_s$ ratio, however, accurate information of velocity ratio is absent in most cases. One can solve the registration problem by minimizing the difference between PP-wave and warped PS-wave data with the constraints of a smooth $V_p/V_s$ ratio field. In order to avoid undesirable foldings and rapid changes in warped PS-wave image, we generally require the warping function to be monotonic with respect to PP-wave travel time and smooth in both time and spatial direction, those requirements are extremely difficult to satisfy in common registration methods.

Multi-component seismic data registration is typically performed by cross-correlation. For instance, Gaiser (1996) introduced a correlation-based method to determine the long wavelength components of $V_p/V_s$ ratio. It can be applied to any pair of PP-wave, PS-wave traces. Fomel and Backus (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 nonlinear problem. All the gradient-based algorithms may easily get trapped in a local minima before converge to global 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. As we mentioned

[1]A version of this chapter is published in Gao. W., and M. D. Sacchi, 2017, Multicomponent seismic data registration by nonlinear optimization, Geophysics, 83, V1-V10.

above, local minima pose a big challenge to optimization-based registration methods. In order to avoid this problem, Yuan et al. (2008) use Simulated Annealing to minimize the normalized cross-correlation between PP-wave image and warped PS-wave image. They also account for frequency-domain differences by time-variant spectrum whitening. Hale (2013) 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.

All the previous mentioned methods try to invert time shifts by minimizing the difference between PP-wave and warped PS-wave data, then estimating the $V_p/V_s$ ratio from the inverted sequence of time shifts. However, rapid changes may occur in the estimated $V_p/V_s$ ratio. In this chapter, I propose a new method which inverts the $V_p/V_s$ ratio directly. To get ride of some of the rapid changes in the inverted $V_p/V_s$ ratio, the cost function is constrained by a smoothing operator. I alleviate the non-linearity problem by decreasing the number of unknowns. The latter is achieved by adopting cubic B-spline functions to model the $V_p/V_s$ ratio. This leads to a non-linear optimization problem that is solved using the Gauss-Newton method (Wright and Nocedal, 1999). Last, by adopting Kronecker products, one is able to extend the single-trace case to multi-channel registration.

## 5.2 Theory

### 5.2.1 Preliminaries

If I denote the PP-wave image by $P(t,x)$ and the converted wave (PS-wave) image by $S(t,x)$, the registration process entails finding the warping function $w(t,x)$ such that

$$P(t,x) - S(w(t,x),x) \approx 0\,. \tag{5.1}$$

In the proposed algorithm, I have preferred to estimate the warping function directly from the envelope of the data rather than the data themselves

$$e_P(t,x) - e_S(w(t,x),x) \approx 0\,. \tag{5.2}$$

In the last equation $e_P(t,x)$ and $e_S(t,x)$ denote the envelope of $P(t,x)$ and $S(t,x)$, respectively. The envelope of the seismic data is given by

$$e_u(t,x) = \sqrt{u^2(t,x) + u_H^2(t,x)}, \tag{5.3}$$

where $u(t,x)$ and $u_H(t,x)$ are the seismic data and its Hilbert transform, respectively. Notice that equation 5.2 is insensitive to any potential polarity mismatch that could exist between PP and PS reflections. The estimation of $w(t,x)$ from $e_P(t,x)$ and $e_S(t,x)$ is posed as a non-linear inverse problem where I minimize the following cost function

$$J = \iint \left(e_P(t,x) - e_S(w(t,x),x)\right)^2 \, dt\, dx + \iint \mu_t \left(\frac{\partial^2 w(t,x)}{\partial t^2}\right)^2 + \mu_x \left(\frac{\partial^2 w(t,x)}{\partial x^2}\right)^2 \, dt\, dx\,. \tag{5.4}$$

The first term of $J$ is the quadratic error function that measures the proximity of warped S-image to the target P-image. The remaining terms penalize non-smooth warping functions. In my algorithm, I have utilized second order quadratic regularization in time and space but bear in mind that other regularization techniques could have also been adopted. The scalars $\mu_t$ and $\mu_x$ are trade-off parameters that control the relative weight between the quadratic error function and the penalty term.

### 5.2.2 Parameterization of the warping function

First, $\gamma(t,x)$ denotes the $V_p/V_s$ ratio. Following Fomel and Backus (2003), for migrated data, the relationship between the $V_p/V_s$ ratio and the warping function is given by

$$\gamma(t,x) = 2\frac{\partial w(t,x)}{\partial t} - 1\,. \tag{5.5}$$

Appendix A provides a simple derivation of equation 5.5.

A discretized version of equation 5.5 is given by

$$\gamma(t_n,x) = 2\frac{w(t_n+\Delta t,x) - w(t_n,x)}{\Delta t} - 1\,, \tag{5.6}$$

where $\Delta t$ is the time sampling interval of the PP-wave data and $t_n = (n-1)\,\Delta t$ is time corresponding to the sample $n$. After a few mathematical derivations I arrive to the following expression

$$w(t_n+\Delta t,x) = \frac{\gamma(t_n,x)+1}{2}\Delta t + w(t_n,x)\,. \tag{5.7}$$

In addition, I will denote $w(t_n,x) = w[n,x]$ and $\gamma(t_n,x) = \gamma[n,x]$. If I assume that the time of the first time sample ($n=1$) is zero for both PP-wave and PS-wave data, then $t_1 = 0$

and $w(0,x) = w[1,x]$ for all $x$. Consequently,

$$\begin{aligned} w[2,x] &= \frac{\gamma[1,x]+1}{2}\Delta t \\ w[3,x] &= \frac{\gamma[2,x]+1}{2}\Delta t + \frac{\gamma[1,x]+1}{2}\Delta t \\ w[n,x] &= \frac{\gamma[n-1,x]+1}{2}\Delta t + \frac{\gamma[n-2,x]+1}{2}\Delta t + \ldots + \frac{\gamma[1,x]+1}{2}\Delta t \end{aligned} \quad . \tag{5.8}$$

Let us first consider the 1D case. In other words, the registration is performed on a pair of PP-wave and PS-wave traces. One can easily reorganize equation 5.8 in matrix form

$$\begin{bmatrix} w[2,x] \\ w[3,x] \\ w[4,x] \\ \vdots \\ w[n,x] \end{bmatrix} = \Delta t \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 1 & 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \frac{\gamma[1,x]+1}{2} \\ \frac{\gamma[2,x]+1}{2} \\ \frac{\gamma[3,x]+1}{2} \\ \vdots \\ \frac{\gamma[n-1,x]+1}{2} \end{bmatrix} , \tag{5.9}$$

furthermore, last equation can be simplified via the following expression

$$\mathbf{w} = \mathbf{A}\mathbf{p}\,, \tag{5.10}$$

where $\mathbf{w}$ is the discrete warping function for a single trace, $m$ is the number of samples of the PP-wave trace and $\mathbf{A}$ indicates the integration matrix. Similarly, $\mathbf{p}$ is a intermediate parameter related to the $V_p/V_s$ ratio via the following expression

$$p(t,x) = \frac{\gamma(t,x)+1}{2}\,.$$

Equation 5.10 is valid for one spatial position $x$. I can generalize equation 5.10 to the multi-channel case. For this purpose, I define time $t$ as the first dimension and $x$ as the second dimension. The 2-D warping function $w(t,x)$ for all traces are discretized and reshaped column-wise into a vector $\mathbf{w}$. Similarly, $p(t,x)$ is reshaped into a vector $\mathbf{p}$. The integration matrix $\mathbf{A}$ is extended to the multi-channel case by adopting the Kronecker product ($\otimes$)

$$\mathbf{w} = (\mathbf{I_n} \otimes \mathbf{A})\mathbf{p} = \mathbf{G}\mathbf{p}\,. \tag{5.11}$$

In the last expression $\mathbf{I}_n$ is an identity matrix of dimension $n \times n$ where $n$ is the number of traces. Similarly, $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{A}$ is a matrix of size $mn \times mn$. The warping function must be monotonic in time. A non-monotonic warping function will create mispositioned warped PS-wave reflections as shown in the following example. Figure 5.1a shows a monotonic warping function. Figure 5.1b illustrates PS-wave, warped PS-wave and PP-wave traces. In

this case, the registration was carried out with the monotonic warping function. The black arrows indicate the correspondence between original PS-wave and warped PS-wave traces. Clearly, PS-wave events are mapped to the correct position in PP-wave time without altering the sequence in which PS-wave reflections have occurred. Conversely, Figure 5.1c portrays a non-monotonic function and Figure 5.1d the associated PS-wave, warped PS-wave and PP-wave traces. It is clear that the PS-wave reflection has been incorrectly mapped to the PP-wave time. Specifically, the PS-wave event at around 0.6 s mapped to a correct position and to two erroneous positions that are marked by red arrows. A non-monotonic warping function will also introduce a physically invalid, non-positive, $V_p/V_s$ ratio.

The warping function obtained via equation 5.11 is monotonic as long as the elements of $\mathbf{p}$ are positive. In addition, I will assume a lower bound for the $V_p/V_s$ ratio, $\gamma(t,x) > \gamma_l$. In my examples I have adopted $\gamma_l = 1.4$. The auxiliary parameter $p(t,x)$ must satisfy the condition $p(t,x) > p_l = (\gamma_l + 1)/2$. After introducing a new variable $u(t,x)$ such that $p(t,x) = p_l + e^{u(t,x)}$, one can adopt an unconstrained non-linear optimization solver in terms of the new unknown $u(t,x)$. This parameterization guarantees that $\gamma > \gamma_l$. The multi-channel warping function can be expressed as follows

$$\mathbf{w} = (\mathbf{I}_n \otimes \mathbf{A})\mathbf{p} = \mathbf{G}(p_l\,\mathbf{1} + e^{\mathbf{u}})\,. \tag{5.12}$$

where $\mathbf{1}$ represents a column vector with ones in all its elements and $e^{\mathbf{u}}$ is a column vector with elements $e^{u_i}$. I also propose to decrease the number of unknowns of the problem by parametrizing $u(t,x)$ in terms of 2D cubic B-splines, which has an advantage for reconstructing continuous smooth model (Hill et al., 2001; Maintz and Viergever, 1998; Klein et al., 2010).

$$u(t,x) = \sum_{j_1=1}^{p_1}\sum_{j_2=1}^{p_2} c_{j_1,j_2} b^{j_1}(t) b^{j_2}(x)\,, \tag{5.13}$$

where $c_{j_1,j_2}$ indicate the nodal points of the cubic B-splines, $p_1$ and $p_2$ indicate the number of nodal point in time and space, respectively. Similarly, $b^{j_1}(t)$ and $b^{j_2}(x)$ are the cubic B-spline basis functions. By re-parameterizing the unknowns in terms of splines, the number of unknowns of my problem are reduced from $m\,n$ to $p_1\,p_2$. The number of nodal points in time is chosen as $p_1 = \lfloor ns/nw \rfloor$ where $ns$ is the number of samples per PS-wave trace and $nw$ is the length of the wavelet of the PS-wave data. The symbol $\lfloor . \rfloor$ is used to indicate the integer part of its argument. The length of the wavelet ($nw$) is estimated from the dominant frequency of the PS-wave data. The number of nodal points in space is computed such that the distance between two consecutive spline nodal points is $2nw$.

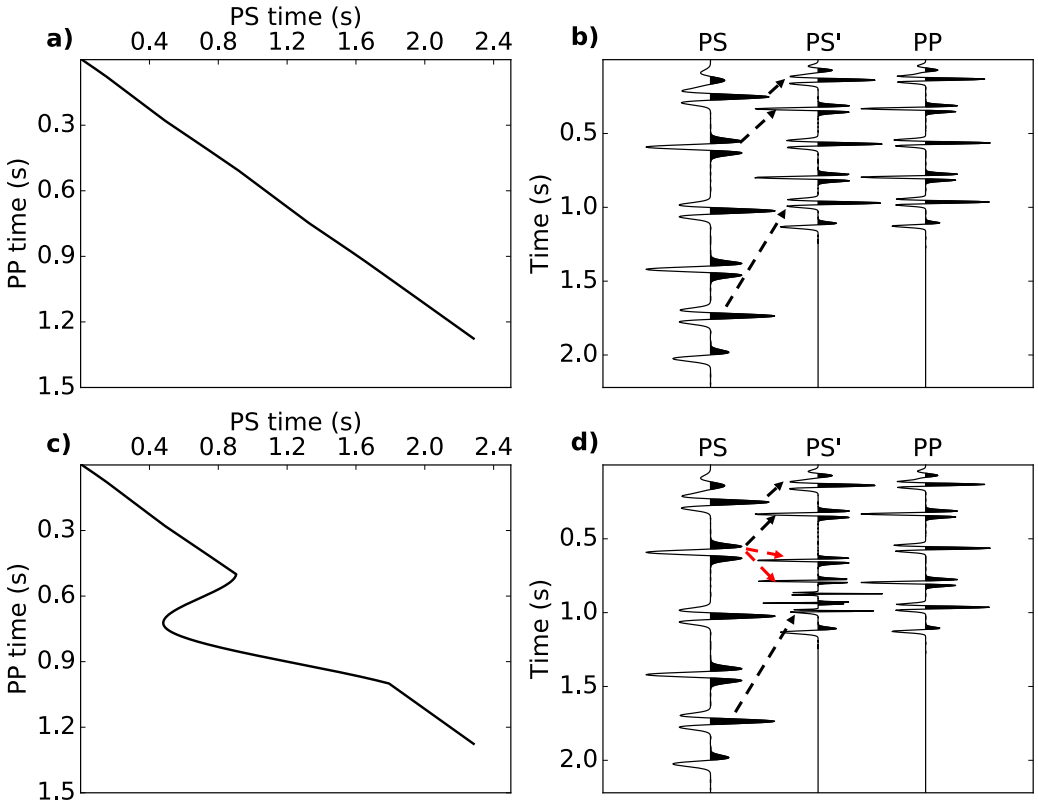


Figure 5.1: a) Monotonic warping function. b) Registration with the monotonic warping function. PS indicates the PS-wave trace and PS' is the warped PS-wave trace after registration, PP denotes the corresponding PP-wave trace, the black arrow indicates the correspondence between PS-wave and warped PS-wave traces. c) Non-monotonic warping function. d) Registration via the non-monotonic warping function. The events marked by red arrows are repeated by the warping process and they appear at wrong positions.

I can also write equation 5.13 in matrix form as follows

$$\mathbf{u} = \mathbf{B}\mathbf{c}\,, \tag{5.14}$$

where $\mathbf{c}$ is a vector of size $(p_1\,p_2 \times 1)$ containing the spline coefficients in vectorized form. The matrix $\mathbf{B}$ of size $(m\,n \times p_1\,p_2)$ is formed by the cubic B-spline basis functions (Klein et al., 2010). Therefore, the final warping function is expressed as follows

$$\mathbf{w} = \quad \mathbf{G}(p_l\,\mathbf{1} + e^{\mathbf{B}\mathbf{c}})\,. \tag{5.15}$$

I also need to compute the derivative of warping function $\mathbf{w}$ with respect to spline coefficients $\mathbf{c}$, a quantity that will be needed by my optimization algorithm,

$$\mathbf{S} = \frac{\partial \mathbf{w}}{\partial \mathbf{c}} = \mathbf{B}^T diag(e^{\mathbf{B}\mathbf{c}})\mathbf{G}^T, \tag{5.16}$$

where the symbol *diag* means to form a diagonal matrix from the vector with elements $e^{[\mathbf{B}\mathbf{c}]_i}$ .

### 5.2.3 Time domain interpolation

In order to evaluate the misfit, I need to evaluate $S(w(t,x),x)$ at discrete time samples that coincide with those of $e_P(t,x)$. This is solved by expressing $e_S(t,x)$ in terms of 1D cubic splines (Philippe et al., 2000)

$$e_S(t,x) = \sum_{j=1}^{m} \alpha_j(x)\phi_j(t)\,, \tag{5.17}$$

I can estimate and save the coefficients of the expansion in terms of splines for each envelope trace. Then, whenever one needs to estimate $e_S(w(t,x),x)$ , I use equation 5.17 to synthesize data by evaluating the sum at $t = w(t,x)$. In my optimization code I will need to compute derivative of the form $\frac{\partial e_S(w(t,x),x)}{\partial w(t,x)}$ which can be easily computed in terms of the spline basis functions via the expression

$$\frac{\partial e_S(w(t,x),x)}{\partial w(t,x)} = \sum_{j=1}^{m} \alpha_j(x)\phi_j^{'}(w(t,x)) \tag{5.18}$$

where $\phi_j^{'}(w(t,x))$ is the derivative of the 1-D spline basis function. When working with multi-channel data the PP-wave image $e_P(t,x)$ is reshaped as a column vector $\mathbf{e}_P$ of length of $mn$. Accordingly, the envelope of the warped PS-wave image $e_S(w(t,x),x)$ is also vectorized and represented by $\mathbf{e}_S(\mathbf{w})$ of length $mn$. The partial derivative of the envelope of the warped

PS-wave image with respect to the warping function can be represented in terms of a diagonal matrix

$$\mathbf{D} = diag(\frac{\partial \mathbf{e}_S(\mathbf{w})}{\partial \mathbf{w}}) \,. \tag{5.19}$$

The argument of $diag(.)$ is a vector with the derivatives of the PS-wave envelope with respect to the warping function at a given time. The vector is conveniently deployed in the main diagonal of $\mathbf{D}$ and it will be used in the optimization code in next subsection.

### 5.2.4 Optimization

The Gauss-Newton method (Wright and Nocedal, 1999) is used to minimize the objective function of the problem. I am now in condition of writing equation 3 in discrete form

$$J(\mathbf{c}) = \|\mathbf{e}_P - \mathbf{e}_S(\mathbf{w})\|_2^2 + \mu_x^2 \|\mathbf{L}_x \mathbf{w}\|_2^2 + \mu_t^2 \|\mathbf{L}_t \mathbf{w}\|_2^2 \,. \tag{5.20}$$

Where $\mathbf{e}_P$ and $\mathbf{e}_S$ are PP-wave and PS-wave multi-channel data in vector form. Similarly, $\mathbf{L}_t$ and $\mathbf{L}_x$ are matrices of second order derivatives in time and space, respectively (Appendix B). To proceed with my algorithm, I first approximate the cost function $J$ in terms of a second order expansion

$$J(\mathbf{c}) = J(\mathbf{c_0}) + \mathbf{g}^T \Delta\mathbf{c} + \Delta\mathbf{c}^T \mathbf{H} \Delta\mathbf{c} \,, \tag{5.21}$$

where is the gradient $\mathbf{g}$ is given by

$$\mathbf{g} = -2\mathbf{S}\mathbf{D}(\mathbf{e}_P - \mathbf{e}_S) + 2\mu_x \mathbf{S}\mathbf{L}_x^T \mathbf{L}_x \mathbf{w} + 2\mu_t \mathbf{S}\mathbf{L}_t^T \mathbf{L}_t \mathbf{w} \,. \tag{5.22}$$

The gradients points in the direction of maximum rate of change of the objective function. The Hessian, after ignoring higher order derivatives of the misfit with respect to model parameters, can be written as follows

$$\mathbf{H} = 2\mathbf{S}\mathbf{D}\mathbf{D}^T\mathbf{S}^T + 2\mu_x \mathbf{S}\mathbf{L}_x^T \mathbf{L}_x \mathbf{S}^T + 2\mu_t \mathbf{S}\mathbf{L}_t^T \mathbf{L}_t \mathbf{S}^T \,. \tag{5.23}$$

It is well-known that the Hessian matrix describes the local curvature of the objective function. The Gauss-Newton step can be expressed as

$$\Delta\mathbf{c} = -\mathbf{H}^{-1}\mathbf{g} \tag{5.24}$$

with update given by

$$\mathbf{c}^{k+1} = \mathbf{c}^k + \alpha\Delta\mathbf{c} \tag{5.25}$$

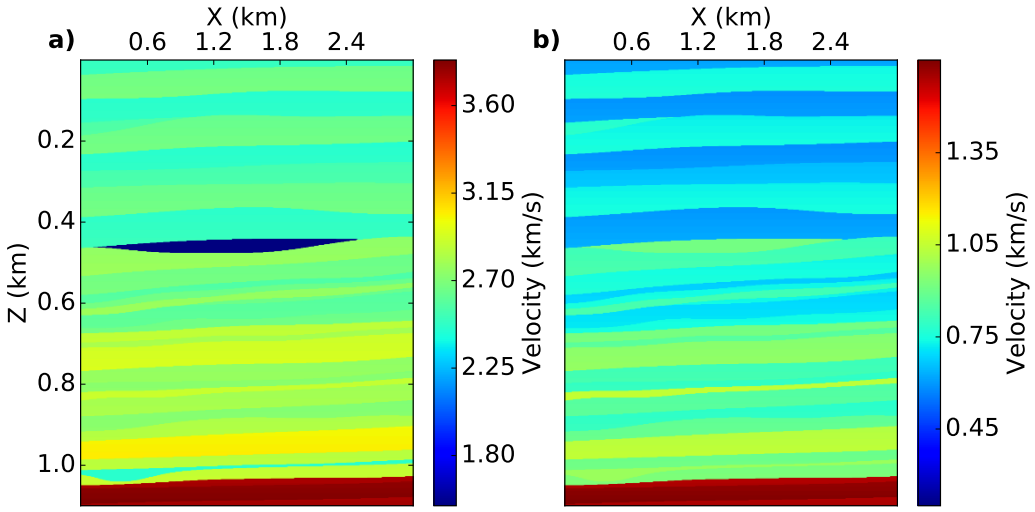


Figure 5.2: a) P-wave velocity model. b) S-wave velocity model. These models were used to generate synthetic data and to test the proposed registration algorithm.

where $\mathbf{c}^{k+1}$ is the vector of B-spline coefficients at the $k+1^{th}$ iteration. Similarly, $\mathbf{c}^{k}$ represents the vector of B-spline coefficients at $k^{th}$ iteration. The scalar $\alpha$ is a step-size that is determined by Armjio's line search method (Nocedal and Wright, 2006).

## 5.3 Examples

### 5.3.1 Synthetic example

To demonstrate the performance of the proposed method, I applied my algorithm to a synthetic and to a real data set. At first, the method is tested on a synthetic data set. Figure 5.2a shows a P-wave velocity model. The P-wave velocity increases from 1.8 km/s to 3.6 km/s with depth. At a depth of about 0.45 km, a rapid P-wave velocity decrease is designed to simulate a gas-bearing layer. The latter is well known for producing sharp P-wave velocity decrease and P-wave energy attenuation (Barkved et al., 2004). On the other hand, the reduction of S-wave velocity caused by gas-bearing layer is much less than on the P-wave velocity. Figure 5.2b shows a S-wave velocity model which increases from 0.55 km/s to 1.6 km/s. I can observe that there is no sharp velocity decrease at the corresponding location of the gas-bearing layer. I point out that this model was extracted from the Marmousi-2 velocity model (Martin et al., 2002).

I used the models in Figure 5.2a-b to compute their associated reflectivity sequences in

time. Then, I computed synthetic seismograms via the convolutional model. Figures 5.3a-b show the post-stack synthetic PP-wave and PS-wave sections, respectively. I have used a zeros-phase Ricker wavelet with a central frequency of 25 Hz to simulate the PP-wave section and the central frequency for PS-wave section is 12.5 Hz. The upper boundary of the gas-bearing layer generates an event with negative reflection coefficients on the PP-wave section. Positive standard polarity and a zero-phase wavelet were applied for generating synthetic seismic data. Under this convention, I expect a trough at the upper boundary of the gas-bearing layer on the PP-wave section. The event is marked by a black arrow that is positioned at about 0.4 s in Figure 5.3a. However, I can see a peak of the corresponding event (marked by a black arrow in Figure 5.3b at a time about 0.9 s) on the PS-wave section. This polarity flip could adversely affect registration algorithms. Another visible discrepancy is the strong reflection from the lower boundary of the gas-bearing layer in the PP-wave section. The latter is missing in the PS-wave data. This extra event in the PP-wave data behaves like a coherent noise signal for the registration algorithm. Both sections were contaminated by band-limited Gaussian noise. Figures 5.3c-d show the sections contaminated with noise. The noise corresponds to an $snr = 2$ where $snr$ is defined as the ratio of the mean square amplitude of the noise-free signal to the variance of the noise. Each PP-wave trace is composed of 258 time samples and the sample interval is $\Delta t = 0.004$ s. Similarly, each PS-wave trace is composed of 528 time samples with same sample interval. There are 300 traces in each section. Finally, I mention that I plot every 10 traces for visualization purposes.

I utilized the proposed non-linear optimization method to solve the registration problem. My algorithm requires an initial $V_p/V_s$ ratio model. The initial model affects the accuracy of the final estimate of the $V_p/V_s$ ratio as the convergence to the global minimum cannot be guaranteed. To alleviate the problem, I decrease the number of model parameters by representing them with cubic B-splines and I ensure that the inverted $V_p/V_s$ ratio honors a predefined lower bound. By taking this special parameterization, the proposed algorithm relaxes the requirement of an accurate initial guess. To explore the sensitivity to the initial $V_p/V_s$ model, a linearly decreasing $V_p/V_s$ model is used as the initial guess for this synthetic data example. The initial 1D $V_p/V_s$ ratio model, which is far away from the true model, is shown in Figure 5.4a. Figure 5.4b is the smoothed true velocity ratio model. A normalized Gaussian filter with length of 40 samples was used to smooth the true velocity ratio model. I choose to show the smoothed version of the $V_p/V_s$ ratio model because only its long-wavelength components can be recovered by the proposed registration procedure. I have tested the algorithm under different scenarios to examine the influence of the number of spline nodal points $(p_1, p_2)$ and the trade-off parameter on the inverted $V_p/V_s$ ratio. For this purpose, I choose $\mu = \mu_x = \mu_t$. I also define a scalar $R$ as the ratio of the data misfit to the regularization term with the given initial guess. The value of $\mu$ is adjusted to yield a user defined value of $R$. This ad-hoc procedure has enabled us to explore a user-friendly

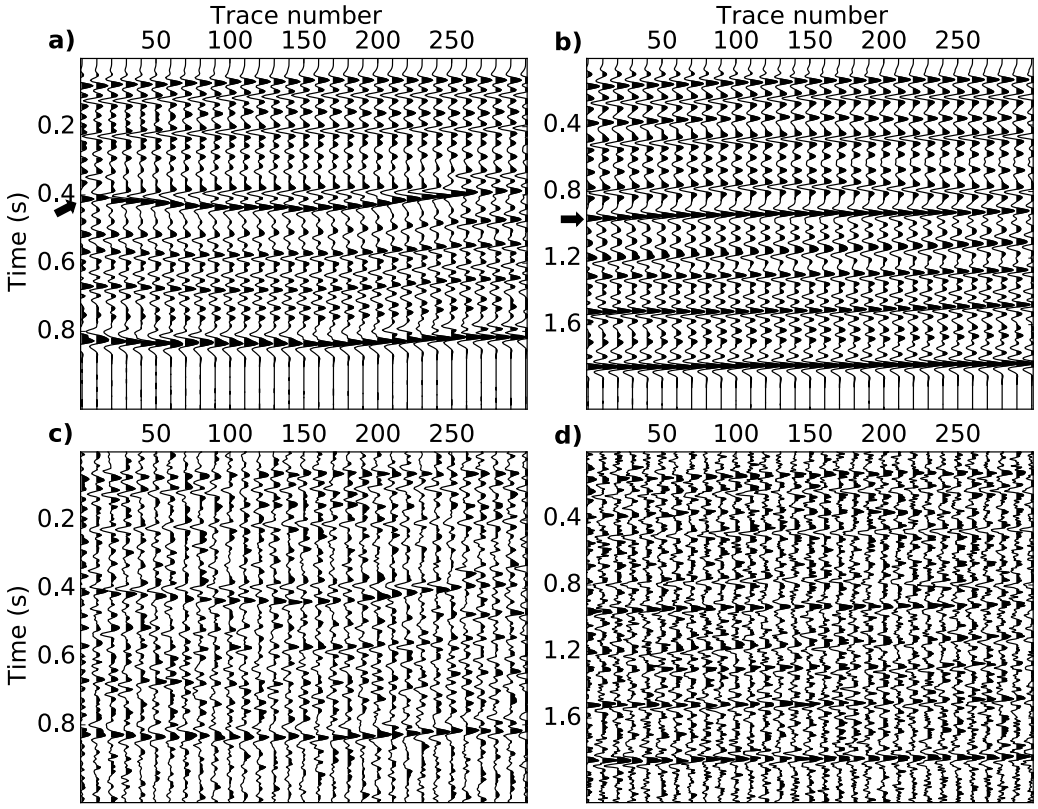


Figure 5.3: a) PP-wave seismic data associated with the velocity model in Figure 5.2a. b) PS-wave seismic data associated with the velocity model in Figure 5.2b. c) and d) PP-wave and PS-wave sections after contamination with random noise, respectively.

procedure for trade-off parameter selection. I have run examples under different scenarios by varying the values of the parameters $(p_1, p_2, R)$. Figure 5.4c shows the estimated smooth $V_p/V_s$ model with parameters $(20, 10, 100)$. Figure 5.4d shows the $V_p/V_s$ model inverted with parameters $(20, 10, 10)$. Comparing Figure 5.4c with Figure 5.4d, the high-frequency perturbations in Figure 5.4c are suppressed by the smoothing constraint and the low $V_p/V_s$ ratio zone at about 0.4 s is correctly retrieved. The inaccurate low $V_p/V_s$ ratio bands at the top of Figures 5.4c-d are caused by the lack of strong reflections near the surface.

To investigate the influence of the number of spline nodal points on the estimated $V_p/V_s$ ratio, I also explore parameters $(10, 5, 100)$ and $(10, 5, 10)$ in Figures 5.4e and 5.4f, respectively. The low $V_p/V_s$ ratio zone is barely identifiable from Figures 5.4e and 5.4f. My results show that decreasing $p_1, p_2$ may lead to overly-smooth solutions.

I also compared results obtained with parameters $(20, 10, 10)$ and $(10, 5, 10)$ in Figure 5.5. I denoted `test1` the solutions with parameters $(20, 10, 10)$. Similarly, `test2` designates solutions with parameters $(10, 5, 10)$. Figure 5.5a shows the initial, the true and inverted models for `test1` and `test2`. The results correspond to the central column in the $V_p/V_s$ ratio model, I can see that the inverted model expressed with large number of spline nodal points is capable of capture more details of the true $V_p/V_s$ model. Figure 5.5b shows the convergence curve for `test1` and `test2`. The horizontal axis represents iteration number $k$ and the vertical axis expresses the normalized data misfit at iteration $k$, $J^{[k]}/J^{[1]}$. The nonlinear optimization algorithm stops when $\mid J^{[k]} - J^{[k-1]} \mid / \mid J^{[1]} \mid < \epsilon$ with $\epsilon = 0.02$ or when the number of iterations reaches a user supplied maximum of 50 iterations. As anticipated, increasing the number of nodal points improves data fitting. The value of the warping function for the central trace in the section is shown in Figure 5.5c for both `test1` and `test2`. The differences between the estimated and true warping functions for both models are minor. Differences in msecs for `test1` and `test2` are shown in Figure 5.5d.

My exploratory analysis of trade-off parameters leads to adopting the set $(10, 5, 10)$ for the remaining test with synthetic data. The $V_p/V_s$ ratio and the warping function are determined from the envelopes of the PP-wave and PS-wave sections. However, the warping function is finally applied directly to the PS-wave section. For completeness, I compared warping functions estimated directly from the data and the envelope. Figure 5.6a shows the PP-wave synthetic section. Figure 5.6b shows the warped PS-wave section with warping function estimated from the envelope. Similarly, Figure 5.6c shows the warped PS-wave section with warping function estimated from seismic data. All the events are correctly shifted to the right location in PP-wave time domain except for the reflections associated with the upper boundary of the gas-bearing layer. These reflections are indicated via black arrows at about 0.4 s. The warping function estimated from the seismic data improperly ties the event associated with the upper boundary in the PS-wave section to the event associated

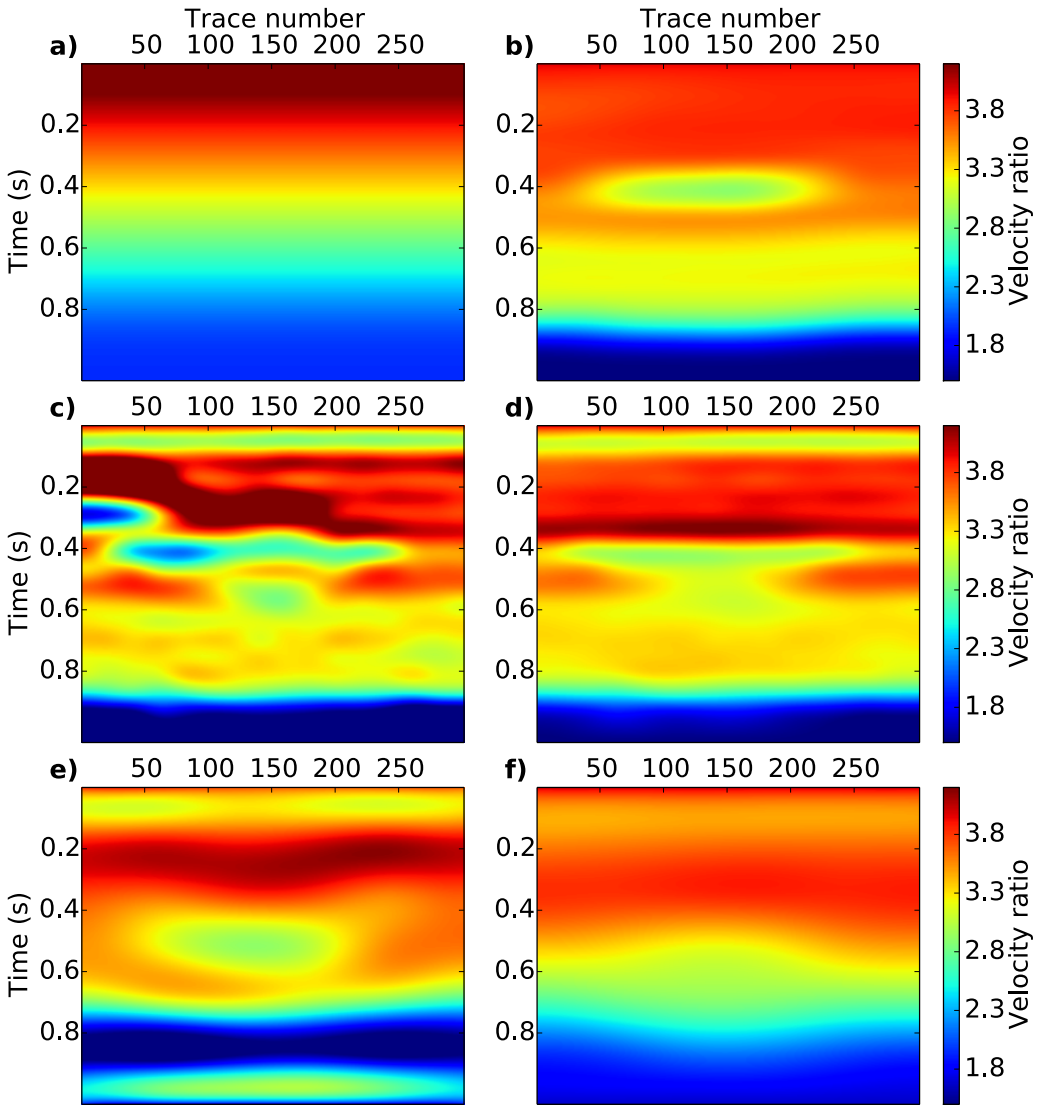


Figure 5.4: a) The initial $V_p/V_s$ ratio adopted by my registration algorithm. b) The smoothed synthetic $V_p/V_s$ ratio. c) The estimated $V_p/V_s$ ratio with parameters $(p_1, p_2, R) = (20, 10, 100)$. d) The estimated $V_p/V_s$ with parameters $(20, 10, 10)$. e) The estimated $V_p/V_s$ with parameters $(10, 5, 100)$. f) The estimated $V_p/V_s$ with parameters $(10, 5, 10)$. The number of nodal points in time and space is given by $p_1$ and $p_2$, respectively. The scalar $R$ is the ratio of the misfit function to the regularization term in the first iteration. The trade-off parameter $\mu$ is adjusted to yield a user-defined $R$.

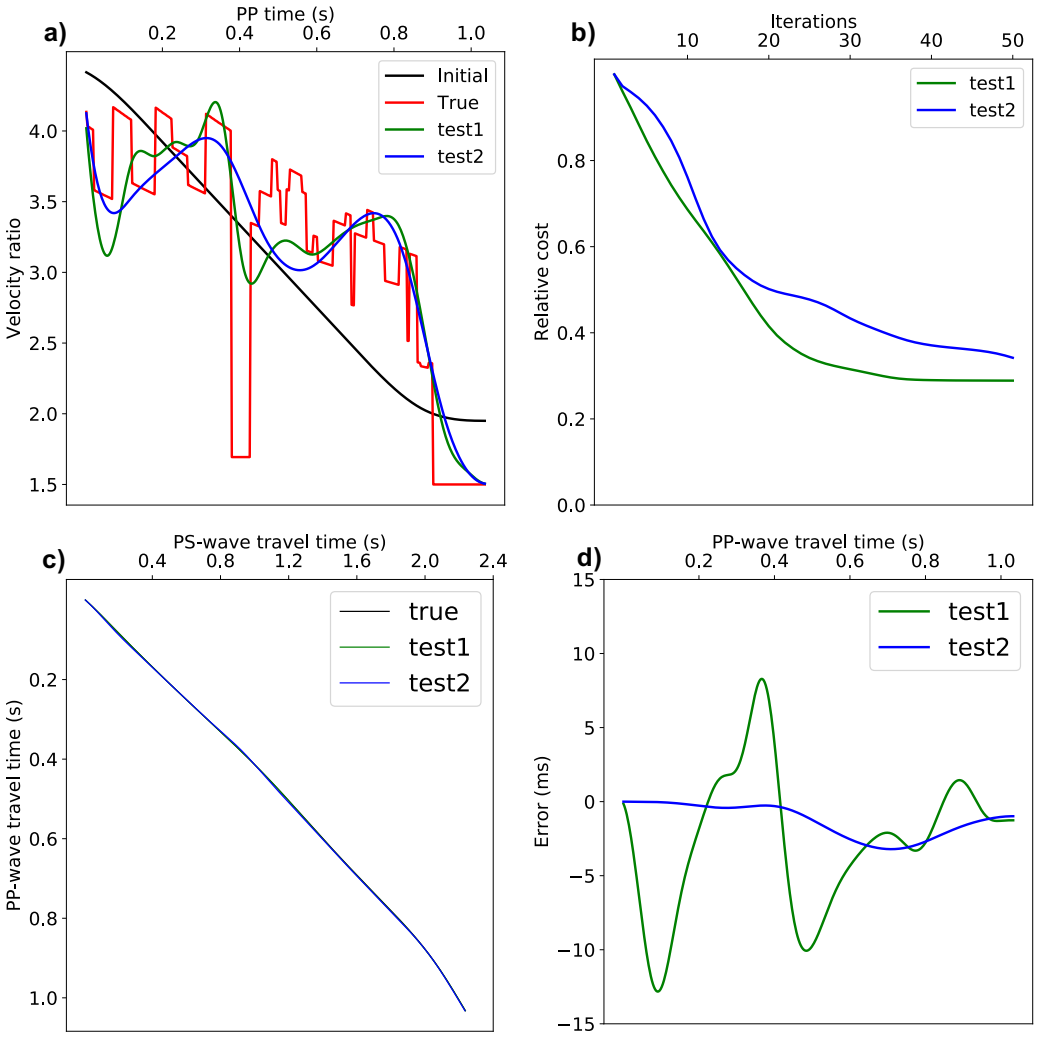


Figure 5.5: a) The central columns of $V_p/V_s$ ratio model, the red line indicates the true $V_p/V_s$ ratio. The black line indicates the initial $V_p/V_s$ ratio. The green line is the $V_p/V_s$ ratio for `test1` which corresponds to using parameters $(p_1, p_2, R) = (20, 10, 10)$. The blue line is the $V_p/V_s$ ratio for `test2` estimated with parameters $(10, 5, 10)$. b) The relative misfit versus iterations. c) The true and estimated warping functions for `test1` and `test2`. d) Warping function error associated to `test1` and `test2`.

with the lower boundary in PP-wave section. I also provided a trace-by-trace comparison in Figure 5.6d where I portray registered traces for traces $5, 100, 173$ and 280. The figure compares traces registered directly from the seismic data versus traces registered from the envelopes. The black circles in Figure 5.6d highlight differences between envelope based and seismic data based registration. The warping function computed using the envelope yields a registered PS-wave reflection with correct polarity.

Figure 5.7 shows the frequency spectrum of the PP-wave data, the PS-wave data and the warped PS-wave data, respectively. The high-frequency components of the PS-wave data are boosted by warping the PS-wave data set to PP-wave time. The PS-wave traces are squeezed when the PS-wave time is transformed to PP-wave time by the registration process. Therefore, the high-frequency components of the PS-wave data are boosted. Ursenbach et al. (2013) pointed out that the latter does not necessarily mean that the resolution of the warped PS-wave data has been improved. The resolution is determined by the wavelength of the wavelet of the seismic data. To improve the resolution one needs to incorporate a deconvolution process that truly expands the bandwidth of the PS-wave data as proposed by Gao and Sacchi (2016).

### 5.3.2 Field 2D data example

The proposed method was also tested on a field data set from East China. The data are shown in Figure 5.8. The PP-wave, warped PS-wave and raw PS-wave data are shown in Figure 5.8a, b, c, respectively. The data set consists of 400 traces. Each PP-wave trace has 600 samples. Similarly, the PS-wave data consist of traces of 1200 samples. The time sample interval is 0.004 s. In this example, I use $p_1 = 30$ and $p_2 = 15$ B-splines nodes to represent the $V_p/V_s$ ratio. The initial model of $V_p/V_s$ is obtained from an initial interval PP-wave and PS-wave velocity model. The trade-off parameter $\mu$ was computed such that the misfit to regularization ratio in the first iteration is $R = 10$. Preprocessing was performed to normalize the amplitude of PP-wave and PS-wave data prior to registration. Preprocessing involves an amplitude scaling method that includes four steps (Margrave, 2007): 1) I first compute the envelope of each trace, 2) the envelope is convolved with a Gaussian smoothing filter of length 15 points, 3) the input traces are divided by the smoothed envelope and 4) the output traces are normalized by dividing each trace by its maximum absolute value.

I also show a small portion of the PP-wave and the warped PS-wave images in Figures 5.9a and 5.9b. These windows are indicated via the black boxes in Figures 5.8a and 5.8b. The dashed lines in Figures 5.9a and 5.9b indicate the joint guided geological interpretations of PP-wave and PS-wave data. Evaluating seismic registration can be problematic given the differences in the spectral content of PP-wave and PS-wave data. I apply a band-pass filter

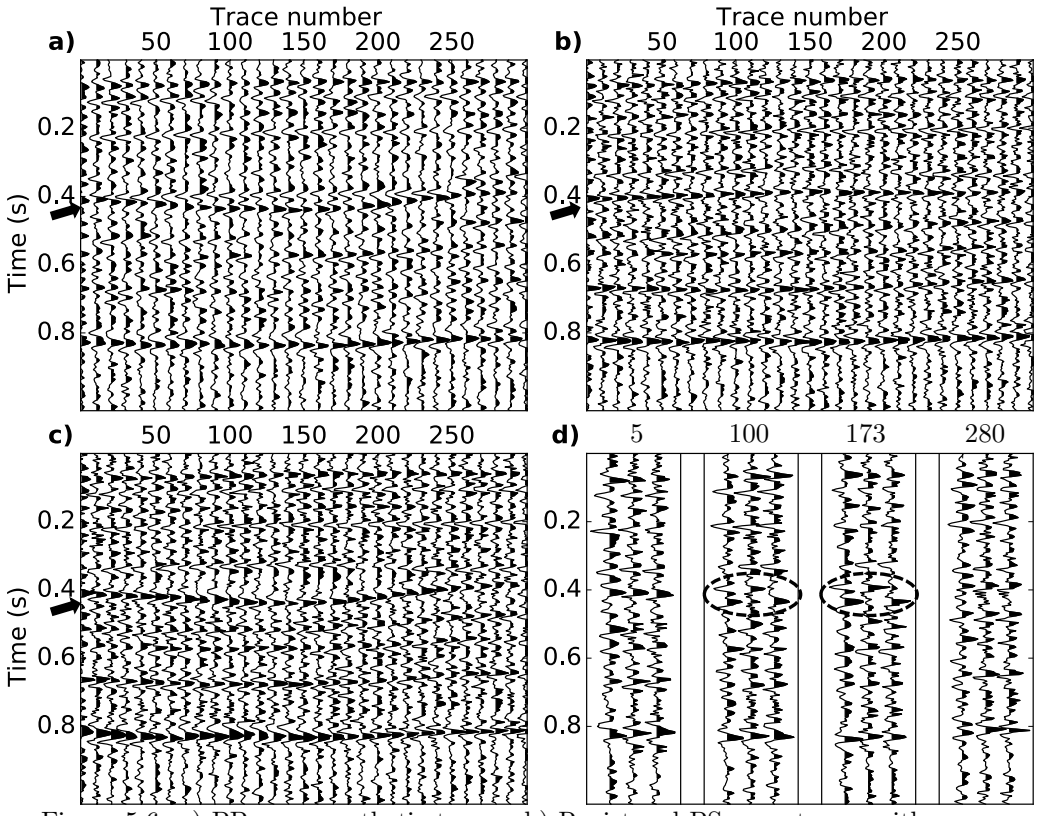


Figure 5.6: a) PP-wave synthetic traces. b) Registered PS-wave traces with warping function estimated from the envelope of the seismic data. c) Registered PS-wave traces obtained with the warping function estimated from the seismic data. The black arrows in Figures 5.6a,b,c mark the reflectors associated with the upper boundary of the gas-bearing layer. d) Trace-by-trace comparison between the PP-wave and the warped PS-wave data. Traces at four locations ($5, 100, 173$ and 280) are extracted for comparison. Each group (separated by two empty traces) consist of 3 traces, from left to right, they are PP-wave data, registered PS-wave data with warping function estimated from the envelope and registered PS-wave data with warping function estimated from seismic data directly. The black circle in Figure 5.6d mark the difference between envelope based and seismic data based registration.

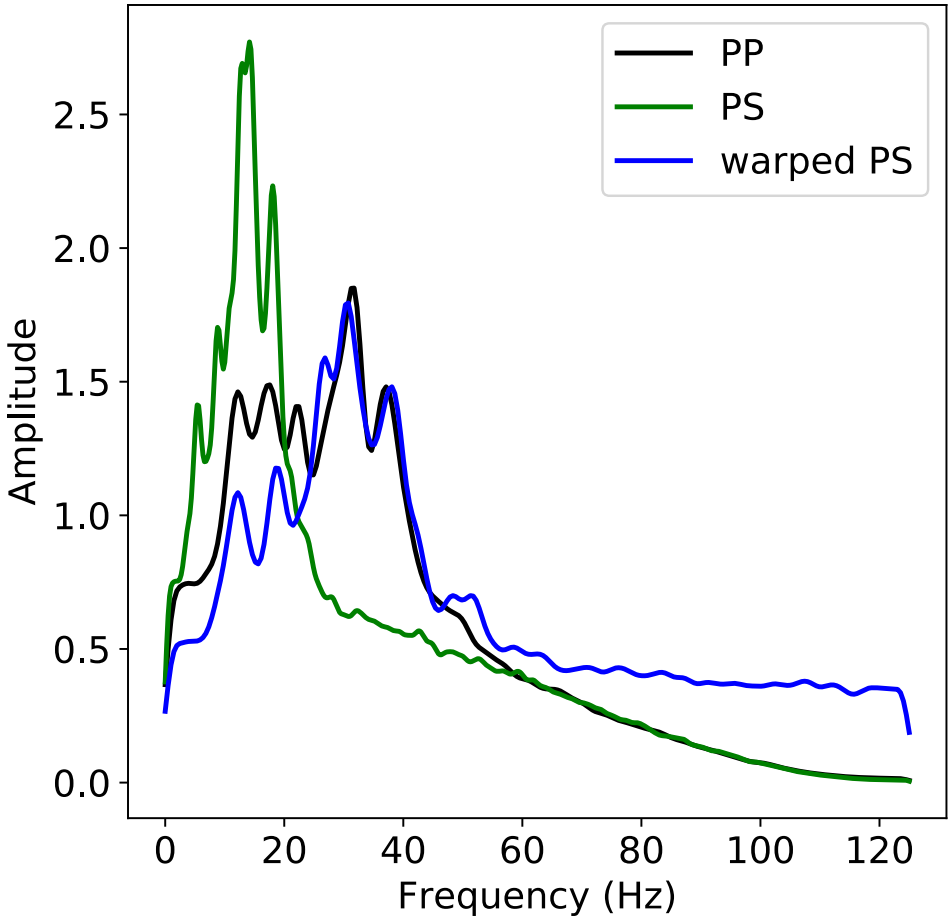


Figure 5.7: Average normalized power spectral density (PSD) of the synthetic data used to test the proposed registration algorithm. The black line is the PSD of the PP-wave section (Figure 5.6a). The green line is the PSD of the PS-wave section (Figure 5.3d). The blue line is the PSD of the PS-wave data after registration (Figure 5.6b).

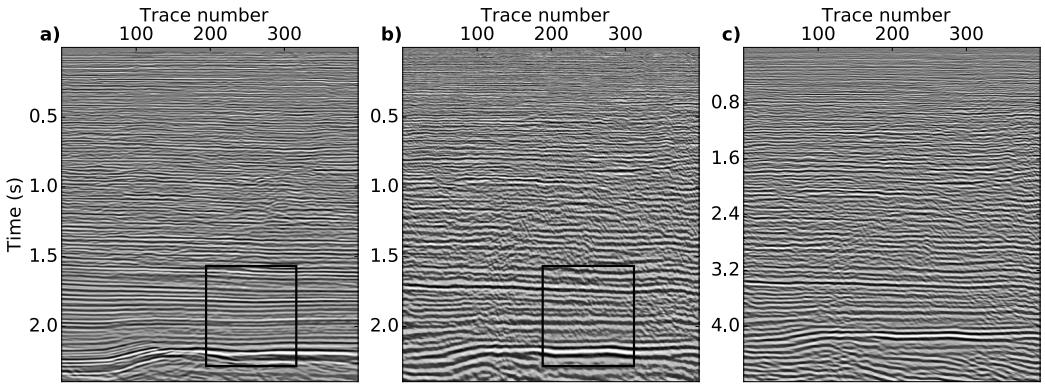


Figure 5.8: Field data example. a) PP-wave section. b) PS-wave section after registration. c) The original PS-wave section before registration.

to the PP-wave data to mimic the bandwidth of the warped PS-wave data. The filtered PP-wave data is shown in Figure 5.9c. The spectra of the PP-wave, warped PS-wave, band-pass filtered PP-wave data are included in Figure 5.9e. Finally, Figure 5.9d shows the difference between band-pass filtered PP-wave data and warped PS-wave data. All images were plotted using the same scale. To further examine the registration in detail, I provided a trace-by-trace comparison between band-pass filtered PP-wave and warped PS-wave traces in Figure 5.9f. These traces were extracted from 3 locations with trace number $230, 260$ and 290. The PP-wave trace and warped PS-wave trace from an identical location were plotted side-by-side, traces for different location are separated by one empty trace.

Figure 5.10a shows the initial $V_p/V_s$ ratio model. Figure 5.10b shows the estimated final smooth $V_p/V_s$ model via the proposed non-linear optimization method.

### 5.3.3 Field 3D data example

I also extended the proposed algorithm to a 3D multicomponent seismic data acquired in central Alberta basin. Figure 5.11a-b show the PP-wave and registered PS-wave cube respectively. After registration, the PS-wave events are shifted to the location of their corresponding PP-wave events. To examine the result of the registration, I will compare the time slice, inline sections PP-wave and warped PS-wave cube.

Figure 5.12 shows the time slices of PP-wave data and warped PS-wave data at time about 1.9 s, the location of these two time slices are indicated by the yellow dash lines in Figure 5.11. The main pattern of the amplitude of PP-wave time slice is very similar to the one of the warped PS-wave data.

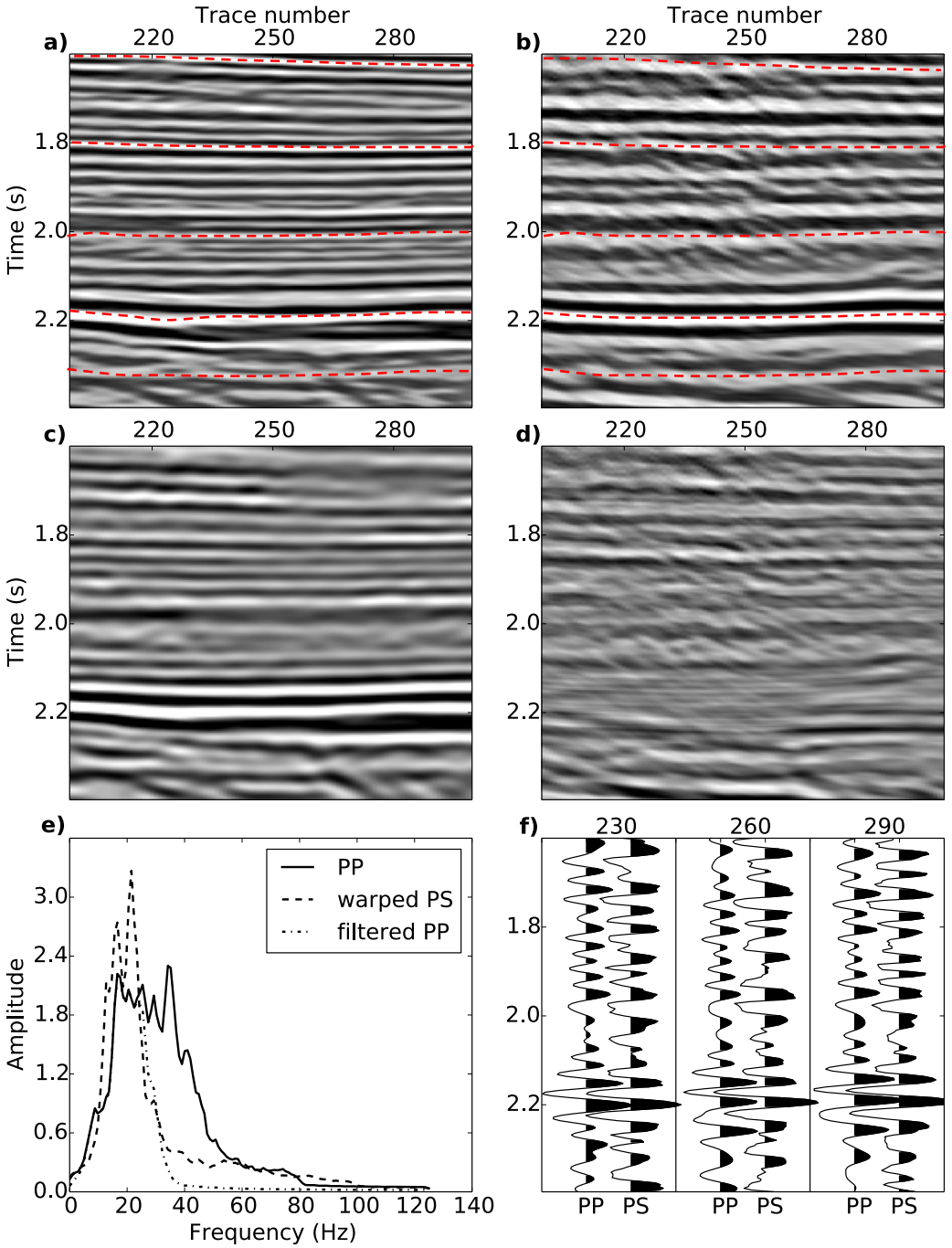


Figure 5.9: Window of the data displayed in Figures 5.8a and b. a) The PP-wave seismic section. b) The warped PS-wave seismic section. The dashed lines in a,b show my interpretations of the PP-wave and warped PS-wave data. c) The band-pass filtered PP-wave section with the same frequency bandwidth as the warped PS-wave data. d) The difference between the warped PS-wave data and the band-pass filtered PP-wave data. All images were displayed with the same scale. e) Average spectrum of the PP-wave data (solid line), the warped PS-wave (dashed line) and the band-pass filtered PP-wave (dotted line) data. f) Trace-by-trace comparison between the band-pass filtered PP-wave and the warped PS-wave data.

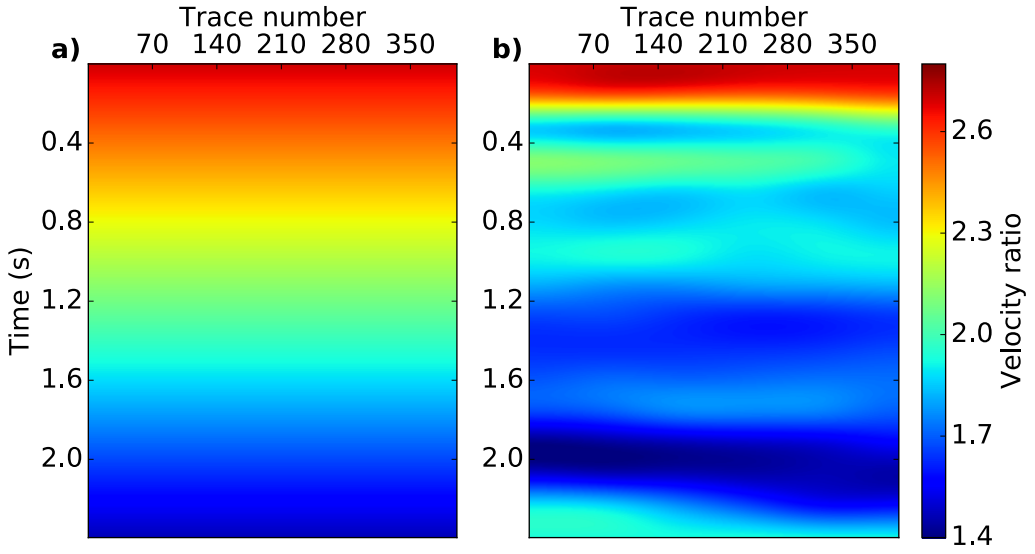


Figure 5.10: Field data example. a) The initial $V_p/V_s$ ratio. b) The estimated $V_p/V_s$ ratio obtained via the proposed registration technique.

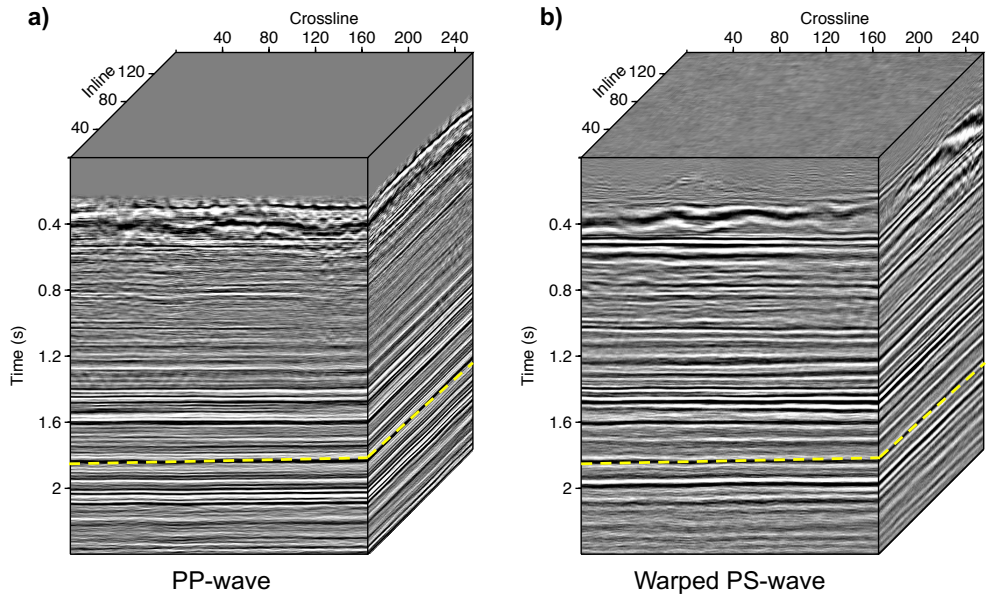


Figure 5.11: 3D seismic cube. a) PP-wave data. b) Registered PS-wave cube.

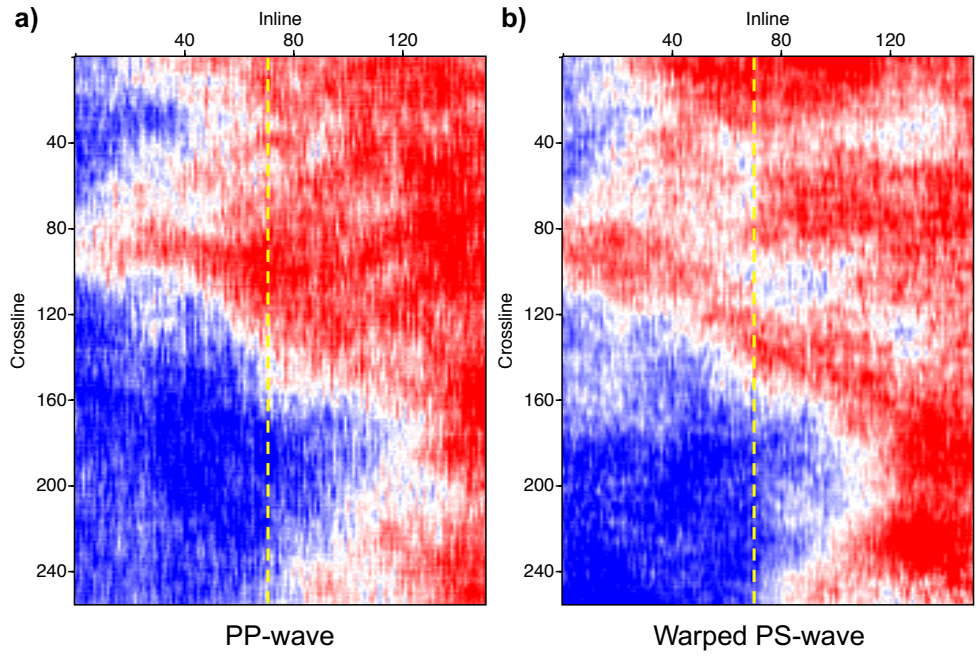


Figure 5.12: a) Time slice of PP-wave data. b) Time slice of registered PS-wave data.

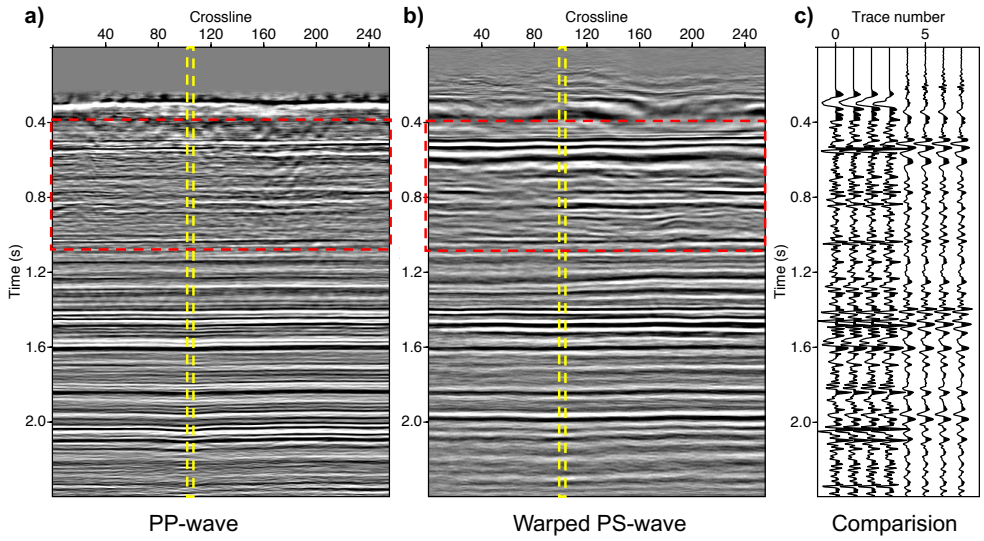


Figure 5.13: a) Inline section of PP-wave data. b) Inline section of registered PS-wave data. c) wiggle plot of PP-wave and registered PS-wave traces.

Figure 5.13a,b show 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. To examine the details of registration, I plot 4 traces of PP-wave data and 4 corresponding traces of PS-wave data side-by-side in Figure 5.13c. The location of the 4 traces are indicated by the yellow dash lines in Figure 5.13a,b.

## 5.4 Conclusions

I have presented a new algorithm for multicomponent seismic data registration. The algorithm inverts for a smooth $V_p/V_s$ ratio. The velocity ratio is represented by splines. Parametrization via splines provides two benefits: the number of unknowns is reduced and the smoothness of the estimated $V_p/V_s$ ratio is guaranteed. The proposed algorithm for seismic data registration entails minimizing a non-quadratic cost function via the Gauss-Newton method. The multi-channel registration case adopts a Gauss-Newton algorithm with sparse matrix-times-vector products, making it computationally efficient and applicable for the registration of a large number of traces. The proposed algorithm was tested via synthetic and field data sets. The synthetic data example demonstrates that this registration method is capable of overcoming polarity flips between PP-wave and PS-wave data. The algorithm is also capable of recovering the long wave-length components of $V_p/V_s$ ratio model. Finally,

the applicability of the proposed method was investigated via a field data example. My field test shows PP-wave and warped PS-wave sections that agree with my preliminary geological interpretations of the sections. A smooth $V_p/V_s$ ratio model was obtained as an important by-product of multi-component seismic data registration.

# CHAPTER 6

# Kronecker least-squares reverse-time migration[1]

## 6.1 Introduction

Reverse-time migration (RTM) is a two-way wave-equation based imaging technique which is suitable for imaging complex structures (McMechan, 1983; Baysal et al., 1983; Whitmore, 1983). however, strong migration artifacts have been observed in seismic images generated by this method (Etgen et al., 2009). Least-squares RTM (LS-RTM) has been proposed to estimate images with enhanced resolution and to suppress acquisition and migration related artifacts (Lailly and Bednar, 1983; Tarantola, 1984b). LS-RTM minimizes the difference between the observed data and the synthetic data and can deliver the so-called true amplitude images with sharper reflectors (Dong et al., 2012). Researchers have also implemented LS-RTM to reduce the crosstalk noise associated with the blended shot gathers (Dai et al., 2012). The least-squares migration (LSM) problem can be alternatively formulated in the image-domain. By studying the Hessian matrix associated with LSM, we can estimate an optimal weight matrix to restore the true amplitude of the migrated events (Chavent and Plessix, 1999) and design deblurring filters to improve the spatial resolution of images (Hu et al., 2001).

One shortcoming of LS-RTM is its prohibitive computational cost which has been the main obstacle preventing LS-RTM from being used in industrial production. In each iteration, one needs to run forward finite-difference modelling, and its adjoint operator for all common shot gathers (Xu and Sacchi, 2017).

In this article, I propose a new and efficient LSRTM method formulated in the image domain. The increased computational efficiency stems from approximating the Hessian matrix as a

[1]A version of this chapter is published in Gao. W., Gian. M. and M. D. Sacchi, 2019, Fast least-squares reverse-time migration via a superposition of Kronecker products, Geophysics, accepted

superposition of Kronecker products. This paper is structured as follows. In the first section I outline the theory of image-domain LSRTM before introducing the factorization of the Hessian as a superposition of Kronecker products (Van Loan and Pitsianis, 1993). I describe two procedures for performing the Kronecker-based factorization: (1) a direct decomposition method when the full Hessian is known (2) by solving a low-rank matrix completion problem from a sparse set of elements from the Hessian. Once the Kronecker factors are obtained, the original LSRTM problem can be readily solved with the conjugate gradient (CG) method. Operations involving small, compact matrices replace expensive migration and demigration operations resulting in fast CG iterations. In the final section, I present numerical examples to evaluate the performance of my proposed method against conventional LSRTM. I observe that my method is able to produce migration images comparable to LSRTM, but at a drastically reduced computational cost.

## 6.2 Theory

### 6.2.1 Least-squares reverse-time migration

My exploration of KLSRTM is limited to the 2-D case to simplify discussions; extensions to 3D are a topic for future research. The forward modeling operators are based on the Born approximation and are expressed in terms of Green's functions. A velocity perturbation at a subsurface point $\mathbf{x} = (z, x)$ is denoted by $m(\mathbf{x})$. The terms image and velocity perturbation are used interchangeably throughout this article. The seismic data $d(\mathbf{x}_r|\mathbf{x}_s;\omega)$ acquired at receiver position $\mathbf{x}_r = (z_r, x_r)$ resulting from a source at position $\mathbf{x}_s = (z_s, x_s)$, can be expressed by the following linear equation in the frequency domain (Plessix and Mulder, 2004):

$$d(\mathbf{x}_r|\mathbf{x}_s;\omega) = \omega^2 \sum_{\mathbf{x}} f_s(\omega) G(\mathbf{x}_r|\mathbf{x};\omega) G(\mathbf{x}|\mathbf{x}_s;\omega) m(\mathbf{x}), \tag{6.1}$$

where $\omega$ is the angular frequency, $f_s(\omega)$ is the source signature, and $G(\mathbf{x}|\mathbf{x}_s;\omega)$ is the Green's function for an impulsive point source at $\mathbf{x}_s$. To find the model that optimally fits the observed data, I formulate the inverse problem in data space and minimize the following cost function (Lailly and Bednar, 1983; Tarantola, 1984a; Nemeth et al., 1999; Kühl and Sacchi, 2003):

$$J(\mathbf{m}) = \frac{1}{2} \sum_{\omega} \sum_{\mathbf{x}_s} \sum_{\mathbf{x}_r} ||d(\mathbf{x}_r|\mathbf{x}_s;\omega) - d_{obs}(\mathbf{x}_r|\mathbf{x}_s;\omega)||_2^2, \tag{6.2}$$

where $d_{obs}(\mathbf{x}_r|\mathbf{x}_s;\omega)$ indicates the observed data recorded at the surface, and $\|\cdot\|_2$ is the L2 norm. The model vector $\mathbf{m}$ is the discretized form of $m(\mathbf{x})$. The Born approximation operator in equation 6.1 can be represented more compactly as a linear operator $\mathbf{L}$. The

least-squares cost function can be simplified to

$$J(\mathbf{m}) = \|\mathbf{L}\mathbf{m} - \mathbf{d}_{obs}\|_2^2, \tag{6.3}$$

where $\mathbf{d}_{obs}$ represents the data, for all time steps, sources and receivers, concatenated into a long vector. The cost function is quadratic in terms of the model parameter $\mathbf{m}$. The minimum of equation 6.3 can be computed by setting the derivative of $J(\mathbf{m})$ with respect to the model parameter $\mathbf{m}$ to zero. The resultant normal equation can be expressed as

$$\mathbf{L}^T\mathbf{L}\mathbf{m} = \mathbf{L}^T\mathbf{d}_{obs}, \tag{6.4}$$

where $\mathbf{L}^T$ denotes the adjoint operator of the Born approximation (Xu and Sacchi, 2017). The normal equations can be reformulated by first substituting the migrated image $\mathbf{L}^T\mathbf{d}_{obs}$ by $\mathbf{m}'$ (Baysal et al., 1983; Levin, 1984; Chang and McMechan, 1986). The term $\mathbf{L}^T\mathbf{L}$ corresponds to the Hessian matrix $\mathbf{H}$ of LSRTM. Elements of $\mathbf{H}$ are second-order derivatives of the objective function $J(\mathbf{m})$ with respect to the model parameter $\mathbf{m}$. Using the two substitutions, equation 6.4 can be written compactly as

$$\mathbf{H}\mathbf{m} = \mathbf{m}'. \tag{6.5}$$

The linear system in equation 6.5 is known as the image-domain formulation of LSRTM (Fletcher et al., 2016). The true image $\mathbf{m} = \mathbf{H}^{-1}\mathbf{m}'$ can be estimated by minimizing the following least-squares problem:

$$\underset{\mathbf{m}}{\operatorname{argmin}} \|\mathbf{H}\mathbf{m} - \mathbf{m}'\|_2^2. \tag{6.6}$$

The minimizer of the above optimization problem is an approximation to the solution of the normal equations expressed in equation 6.4.

### 6.2.2 Superposition of Kronecker products

My efficient LSRTM algorithm is formulated by approximating the Hessian matrix as a superposition of Kronecker products. Before introducing the approximation, I examine the structure of the Hessian matrix for a small homogeneous velocity model with constant density; the complete matrix is displayed in Figure 6.1a. The model is discretized on a regular grid of size $n_z$=50 $\times$ $n_x$=101. The model vector (image) $\mathbf{m}$ and the Hessian matrix in equation 6.6 have dimensions of $n_z \cdot n_x \times 1$ and $n_z \cdot n_x \times n_z \cdot n_x$, respectively. The model vector can be represented as an $n_z \times n_x$ matrix $\mathbf{M}$ after a reshaping operation. Lexicographical column-wise ordering is adopted for the vectorization of matrices and vice

versa. Figure 6.1b shows a windowed portion of the Hessian matrix (indicated by the red box in Figure 6.1a). The top-leftmost block of the window is displayed in Figure 6.1c. Figure 6.1 depicts the block-band structure of the Hessian matrix where each block has dimensions $n_z \times n_z$. The number of blocks is $n_x$ in both vertical and horizontal directions. The block-band structure is characteristic of the LSRTM Hessian and is a consequence of the finite-frequency nature of seismic data and the lexicographical ordering used when discretizing multi-dimensional model spaces into vectors (Pratt et al., 1998).

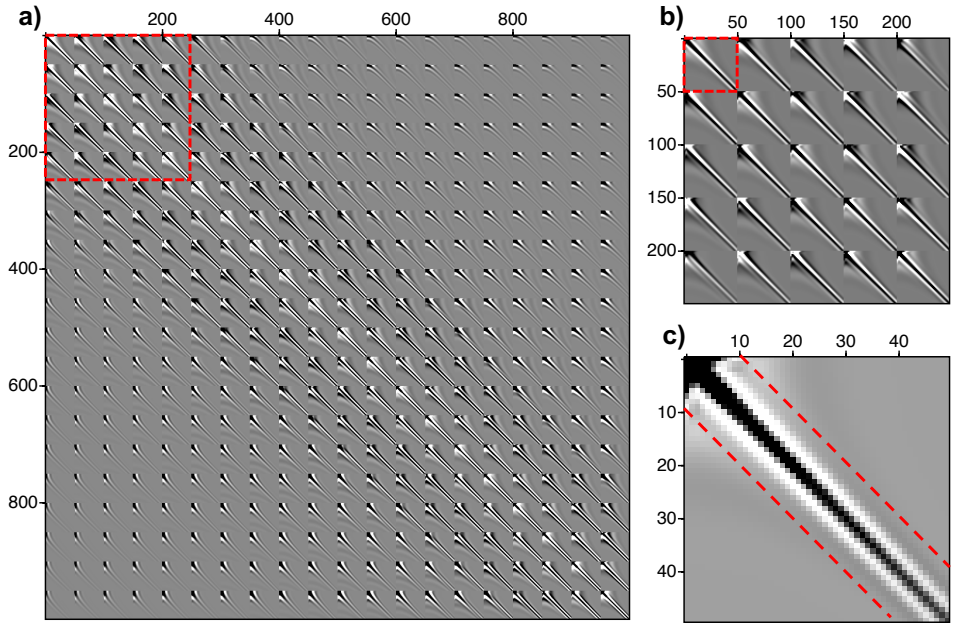


Figure 6.1: Block-band structure of the Hessian matrix. (a) Upper left crop of the full matrix ($0.2n_z \cdot n_x \times 0.2n_z \cdot n_x$). (b) Windowed portion of the Hessian ($5n_z \times 5n_z$, dashed red square in (a)). (c) Single $n_z \times n_z$ block of the Hessian matrix (dashed red square in (b)).

Low-rank approximations are one of the most prevalent forms of matrix approximation and can be obtained from a truncated singular value decomposition (SVD). For a highly structured, diagonally-dominant matrix such as the LSRTM Hessian, low-rank approximations may not be ideal. I observe that a superposition of Kronecker products can approximate the Hessian matrix more compactly than a superposition of rank-one matrices obtained by the SVD (Van Loan and Pitsianis, 1993; Van Loan, 2000). The approximation can be expressed

as follows:

$$\mathbf{H} \approx \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i, \tag{6.7}$$

where the Kronecker factors $\mathbf{A}_i$ and $\mathbf{B}_i$ are square matrices of size $n_x \times n_x$ and $n_z \times n_z$, respectively. The symbol $\otimes$ and $k$ denote the Kronecker product and the number of factors, respectively. Replacing the Hessian matrix in equation 6.6 with the Kronecker-based approximation, I obtain

$$\underset{\mathbf{m}}{\operatorname{argmin}} \| \left( \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i \right) \mathbf{m} - \mathbf{m}' \|_2^2. \tag{6.8}$$

The size of $\mathbf{H}$ ($n_z \cdot n_x \times n_z \cdot n_x$) makes it prohibitively expensive to store in memory. As the approximation $\sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i$ has the same dimensions as $\mathbf{H}$, it is also too large to store. The storage requirement can be vastly reduced by utilizing the following property of Kronecker products (Van Loan and Pitsianis, 1993)

$$(\mathbf{A} \otimes \mathbf{B})\mathbf{m} = vec(\mathbf{B}\mathbf{M}\mathbf{A}^T), \tag{6.9}$$

where the operator $vec$ represents the vectorization of a matrix (e.g., $\mathbf{m} = vec(\mathbf{M})$). Equation 6.8 can be converted to

$$\underset{\mathbf{M}}{\operatorname{argmin}} \| \sum_{i=1}^{k} \mathbf{B}_i \mathbf{M} \mathbf{A}_i^T - \mathbf{M}' \|_F^2, \tag{6.10}$$

where $\mathbf{M}'$ is the migrated image in matrix form and $\| \cdot \|_F$ denotes the fmatrix Frobenius norm. The Frobenius norm has been used to ensure that equations 6.8 and 6.10 are equivalent. Equation 6.10 indicates that we only need to save the Kronecker factors to solve the image-domain LSRTM problem. The Kronecker factors are small matrices relative to the size of the full Hessian. I solve equation 6.10 via iterative algorithms such as the CG method (Paige and Saunders, 1982; Shewchuk et al., 1994). These iterative methods require the adjoint operator of the Kronecker-based Hessian approximation. By definition, the adjoint operator satisfies

$$\begin{aligned} \left\langle (\sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i)\mathbf{m}, \mathbf{m}' \right\rangle &= \left\langle \mathbf{m}, (\sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i)^T \mathbf{m}' \right\rangle, \\ &= \left\langle \mathbf{m}, (\sum_{i=1}^{k} (\mathbf{A}_i \otimes \mathbf{B}_i)^T) \mathbf{m}' \right\rangle, \\ &= \left\langle \mathbf{m}, (\sum_{i=1}^{k} \mathbf{A}_i^T \otimes \mathbf{B}_i^T) \mathbf{m}' \right\rangle, \end{aligned} \tag{6.11}$$

where $\langle , \rangle$ represents the dot product between two vectors. Using equation 6.9, it is evident that the adjoint of the Kronecker-based Hessian approximation maps from the space of migrated images to the model space via $vec(\sum_{i=1}^{k} \mathbf{B}_i^T \mathbf{M}^{'} \mathbf{A}_i)$ (Schacke, 2004).

The Kronecker factors can be estimated by solving the following optimization problem

$$\underset{\mathbf{A}_i, \mathbf{B}_i}{\operatorname{argmin}} \|\mathbf{H} - \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i\|_F^2. \tag{6.12}$$

Pitsianis (1997) solved equation 6.12 by converting it into a low-rank matrix decomposition problem

$$\underset{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}}{\operatorname{argmin}} \|\tilde{\mathbf{H}} - \tilde{\mathbf{B}} \tilde{\mathbf{A}}^T\|_F^2, \tag{6.13}$$

where $\tilde{\mathbf{A}} = \begin{bmatrix} vec(\mathbf{A}_1) & vec(\mathbf{A}_2) & \cdots & vec(\mathbf{A}_k) \end{bmatrix}$, $\tilde{\mathbf{B}} = \begin{bmatrix} vec(\mathbf{B}_1) & vec(\mathbf{B}_2) & \cdots & vec(\mathbf{B}_k) \end{bmatrix}$f, and $\tilde{\mathbf{H}}$ is the rearranged Hessian. The rearranged Hessian is defined as $\tilde{\mathbf{H}} = \mathcal{R}(\mathbf{H})$, where $\mathcal{R}$ is a rearrangement operator that reorders the elements of a matrix. A complete description of $\mathcal{R}$ is provided in Appendix C. The size of $\tilde{\mathbf{H}}$ is $n_z \cdot n_z \times n_x \cdot n_x$, the size of $\tilde{\mathbf{A}}$ is $n_x \cdot n_x \times k$ and the size of $\tilde{\mathbf{B}}$ is $n_z \cdot n_z \times k$. The solution to equation 6.13 is given by the SVD of $\tilde{\mathbf{H}}$ followed by a reshaping of the first $k$ singular vectors into square matrices. Algorithm 6 in Appendix C describes how to estimate the Kronecker factors using an SVD of the rearranged Hessian.

Figure 6.2 compares the relative approximation error (in the Hessian) as a function of the number of factors ($k$) used for the matrix approximation. In the low-rank case, the term factor refers to the singular vectors of $\mathbf{H}$, whereas in the Kronecker-based approximation it refers to the pair $\{\mathbf{A_i}, \mathbf{B_i}\}$. The relative matrix approximation error is defined as

$$\mathcal{E}_1(k) = \frac{\|\mathbf{H} - \sum_{i=1}^{k} \lambda_i \mathbf{u}_i \mathbf{v}_i^T\|_F^2}{\|\mathbf{H}\|_F^2}$$

for the low-rank approximation and

$$\mathcal{E}_2(k) = \frac{\|\mathbf{H} - \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i\|_F^2}{\|\mathbf{H}\|_F^2}$$

for the Kronecker-based approximation. fHere, $\lambda_i$, $\mathbf{u}_i$ and $\mathbf{v}_i$ are the singular value and vectors obtained via SVD decomposition. For a fixed number of factors, the Kronecker factorization produces considerably lower relative approximation errors than the low-rank approximation. For instance, for $k = 50$ factors the error $\mathcal{E}_1 = 0.43$ for the low-rank approximation compared to $\mathcal{E}_2 = 0.037$ for the Kronecker-based approximation. The Kronecker factorization naturally results in block-band matrices and can therefore approximate the Hessian with a small number of factors (see Appendix E for more on this). In contrast,

the low-rank approximation requires a large number of singular vectors in order to capture the block-band structure of the Hessian. The interpretation of this result is that while the original Hessian may not necessarily be low-rank, the rearranged Hessian is. The error $\mathcal{E}_2(k)$ approaches 0 as $k$ approaches the rank of $\tilde{\mathbf{H}}$.

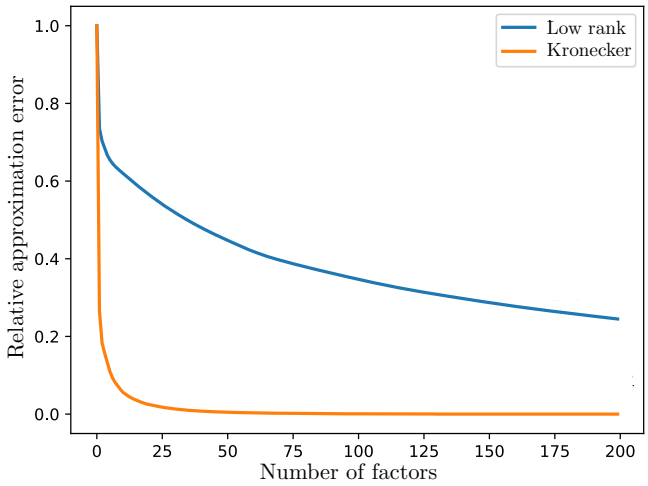


Figure 6.2: Relative approximation error as a function of the number of factors $k$. For the low-rank approximation, the term 'factor' refers to a singular vector, whereas for the Kronecker factorization it refers to a pair of Kronecker factors ($\{\mathbf{A}_i, \mathbf{B}_i\}$). The blue line corresponds to the error $\mathcal{E}_1$ incurred when we approximate the Hessian with a truncated SVD. The orange line corresponds to the error $\mathcal{E}_2$ associated with the Kronecker-based approximation of the Hessian.

### 6.2.3 Kronecker factor estimation using sparse Hessian samples

For problems of typical size, it becomes infeasible to compute the full Hessian explicitly thus making Algorithm 6 impractical. Fortunately, only the Kronecker factors $\mathbf{A}_i$ and $\mathbf{B}_i$, not $\mathbf{H}$, are required to solve image-domain LSRTM (equation 6.10). The final stage in my development of KLSRTM tackles the challenge of estimating the Kronecker factors without forming $\mathbf{H}$ explicitly. To do this, I proceed from my observation that the rearranged Hessian $\tilde{\mathbf{H}}$ is approximately low-rank (Figure 6.2). Recent advances in the field of matrix completion have shown that a few randomly extracted elements of a low-rank matrix can be used to reconstruct the entire matrix (Candès and Recht, 2009; Cai et al., 2010; Candès and Tao, 2010; Jain et al., 2013). The problem of estimating Kronecker factors can thus be reformulated as a matrix completion problem. The estimation problem posed in equation

6.13 becomes

$$\underset{\tilde{\mathbf{A}}\in\mathbb{R}^{{n_x}^2\times k},\ \tilde{\mathbf{B}}\in\mathbb{R}^{{n_z}^2\times k}}{\operatorname{argmin}} \|P_\Omega(\tilde{\mathbf{H}}-\tilde{\mathbf{B}}\tilde{\mathbf{A}}^T)\|_F^2, \tag{6.14}$$

where $\Omega$ represents the set of observed samples of $\tilde{\mathbf{H}}$. Wang et al. (2015b) propose an efficient and scalable low-rank matrix completion algorithm that extends the orthogonal matching pursuit method from the vector case to the matrix case. Their algorithm is implemented to estimate the matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ from random samples of $\tilde{\mathbf{H}}$; the factors $\mathbf{A}_i$, and $\mathbf{B}_i$ are extracted from the columns of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$, respectively. Algorithm 5 in Appendix D details the matrix completion algorithm. The completion algorithm requires samples from the Hessian, a potentially expensive task. In the following section, I devise a preferential sampling strategy that exploits fa priori knowledge about the characteristic structure of the Hessian. In doing so, I limit the computational cost of the sampling step.

### 6.2.4 A preferential sampling strategy for Hessian samples

In terms of Green's functions, an arbitrary element of the Hessian can be expressed as (Plessix and Mulder, 2004; Valenciano et al., 2006; Tang, 2009)

$$H(\mathbf{x},\mathbf{y}) = \mathscr{R}\left\{\sum_\omega \omega^4|f(\omega)|^4 \sum_{\mathbf{x}_s} G(\mathbf{x}|\mathbf{x}_s;\omega)G^*(\mathbf{y}|\mathbf{x}_s;\omega) \times \sum_{\mathbf{x}_r} G(\mathbf{x}|\mathbf{x}_r;\omega)G^*(\mathbf{y}|\mathbf{x}_r;\omega)\right\}, \tag{6.15}$$

where $\mathscr{R}$ denotes the real part of a complex value and $\mathbf{x}$ and $\mathbf{y}$ represent two image points in the subsurface. Adjoint Green's functions are denoted with the symbol $^*$. The Green's function is computed by solving the acoustic wave-equation in the frequency domain using the mixed-grid finite-difference method (Hustedt et al., 2004). Given the size of $\mathbf{H}$, even 10% of the elements of $\mathbf{H}$ can take a large amount of memory. Previous studies have established that the Hessian is a banded, diagonally-dominant matrix with elements that decay away from the diagonals (Pratt et al., 1998; Chavent and Plessix, 1999; Plessix and Mulder, 2004; Valenciano et al., 2006). To avoid sampling small values that have a negligible contribution to the Hessian, I adopt a preferential sampling scheme that confines samples to predetermined regions.

To restrict sampling to a subset of Hessian elements, I only consider points $\mathbf{x}+\mathbf{r}$ within the vicinity of a point $\mathbf{x}$. The "offset" $\mathbf{r} = (r_z, r_x)$ satisfies $|r_x| \leq r_x^{win}$ and $|r_z| \leq r_z^{win}$. The values $r_x^{win}$ and $r_z^{win}$ are user-defined half-lengths of the horizontal and vertical windows, respectively (demonstrated by the bounding grey box in Figure 6.3a). With this restriction,

I replace equation 6.15 with

$$H(\mathbf{x}, \mathbf{x}+\mathbf{r}) = \mathscr{R}\left\{ \sum_{\omega} \omega^4 |f(\omega)|^4 \sum_{\mathbf{x}_s} G(\mathbf{x}|\mathbf{x}_s;\omega) G^*(\mathbf{x}+\mathbf{r}|\mathbf{x}_s;\omega) \times \sum_{\mathbf{x}_r} G(\mathbf{x}|\mathbf{x}_r;\omega) G^*(\mathbf{x}+\mathbf{r}|\mathbf{x}_r;\omega) \right\}. \tag{6.16}$$

Elements outside of the user-defined window are ignored as their magnitudes are assumed to be small. I define $n_{r_x}$ and $n_{r_z}$ as the number of discrete grid points for the half window lengths $r_x^{win}$ and $r_z^{win}$, respectively. After restriction, the number of Hessian samples considered reduces by a factor of $(n_z\, n_x\, (2n_{r_z}+1)\, (2n_{r_x}+1))/(n_z\, n_x)^2$. This sampling strategy is equivalent to imposing a block-band structure on the Hessian where elements outside of the designated band are zero (instead of small, non zero elements). Schematic diagrams displayed in Figure 6.3 illustrate the physical interpretation of equation 6.16 and how it imposes block-band structure on the Hessian. Mathematically, the sampling restriction can be enforced by constraining the Kronecker factors $\mathbf{A}_i$ and $\mathbf{B}_i$ to be band matrices with bandwidths $n_{r_x}$ and $n_{r_z}$, respectively. For details about how this constraint restricts the Hessian samples, see Appendix E. The original factor estimation in equation 6.12 becomes a constrained optimization problem of the form:

$$\underset{\mathbf{A}_i \in \mathbb{M}_{n_x}^{n_{r_x}},\, \mathbf{B}_i \in \mathbb{M}_{n_z}^{n_{r_z}}}{\operatorname{argmin}} \|\mathbf{H} - \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i\|_F^2, \tag{6.17}$$

where $\mathbb{M}_{n_x}^{n_{r_x}}$ represents a set of square band matrices of size $n_x \times n_x$ and bandwidth $n_{r_x}$. In other words, for a square matrix $\mathbf{A} \in \mathbb{M}_{n_x}^{n_{r_x}}$, an element $a_{ij} = 0$ if $|i-j| > n_{r_x}$. Equation 6.17 can be converted to an unconstrained low-rank matrix decomposition problem expressed as

$$\underset{\hat{\mathbf{A}},\, \hat{\mathbf{B}}}{\operatorname{argmin}} \|\mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a - \hat{\mathbf{B}} \hat{\mathbf{A}}^T\|_F^2, \tag{6.18}$$

where $\hat{\mathbf{A}} = \mathbf{Q}_a^T \tilde{\mathbf{A}}$ and $\hat{\mathbf{B}} = \mathbf{Q}_b^T \tilde{\mathbf{B}}$ for projection matrices $\mathbf{Q}_a$ and $\mathbf{Q}_b$. The band Kronecker factors $(\mathbf{A}_i, \mathbf{B}_i)$ are obtained by reshaping the columns of $\mathbf{Q}_a \hat{\mathbf{A}}$ and $\mathbf{Q}_b \hat{\mathbf{B}}$ into square matrices. Thorough descriptions of the sampling strategy and the constrained optimization problem are provided in Appendix E. The (constrained) Kronecker factors can be estimated by converting equation 6.18 into a matrix completion problem analogous to equation 6.14,

$$\underset{\hat{\mathbf{A}},\, \hat{\mathbf{B}}}{\operatorname{argmin}} \|P_\Omega(\mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a - \hat{\mathbf{B}} \hat{\mathbf{A}}^T)\|_F^2. \tag{6.19}$$

To supplement the development in this section Appendix E discusses the constrained Kronecker factor estimation problem, Appendix D outlines the matrix completion algorithm and Appendix F explains the connection between $\mathbf{H}$ and $\mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a$.

| Algorithm | PDE solves | Memory cost |
|---|---|---|
| **RTM** (I. Compute $\mathbf{L}^T\mathbf{d}$) | $2N_s + N_s$ | $N_f N + N_f N$ |
| **LSRTM** | $4N_{iter} \cdot N_s + N_s$ | $N_f N + N_f N$ |
| **KLSRTM** | | |
| II. Compute $G(\mathbf{x}\,\vert\,\mathbf{x}_s;\omega),\ G(\mathbf{x}\,\vert\,\mathbf{x}_r;\omega)$ | $N_s + N_r$ | $3 \times 9N + 2N$ |
| III. Compute $\mathbf{P}_\Omega(\mathbf{H})$ | - | $3N \cdot (2n_{r_z} + 1) \cdot (2n_{r_x} + 1) \cdot sr$ |
| IV. Solve **MC** | - | $3k \cdot (n_z \cdot (2n_{r_z} + 1) + n_x \cdot (2n_{r_x} + 1))$ |

Table 6.1: Computational cost summary. The numbered stages are identified in the image-domain LSRTM branch of Figure 6.3. Requirements are provided for a single frequency.

### 6.2.5 Computational cost

At this stage I compare the computational resource requirements for the proposed algorithm and LSRTM. To aid the comparison, I provide a simplified workflow (Figure 6.4) that describes only the essential, most resource intensive stages of each algorithm. Since KLSRTM using the SVD is not practical for large problems, I do not explore its resource requirements. The computational cost and memory requirements of RTM, LSRTM and KLSRTM are summarized in Table 6.1. Computational cost is evaluated in terms of partial-differential equation (PDE) solves for a single frequency; other operations are neglected as their costs are minimal in comparison. Memory costs are evaluated by counting the number of floating point allocations required.

For $N_s$ sources in a survey, RTM performs $3N_s$ PDE solves. In my implementation, I pre-compute the source-side wavefield for each source location and the save the boundary values of each snapshot to disk. The boundary values are used to reconstruct the source-side wavefield on-the-fly during migration to reduce the overhead of large disk IO (Dussaud et al., 2008). The three PDE solves consist of computing the source and receiver side wavefields along with reconstruction of the background wavefield. I allocate memory for snapshots of the source- and receiver-side wavefields comprised of $N_f$ components (e.g. pressure, velocity components); $N_f$ is used to provide a more implementation-independent cost. In my application $N_f = 8$ following Xu and Sacchi (2017). For a 2D model, the total number of discrete grid points is $N = n_z \cdot n_x$. Memory costs in Table 6.1 are for a single shot and scale linearly for parallel computations. For RTM and LSRTM, memory costs are split into source and receiver-side wavefield contributions.

While the memory cost of LSRTM is similar to that of RTM, the computational cost differs due to the use of iterative solvers to solve equation 6.3. Each iteration requires the application of the forward and adjoint operators at a cost of two PDE solves each. For a maximum

number of iterations $N_{iter}$, the total number of PDE solves becomes $4N_{iter} \cdot N_s + N_s$. The additional $N_s$ PDE solves compute the boundary values of the background source-side wavefield.

I divide the computational cost analysis of KLSRTM into three parts: Green's function computation, evaluation of the Hessian samples and matrix completion (MC). Green's functions are obtained by solving $N_s + N_r$ PDEs, where $N_r$ is the number of receivers. The Helmholtz operator for given frequency is a complex sparse matrix with $9N$ non-zero elements (Hustedt et al., 2004). To simplify the discussion of memory cost, I assume only the non-zero elements are saved, along with the corresponding row and column indices, resulting in a cost of $3 \times 9N$. The memory cost can be potentially reduced with other sparse-matrix formats (e.g., compressed sparse column). A further $2N$ complex numbers are allocated for the wavefield and source term. In KLSRTM, the largest memory cost stems from saving preferential samples of the Hessian matrix. If I consider a sampling rate $sr$ and a window size $(2n_{r_z}+1) \times (2n_{r_x}+1)$ about each image point, the total number of floating point numbers stored is $3N \cdot (2n_{r_z}+1) \times (2n_{r_x}+1) \cdot sr$. After the samples of the Hessian matrix are obtained, I estimate the Kronecker factors $\mathbf{A}_i$ and $\mathbf{B}_i$. Both are also saved in sparse matrix format and the memory cost of each $\mathbf{A}_i$ is $3n_x \cdot (2n_{r_x}+1)$ and $\mathbf{B}_i$ is $3n_z \cdot (2n_{r_z}+1)$. For $k$ Kronecker factor pairs, the final memory cost of the factors is $3k \cdot (n_z \cdot (2n_{r_z}+1) + n_x \cdot (2n_{r_x}+1))$. While I have not verified them, approximate memory costs for elastic or 3D extensions of KLSRTM can be gauged by either adjusting the number of field variables, factors and/or increasing the dimensions of the grid and windows. The number of PDE solves should be the same although each PDE will be more expensive to solve.

## 6.3 Numerical Examples

I use a series of numerical examples to test aspects of the proposed method. The examples can be separated into two primary categories. The first set examines KLSRTM when the Kronecker factors are estimated using an SVD of the rearranged Hessian. Examples in this category explicitly compute the entire Hessian; I refer to this category as KLSRTM-SVD. Examples from the second category do not compute the full Hessian, only a small percentage of its samples. In this group, the Kronecker factors are computed using a matrix-completion algorithm; this category is referred to as KLSRTM-MC. The first set of examples (KLSRTM-SVD) are designed as proof of concepts for the Kronecker factorization and KLSRTM. The second set of examples (KLSRTM-MC) demonstrate applications of KLSRTM for larger models using a more efficient and scalable algorithm to estimate the Kronecker factors. For examples that perform LSRTM, data are generated using a constant-density, acoustic wave-equation solver implemented using the staggered-grid, finite-difference time-domain method

(Virieux, 1986). A summary of the numerical parameters and algorithmic categories for each example are presented in Table 6.2.

### 6.3.1 KLSRTM-SVD

#### Example 1

The first example presents Kronecker factors computed using an SVD of the rearranged Hessian. The factors are used to evaluate the merits of a Kronecker-based factorization compared to a low-rank decomposition of the Hessian. The Earth model used for modeling is a cropped version of the Marmousi model discretized on a regular 2D grid with dimensions $n_x = 200$ and $n_z = 100$. The true and smooth models are displayed in Figure 6.5. The smooth velocity model is obtained by convolving the true model with a 2D Gaussian kernel with a standard deviation of 50 m. Selected Kronecker factors $\mathbf{A}_i$ and $\mathbf{B}_i$ are displayed in Figures 6.6 and 6.7. For this model, $\mathbf{A}_i$ and $\mathbf{B}_i$ have sizes of $200 \times 200$ and $100 \times 100$, respectively. The Kronecker factors are band matrices with elements that decay away from the diagonals. The complexity of the factors appears to increase for larger factor numbers. The factors appear to be a mixture of (approximately) symmetric and antisymmetric matrices.

#### Example 2

The second example demonstrates the feasibility of KLSRTM and compares it against conventional LSRTM. I only examine the dependency of the inversion on the number of Kronecker factors used to approximate the Hessian. I do not assess performance gains for this example since the Kronecker factors are computed with the SVD.

The target perturbation model consists of regularly spaced point scatterers and is displayed in Figure 6.8a. The true migration image $\mathbf{Hm}$ exhibits low-frequency artifacts and unbalanced amplitudes that can be observed in Figure 6.8b. Figures 6.8c-h display the approximate migration image achieved by replacing $\mathbf{H}$ with a superposition of Kronecker products ($\sum_{i=1}^{k} \mathbf{B}_i \mathbf{M} \mathbf{A}_i^T$). The similarity between the true (Figure 6.8b) and the approximate (Figure 6.8c-h) migration images increases as more Kronecker factors are included in the approximation.

Figure 6.9 compares the results of conventional LSRTM and KLSRTM. The result of data-domain LSRTM appears in Figure 6.9a and is obtained from 50 CGLS iterations. The KLSRTM results for $k = 1$, 4, 8, 16 and 32 factors are shown in Figures 6.9b-f. Figures 6.9b-f ,which are obtained by solving equation 6.10 with 50 CGLS iterations. In all cases, migration artifacts have been attenuated and the scatterers are better focused when compared to the

corresponding migration image. The more Kronecker factors that are used to approximate the Hessian, the closer the similarity between the LSRTM and KLSRTM results. For $k > 32$, the KLSRTM image is largely indistinguishable from the LSRTM one. This example verifies the viability of KLSRTM when the Hessian and Kronecker factors are computed directly.

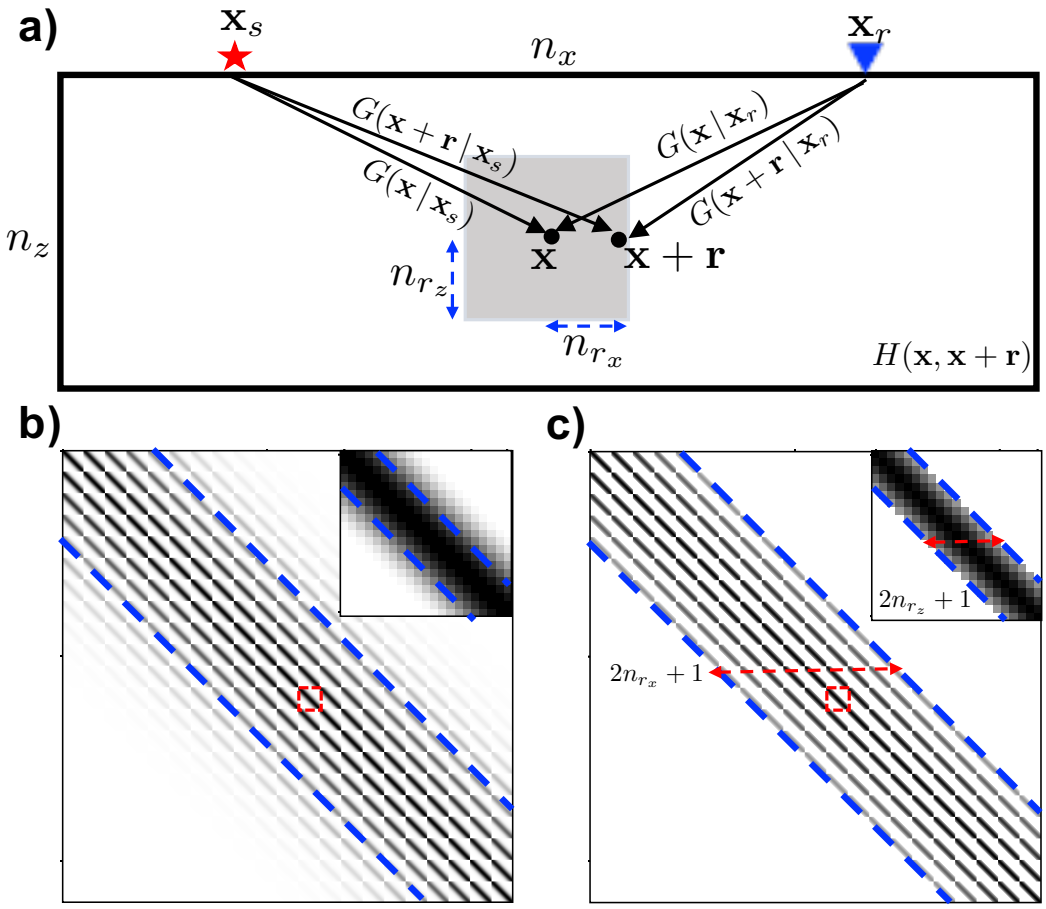


Figure 6.3: Schematic representation of (a) two point scatterers in a 2D acoustic model. The interaction of two scatterers at a particular set of locations produces a single element of the Hessian. Only considering interactions within a $(2n_{r_z}+1)\times(2n_{r_x}+1)$ local window (grey box) around a point scatterer equates to only considering a subset of the elements of the Hessian. Schematic representations of (b) the full Hessian and (c) the subset of elements that are used for preferential sampling of the Hessian. For (b) and (c), the insets represent a single block from the Hessian (red square). The dashed blue lines mark the bandwidth in relation to the local window in (a). The preferential sampling neglects elements that are small.

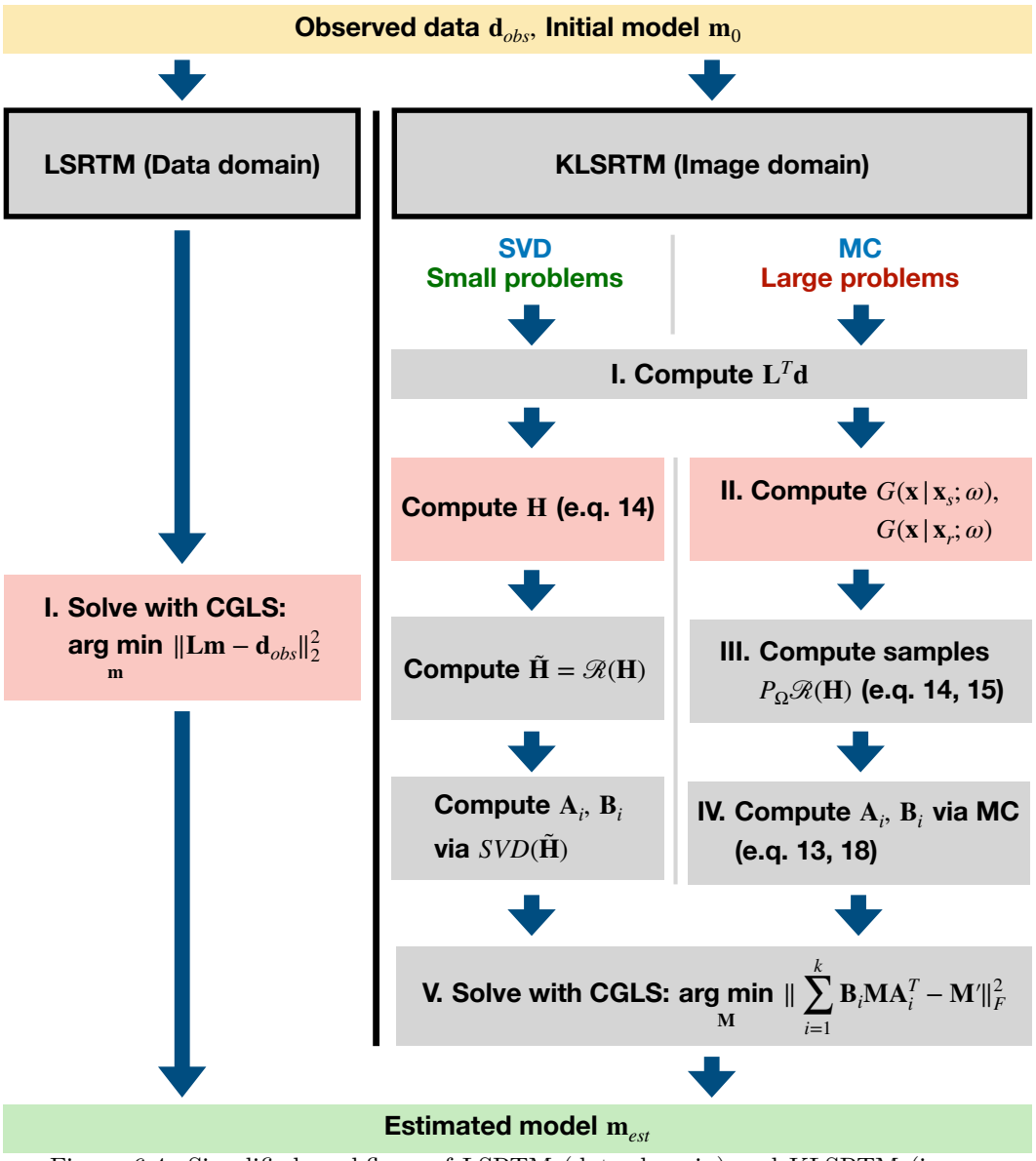


Figure 6.4: Simplified workflows of LSRTM (data domain) and KLSRTM (image domain). KLSRTM-SVD computes the exact Hessian and is only suitable for small problems (e.g. Examples 1 and 2). KLSRTM-MC estimates the Kronecker factors using a matrix completion algorithm and is designed for larger problems (e.g. Examples 3-5). The workflows are composed of distinct computational steps; red coloured boxes indicate the most resource intensive stage of each algorithm. The computational resource requirements of the numbered boxes are explored in Table 6.1.

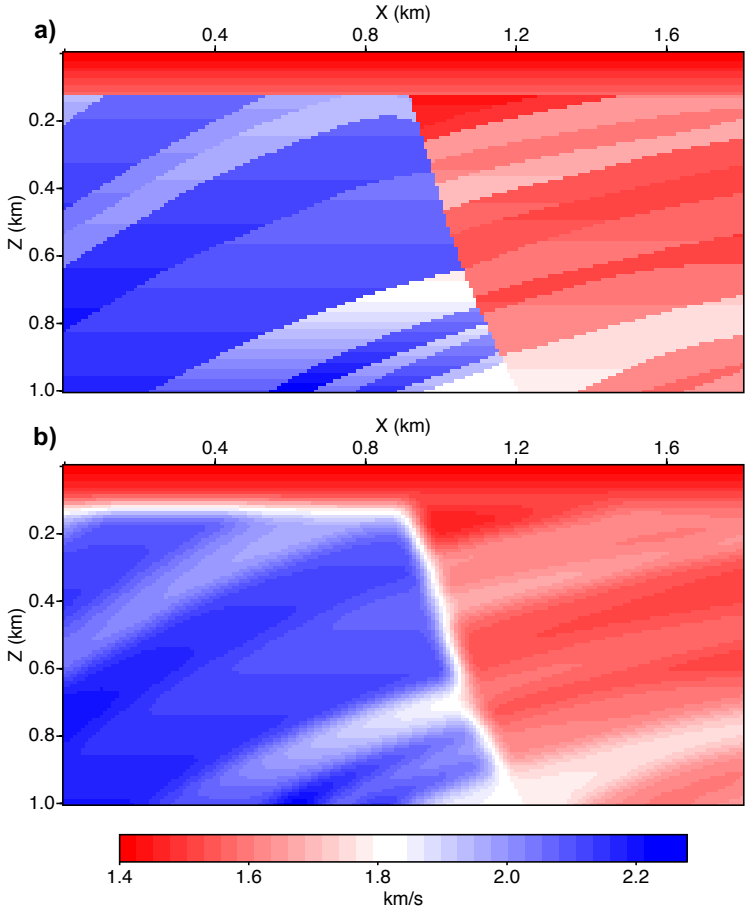


Figure 6.5: Cropped Marmousi model. (a) True velocity model. (b) Smooth velocity model used for testing Kronecker factor estimation.

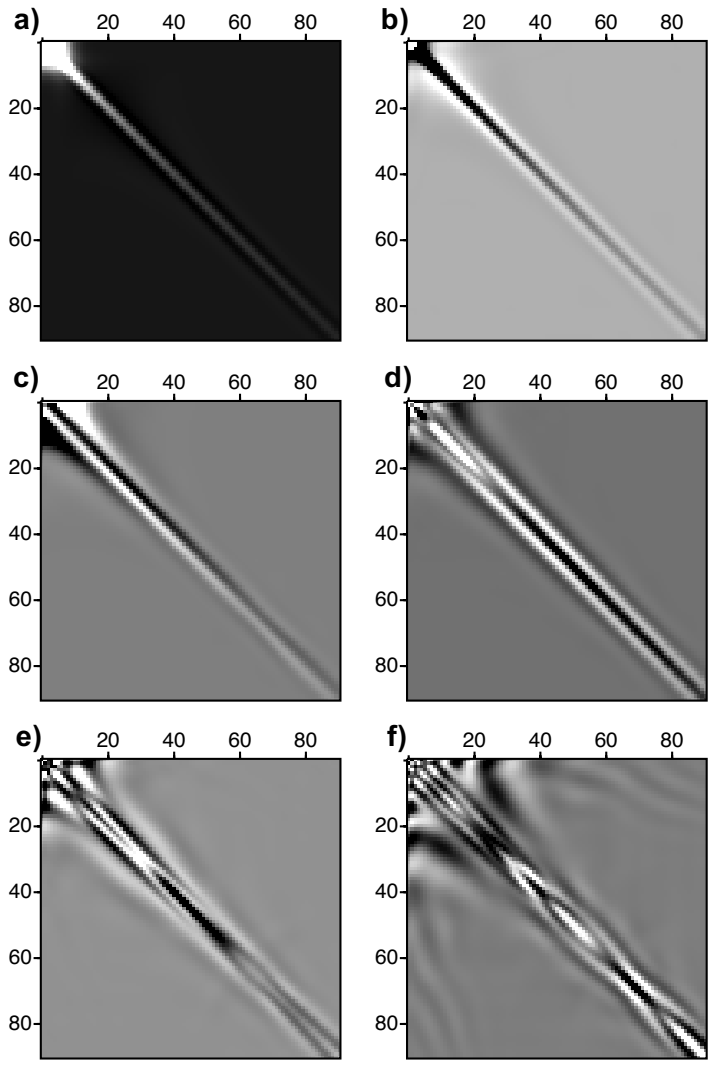


Figure 6.6: The Kronecker factor matrices $\mathbf{B}_i$ (size $n_z \times n_z$) for (a) $i = 1$, (b) $i = 4$, (c) $i = 8$, (d) $i = 16$, (e) $i = 32$, and (f) $i = 64$.

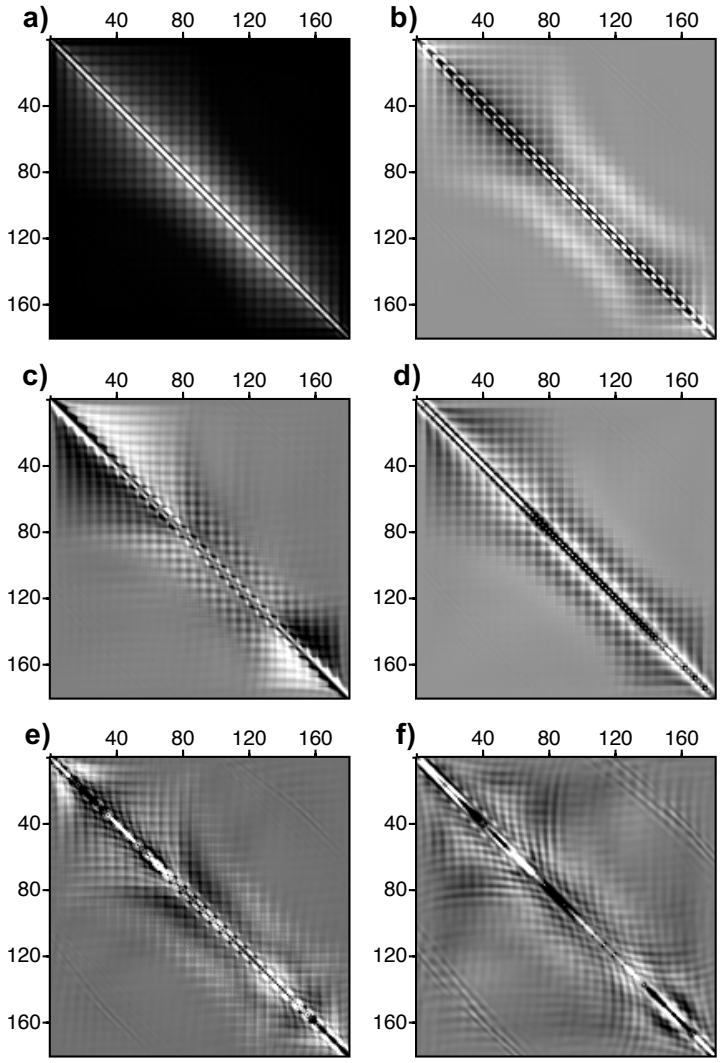


Figure 6.7: The Kronecker factor matrices $\mathbf{A}_i$ (size $n_x \times n_x$) for (a) $i = 1$, (b) $i = 4$, (c) $i = 8$, (d) $i = 16$, (e) $i = 32$, and (f) $i = 64$.

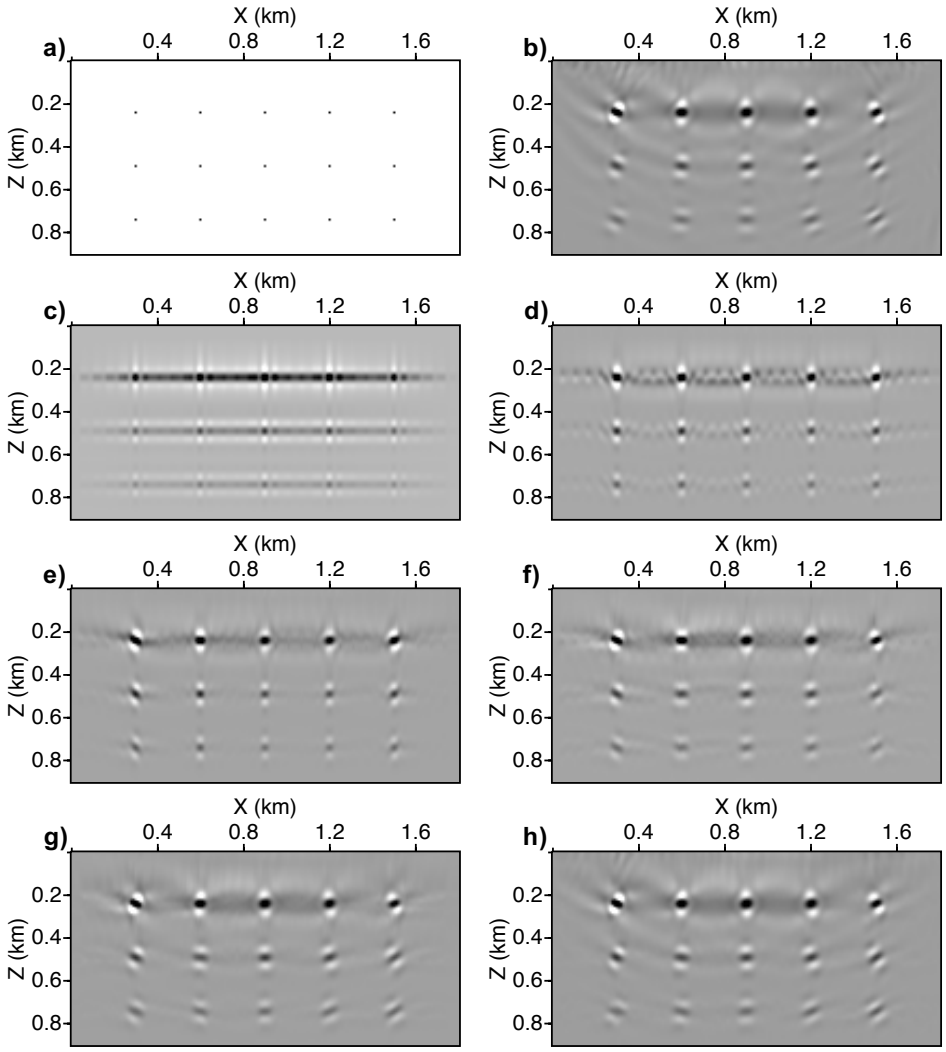


Figure 6.8: Point scatterer experiment. (a) The true image $\mathbf{M}$ consisting of evenly distributed point scatterers. (b) Migration image obtained by applying the Hessian $\mathbf{H}$ to $\mathbf{m}$. The remaining panels depict approximate migration images obtained via the sum of the first $k$ Kronecker products $\sum_{i=1}^{k} \mathbf{B}_i \mathbf{M} \mathbf{A}_i^T$ for (c) $k = 1$, (d) $k = 4$, (e) $k = 8$, (f) $k = 16$, (g) $k = 32$, and (h) $k = 64$.

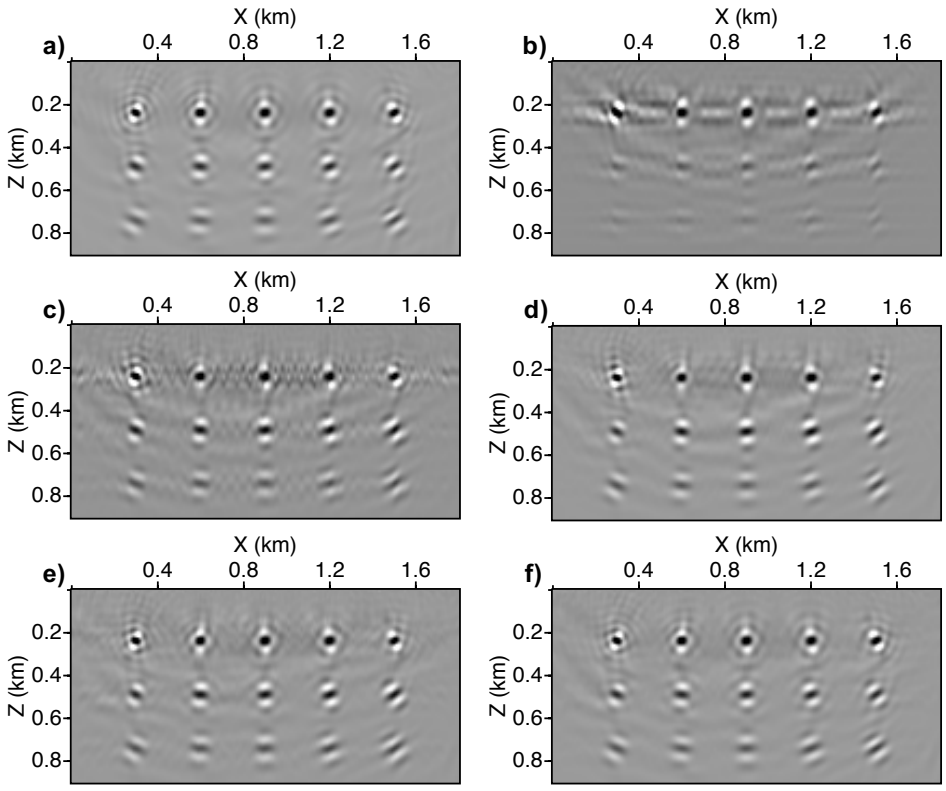


Figure 6.9: A comparison of LSRTM results for the point scatterer experiment. (a) Conventional LSRTM image obtained by solving $\|\mathbf{Hm} - \mathbf{m}'\|_2^2$. The remaining panels depict the KLSRTM images obtained by solving $\|\sum_{i=1}^{k} \mathbf{A}_i \mathbf{M} \mathbf{B}_i^T - \mathbf{M}'\|_F^2$ for (c) $k = 4$, (d) $k = 16$, (e) $k = 32$, and (f) $k = 64$.

| | Example 1<br>Cropped Marmousi | Example 2<br>Point scatterers | Example 3<br>ANON | Example 4<br>Marmousi | Example 5<br>BP 2004 |
|---|---|---|---|---|---|
| **Category** | | | | | |
| Full Hessian computed? | Yes | Yes | No | No | No |
| No. elements in Hessian | $4.0 \times 10^8$ | $4.0 \times 10^8$ | $4.41 \times 10^{10}$ | $2.04 \times 10^{11}$ | $1.70 \times 10^{13}$ |
| Method to compute Kronecker factors | SVD | SVD | MC | MC | MC |
| No. factors | 64 | 64 | 100 | 100 | 100 |
| Hessian samples (%) | 100 | 100 | 0.31 | 0.07 | 0.002 |
| Bandwidth | - | - | 61 × 61 | 41 × 41 | 31 × 31 |
| **Numerical grid** | | | | | |
| Dimensions (km) | 1.0 x 2.0 | 1.0 x 2.0 | 2.4 x 5.6 | 3.6 x 8.0 | 8.43 x 19.07 |
| Grid | 100 x 200 | 100 x 200 | 300 x 700 | 451 x 1001 | 1350 x 3051 |
| Spacing ($\Delta x = \Delta z$) (m) | 10.0 | 10.0 | 8.0 | 8.0 | 6.25 |
| **Acquisition** | | | | | |
| No. Src/Rec | 29/100 | 29/100 | 99/300 | 100/500 | 120/1526 |
| Src/Rec int. ($\Delta x_s/\Delta x_r$) (m) | 60/20 | 60/20 | 56/16 | 80/16 | 150/12.5 |
| **Simulation** | | | | | |
| Recording time (s) | 2.5 | 2.5 | 3.6 | 5.0 | 6.0 |
| Time interval ($\Delta t$) (ms) | 1.0 | 1.0 | 1.0 | 0.5 | 0.7 |
| Source wavelet | 15 Hz Ricker | 8 Hz Ricker | 5 Hz Ricker | 15 Hz Ricker | 20 Hz Ricker |
| **LSRTM** | | | | | |
| No. CGLS iterations | - | 50 | 50 | 50 | 50 |

Table 6.2: Parameters for numerical tests.

### 6.3.2 KLSRTM-MC

The final three examples present applications of KLSRTM for which the Kronecker factors are estimated using low-rank matrix completion. Elements of the Hessian matrix are computed using Green's functions calculated using a frequency-domain finite-difference solver (Hustedt et al., 2004). The LU decomposition (Davis, 2004) solves the frequency-domain wave equation. I consider frequencies between 1 Hz to 30 Hz at 0.5 Hz intervals; the range of frequencies, for Examples 3-5, is selected based on the frequency band of the source wavelets (Table 6.2). The total number of forward-modeling operations is equal to $N_f \times (N_{rec} + N_{src})$ where $N_f, N_{rec}$ and $N_{src}$ are the number of discrete frequencies, sources and receivers, respectively. A pseudo-Hessian preconditioner is used to accelerate the convergence of the CGLS iterations (Shin et al., 2001). References to window size follow the definition in equation 6.16; a schematic representation is displayed in Figure 6.3. The purpose of the following examples is to demonstrate the gain in performance that is achieved when using a practical implementation of KLSRTM over conventional LSRTM.

#### Example 3

The third test is conducted using the ANON velocity model displayed in Figure 6.10. The Kronecker factors are estimated by solving the matrix completion problem specified in equation 6.19. For the sake of discussion, I define the reduced Hessian $\hat{\mathbf{H}} = \mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a$ (from equation 6.19). The reduced Hessian represents the subset of Hessian elements after enforcing banded structure constraints. In practice, the entire reduced Hessian is not computed; however, it is computed for Figure 6.11 to assess the reconstruction error. Figure 6.11 displays a comparison between parts of the true and reconstructed reduced Hessian. For this 2D model, the size of the reduced Hessian is $17300 \times 41770$. Considering its size, only parts of the reduced Hessian are displayed in the comparisons. Figure 6.11 demonstrates that the reduced Hessian matrix can be recovered via large-scale matrix completion methods. I reiterate that the reduced Hessian itself is not used for KLSRTM; rather, the band Kronecker factors are used.

Figures 6.12a-c depict migration images obtained via RTM, LSRTM and KLSRTM. In all cases, a Laplacian filter is applied to the migration images to remove low-frequency artifacts (Youn and Zhou, 2001). Data-domain LSRTM and image-domain KLSRTM solutions are produced after 50 CGLS iterations. To reiterate the procedure for KLSRTM, I describe the step-by-step process for this example. I first specify window sizes of $61 \times 61$ grid points for any given image point (as demonstrated in Figure 6.3). This limits Hessian samples to select elements clustered around the diagonals of the block elements; the matrix containing all the elements from this subset is the reduced Hessian (Figure 6.3). For the matrix completion

problem, I randomly sample 20% of the *reduced* Hessian (0.31% of the full Hessian). 100 Kronecker products are obtained through the solution of the resultant matrix completion problem (equation 6.19). Qualitatively, the LSRTM and KLSRTM images are virtually indistinguishable from one another. The computational costs for LSRTM and KLSRTM are summarized in Figure 6.13. Sampling of the reduced Hessian constitutes the most compute-intensive component of KLSRTM, requiring 1.3 hours. Estimating the Kronecker factors requires 10 minutes and solving the image-domain LSRTM problem only requires an additional 3 minutes. The fast CG iterations are a result of replacing costly applications of migration/demigration operators, with matrix multiplications involving small matrices.

**Example 4**

Example 4 repeats the tests from Example 3 using the Marmousi velocity model (Martin et al., 2006). The true and smooth velocity models are presented in Figure 6.14. The images obtained via RTM, LSRTM and KLSRTM are displayed in Figures 6.15a-c, respectively. The window size implemented for KLSRTM is $41 \times 41$. As in the previous example, the outputs of each method are essentially identical; however, KLSRTM achieves the result at a significantly reduced computational cost (compared to data-domain LSRTM). A summary of the computational resource usage is displayed in Figure 6.16.

**Example 5**

The final example is performed on the BP 2004 velocity model to explore the scalability of the algorithm to a larger model. The true velocity and density models, presented in Figure 6.17, are used to generate synthetic data. The observations are modeled without a free surface boundary condition therefore multiples are not present in the data. Imaging is performed in a constant density medium with a smoothed version of the true velocity model. The RTM, LSRTM and KLSRTM images are displayed in Figures 6.18a-c, respectively. The window size implemented for KLSRTM is $31 \times 31$. A summary of the computational resource usage is displayed in Figure 6.19. The results are consistent with earlier examples, the speedup is not as significant in this example still good though.

## 6.4 Discussion

The extension of KLSRTM to the 3D case is possible; however, this requires adapting the proposed framework. In 3D, the Hessian matrix is a 3-level block matrix that can be

approximated as a superposition of Kronecker products between three square matrices

$$\mathbf{H} = \sum_{i=1}^{k} \mathbf{A}_i \otimes \mathbf{B}_i \otimes \mathbf{C}_i, \tag{6.20}$$

where $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{C}_i$ are Kronecker factors of size $n_z \times n_z$, $n_x \times n_x$ and $n_y \times n_y$, respectively. To compute the Kronecker factors, the Hessian $\mathbf{H}$ needs to be rearranged into a third-order tensor of size $n_z^2 \times n_x^2 \times n_y^2$. The Kronecker factors would be obtained by solving a tensor completion problem, using samples of the Hessian, to find the optimal Canonical-Parafac decomposition (Liu et al., 2013). The tensor analog to equation 6.10 leads to the 3D form of image-domain KLSRTM:

$$\underset{\mathcal{M}}{\operatorname{argmin}} \| \sum_{i=1}^{k} \mathcal{M} \times_1 \mathbf{C}_i \times_2 \mathbf{B}_i \times_3 \mathbf{A}_i - \mathcal{M}^{mig} \|_F^2, \tag{6.21}$$

where $\mathcal{M}$ is the discretized 3D model represented as a tensor and $\mathcal{M}^{mig}$ is the tensor-form of the migration image volume. The interaction between the model tensor and the Kronecker factors is in the form of tensor-matrix multiplications (Kolda and Bader, 2009b). In this study, I compute source and receiver Green's functions in the frequency domain; however, this poses a memory challenge in 3D applications (Operto et al., 2007). For 3D, a more viable strategy would be to compute the Green's functions in the time domain. Computing samples of the Hessian in the time domain requires storing the Green's function volumes for all time steps. The storage cost can be reduced by saving a selection of discrete frequency representations of the wavefield using the discrete Fourier transform (Nihei and Li, 2007; Sirgue et al., 2008). The use of source and receiver encoding during the computation of the Green's functions can further reduce computational cost (Tang, 2009).

Elastic forms of KLSRTM are readily achievable after replacing equation 6.14 with an analogous expression for the multi-parameter Hessian (in terms of source and receiver Green's functions e.g., Chen et al. (2007)). In 2D, the multi-parameter Hessian has dimensions $n_p \cdot n_x \cdot n_z \times n_p \cdot n_x \cdot n_z$, where $n_p$ denotes the number of independent physical parameters. The increased size may necessitate computing a greater proportion of Hessian samples to suitably solve equation 6.19.

Sparsity promotion is a common strategy used to suppress migration artifacts and improve image quality. Sparsity-constrained optimization problems can be solved via iterative-reweighted least-squares (IRLS) or the iterative-thresholding algorithm (Daubechies et al., 2010, 2004). Compared to the original non-constrained least-squares problem, the number of iterations required for solving sparsity-constrained imaging problem is much higher. For conventional LSRTM, the computational cost becomes formidable as the number of iterations increases. In KLSRTM, sparsity-promoting regularization is more accessible due to

the fast CG iterations. In a similar vain, the fast CG iterations facilitate more efficient testing of regularization hyperparameters in regularized forms of KLSRTM.

For large 3D imaging problems, the proposed algorithm requires a large amount of memory to save even a small portion of the Hessian samples. A potential solution that alleviates the memory burden, would be to formulate the imaging problem as a target-oriented inversion problem (e.g., Valenciano et al. (2006); Tang (2009)). Another limitation of the proposed algorithm stems from the block-band structure assumed for the Hessian matrix. If the action of the Hessian on a point scatterer is not local i.e. if significant contributions occur outside a small window about the scattering point, then the bandwidth of the Kronecker factors must increase. This increases the proportion of Hessian samples that need to be considered in the matrix-completion stage and increases the computational cost. This situation may occur if the data is sparsely sampled or the frequency band of the data is narrow.

The Kronecker-based Hessian approximation can be incorporated into full waveform inversion to accelerate convergence. In the simplest use case, the diagonal of the approximated Hessian can be used as a preconditioner for first-order, gradient-based optimization methods. Alternatively, the Hessian approximation can be used in inexact second-order optimization schemes such as the Truncated-Newton method (Métivier et al., 2013). Truncated Newton methods iteratively solve a linear system of equations, similar to the image-domain LSRTM problem, to estimate the FWI update direction at each iteration. The internal solution of the linear system is expensive due to added wave equation solves associated with computing Hessian-vector products. In principle, the Kronecker-based Hessian approximation could be used for fast Hessian-vector products. Unlike LSRTM, in FWI the velocity model updates at each iteration thus changing the Hessian. Recomputing the Kronecker-based Hessian approximation at each iteration makes the approach less attractive. Heuristic strategies that only estimate the Hessian every $n$ iterations may be beneficial; however, I have not explored them. Source-encoding strategies (e.g. Anagaw and Sacchi (2014); Castellanos et al. (2015)) or stochastic second-order optimization strategies (Matharu and Sacchi, 2019) provide more flexible approaches to second-order optimization in FWI.

## 6.5 Conclusion

We propose a fast method for image-domain LSRTM based on approximating the Hessian matrix as a superposition of Kronecker products. I first demonstrate how to estimate Kronecker factors through an SVD of the rearranged Hessian for cases when the full Hessian is known. For more practical implementations, I devise an efficient scheme to estimate Kronecker factors without explicitly forming the Hessian. The approach preferentially samples elements of the Hessian and solves a matrix completion problem to estimate the Kronecker

factors. Hessian samples are computed using source and receiver Green's functions computed in the frequency domain. A series of numerical examples validate various stages of the proposed algorithm. Ultimately, I observe the KLSRTM yields near identical images to LSRTM, but at substantially reduced computational cost. The bulk of the computational cost associated with KLSRTM arises from the computation of Hessian samples. Solution of the subsequent matrix completion problem along with the iterative solution of the LSRTM problem are comparatively fast. Using the Kronecker-based approximation, I replace expensive migration/demigration operations with small matrix multiplications resulting in fast CG iterations. While sampling of the Hessian results in increased memory requirements for KLSRTM, the increases are manageable for typical 2D problems. The extension of KLSRTM to 3D or elastic applications is, in principle, possible with some modifications to the algorithm; however, this is a topic for future research. The primary challenge in 3D extensions is the memory cost associated with computing Hessian samples.

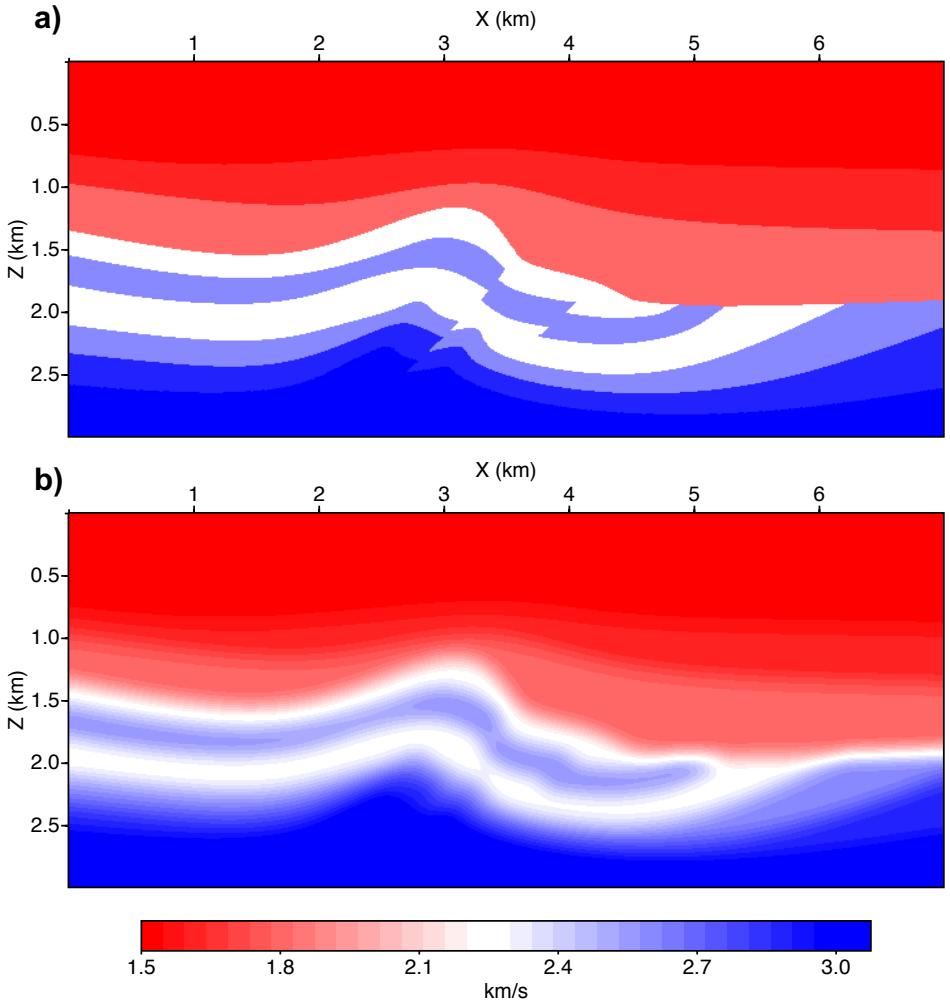


Figure 6.10: ANON velocity model (a) True velocity model. (b) Smooth velocity model used for imaging.

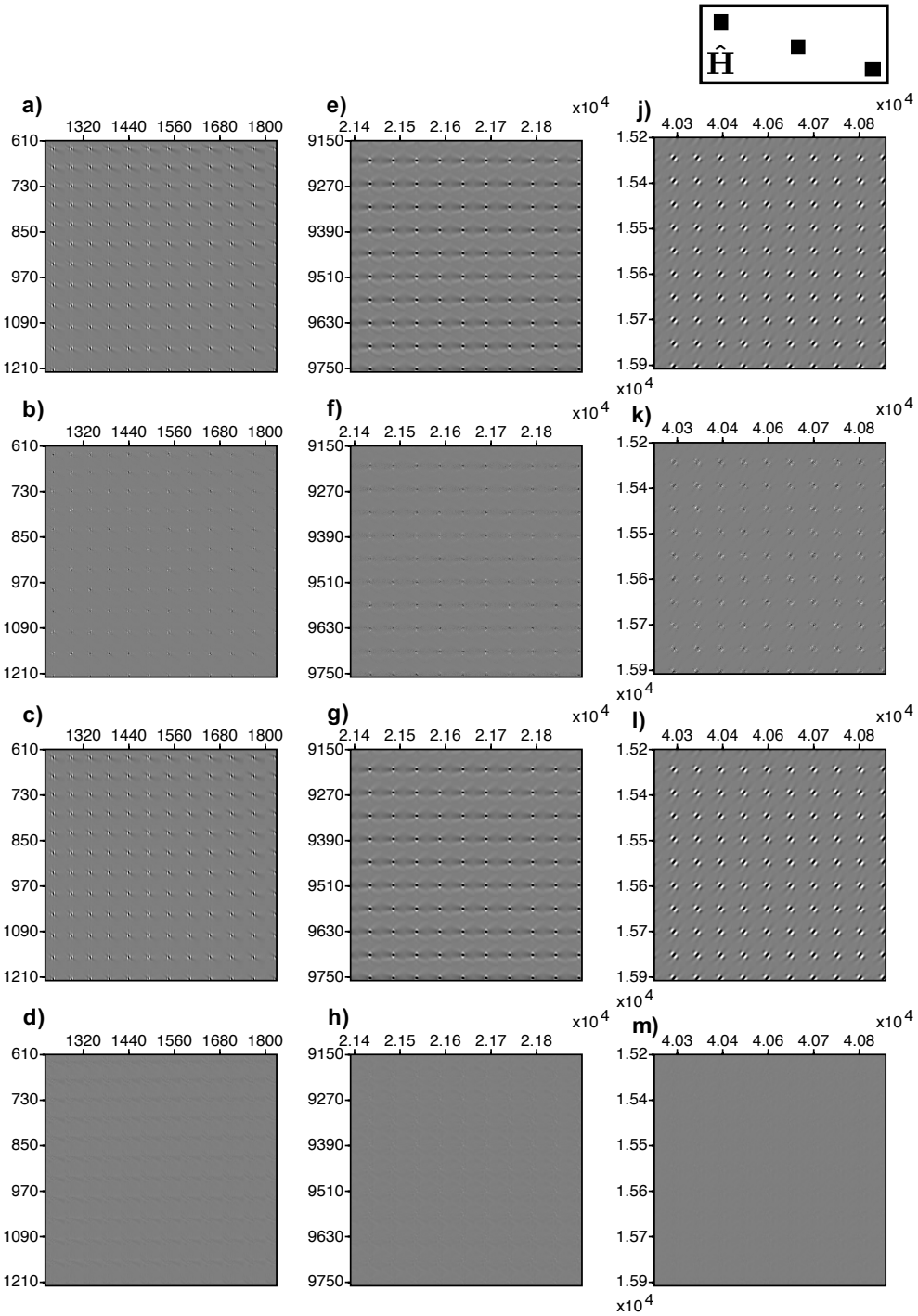


Figure 6.11: A comparison of the (a, e, j) true reduced Hessian, (b, f, k) randomly subsampled reduced Hessian (20% of full Hessian), (c, g, l) restored reduced Hessian and (d, h, m) the reconstruction error for three distinct windows. Locations of the Hessian windows are displayed in the schematic diagram (top right).

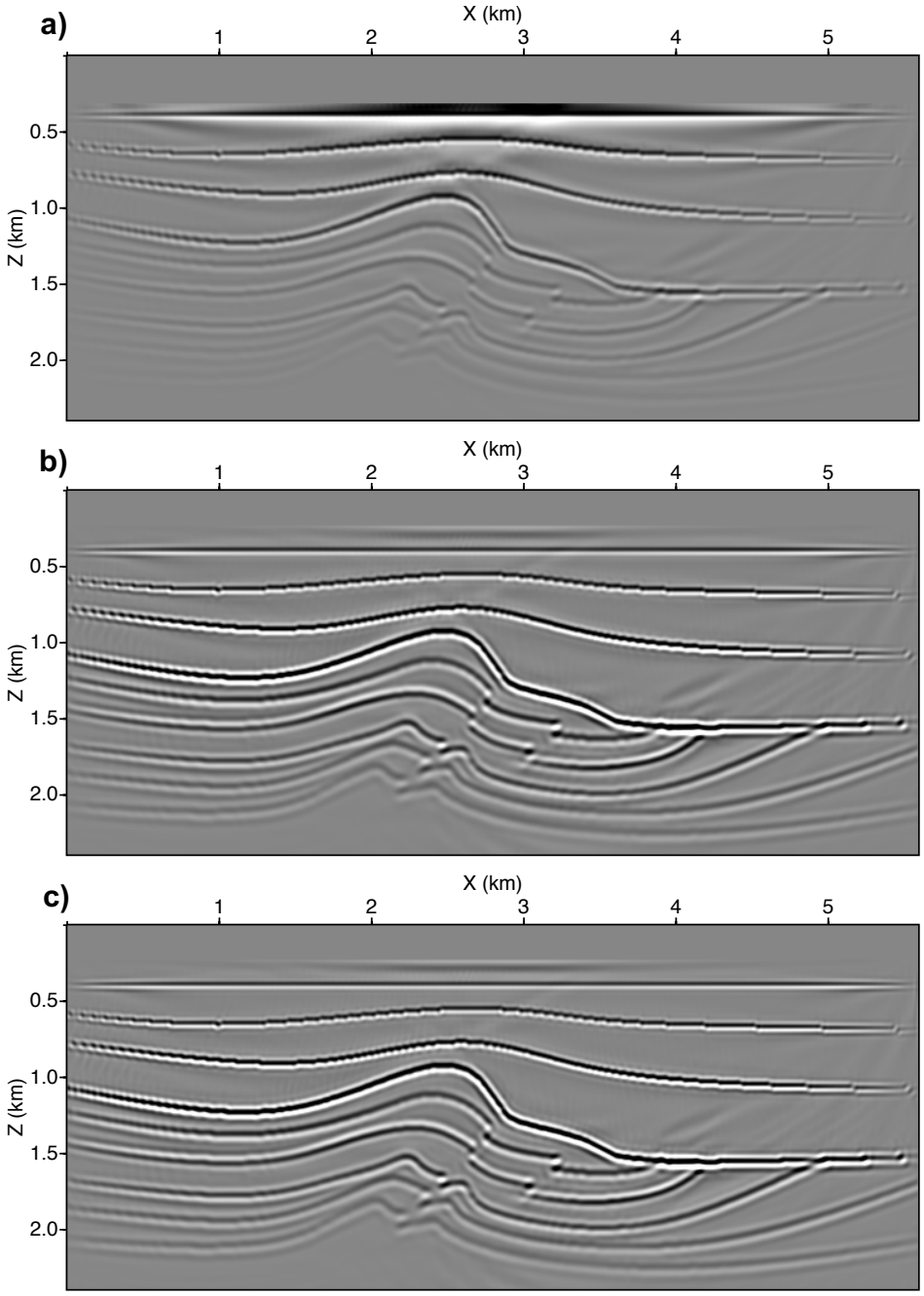


Figure 6.12: Migration results for the ANON velocity model. The images are computed using (a) RTM, (b) LSRTM, and (c) KLSRTM. For (b) and (c), 50 CGLS iterations are performed.

| 2D 300-by-700 acoustic model, 50 iterations, 100 factors, 48 cores | | | |
|---|---|---|---|
| | | **LSRTM** | **KLSRTM** |
| **Memory** | | 25.3 M | 394.5 M |
| **CPU time (Hours)** | Hessian | - | 1.3 |
| | Kronecker | - | 0.16 |
| | CGLS | 23.5 | 0.05 |
| | *Total* | 23.5 | 1.51 |

Figure 6.13: A comparison of the memory and computational requirements for LSRTM and KLSRTM in the ANON velocity model.

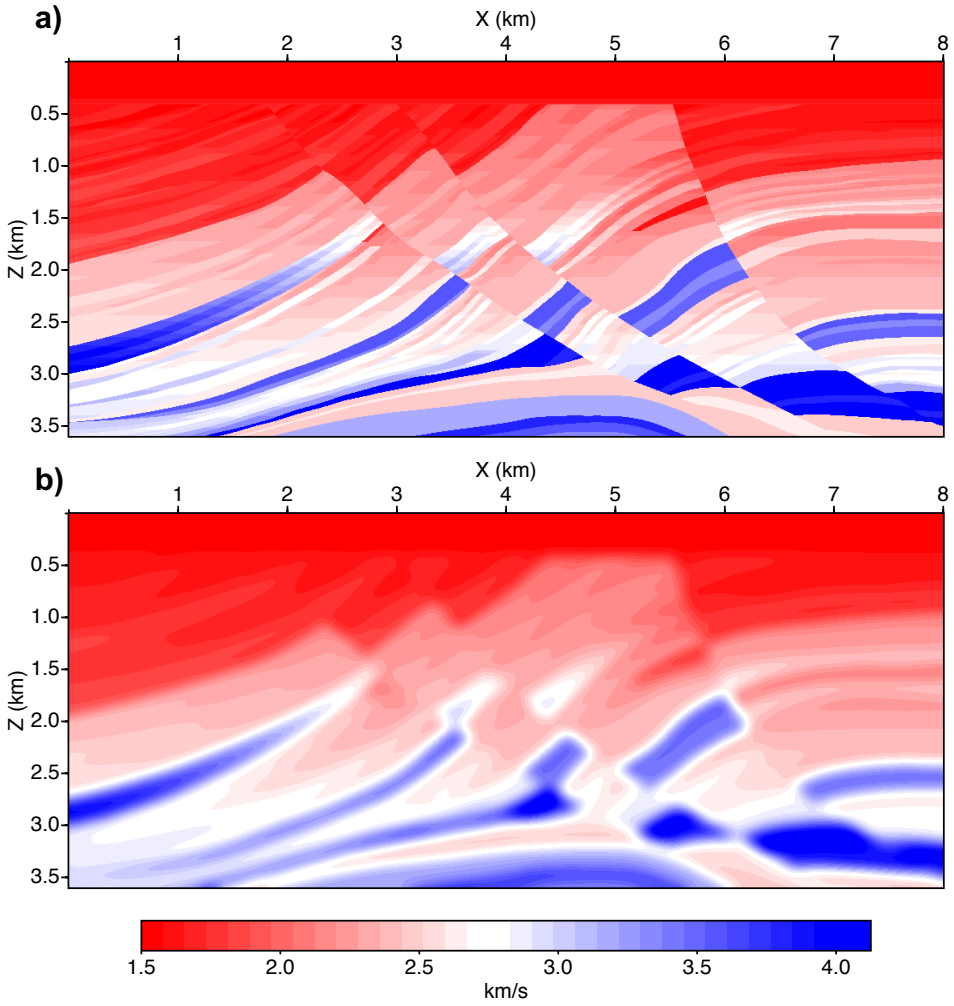


Figure 6.14: Marmousi velocity model. (a) True velocity model. b) Smoothed velocity model used for imaging.

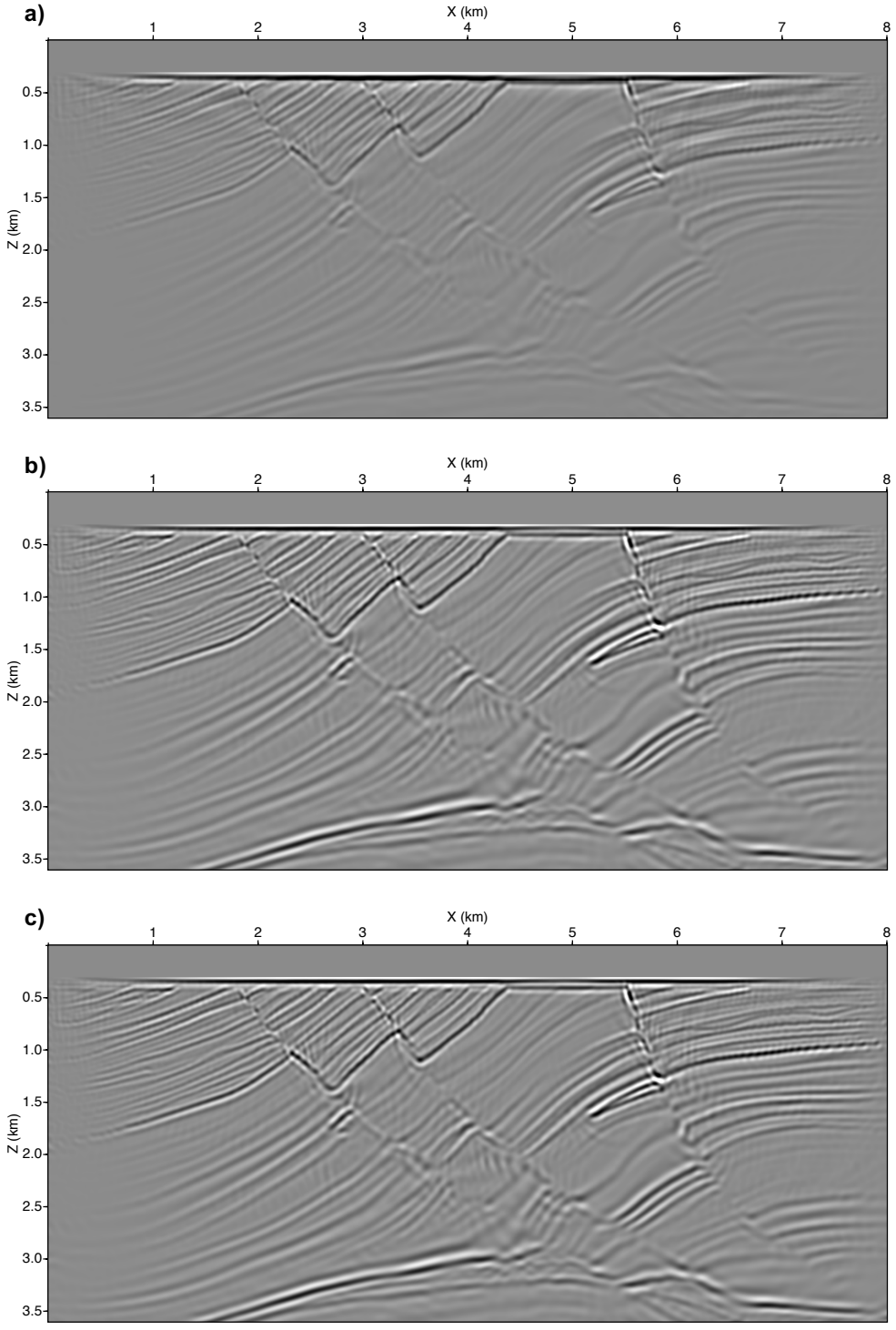


Figure 6.15: Migration results for the Marmousi velocity model. Migration images computed using (a) RTM, (b) LSRTM, and (c) KLSRTM. The images are obtained after 50 CGLS iterations in either case.

| 2D 451-by-1001 acoustic model, 50 iterations, 100 factors, 48 cores | | | |
|---|---|---|---|
| | | **LSRTM** | **KLSRTM** |
| **Memory** | | 53.8 M | 848.1 M |
| **CPU time (Hours)** | Hessian | - | 4.3 h |
| | Kronecker | - | 0.5 |
| | CGLS | 37.5 | 0.1 |
| | *Total* | 37.5 | 4.9 |

Figure 6.16: A comparison of the memory and computational requirements for LSRTM and KLSRTM in the Marmousi velocity model.

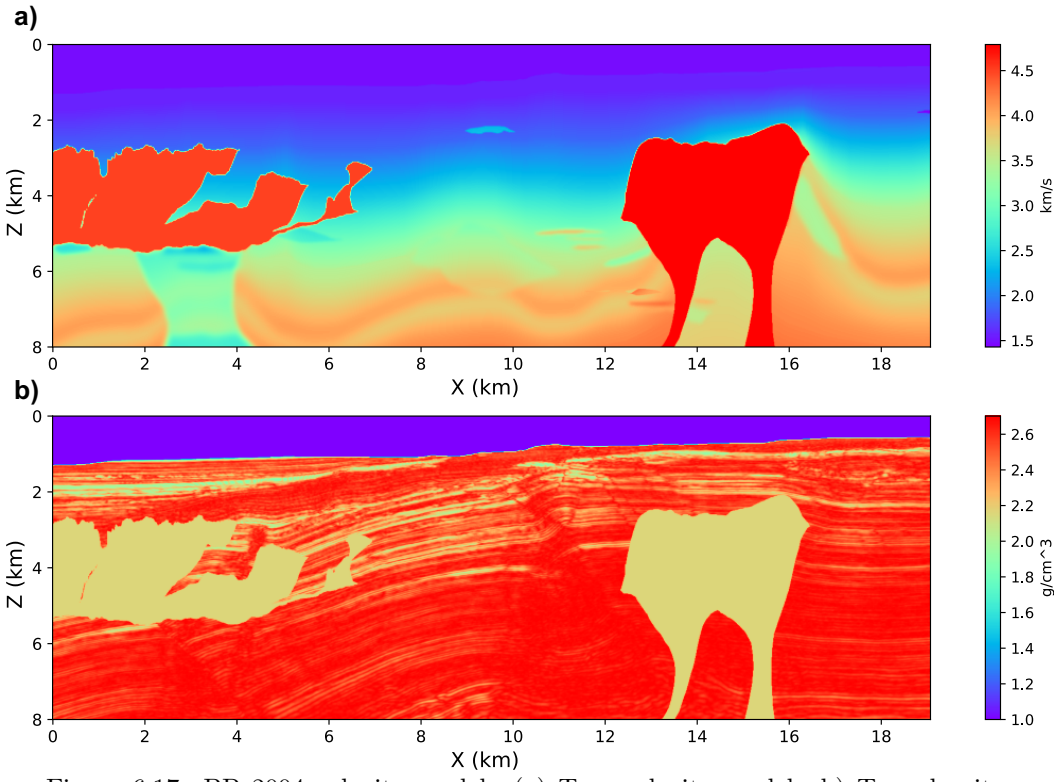


Figure 6.17: BP 2004 velocity model. (a) True velocity model. b) True density model.

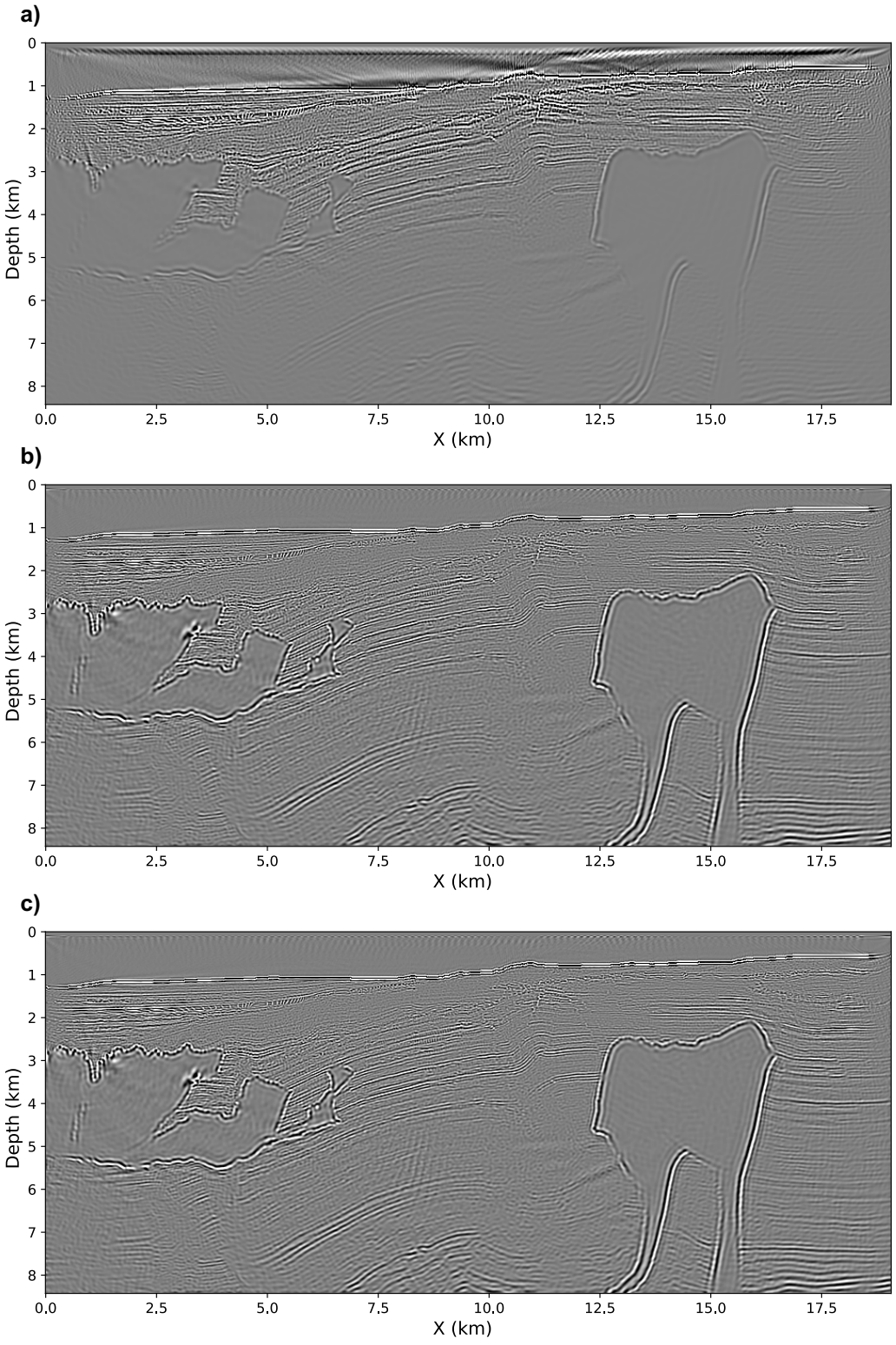


Figure 6.18: Migration results for the BP 2004 velocity model. Migration images computed using (a) RTM, (b) LSRTM and (c) KLSRTM. The images are obtained after 50 CGLS iterations in either case.

| 2D 1350-by-3051 acoustic model, 50 iterations, 100 factors, 48 cores | | | |
|---|---|---|---|
| | | **LSRTM** | **KLSRTM** |
| **Memory** | | 61.3 M | 2.9 G |
| **CPU time (Hours)** | Hessian | - | 18.0 |
| | Kronecker | - | 1.2 |
| | CGLS | 91.6 | 0.2 |
| | *Total* | 91.6 | 19.4 |

Figure 6.19: A comparison of the memory and computational requirements for LSRTM and KLSRTM in the BP 2004 velocity model.

# CHAPTER 7

# Conclusions

Multicomponent seismic surveys have gained significant attention and demonstrated substantial growth in the last decade. They contribute supplementary information to help reduce drilling risk when, for instance, assessing new plays. However, PS-wave data often suffer from considerable noise contamination and the absence of high-frequency components. Another challenge of a multicomponent seismic survey is associated to the interpretation of PP-wave and PS-wave volumes. The time migrated PS-wave volume bear little resemblance to the corresponding PP-wave volume due to velocity differences between P and S waves. The registration process, which aligns PS-wave events to their PP-wave events, is largely driven by the availability of dipole sonic logs. These dipole sonic are not as common as standard sonic due to their high costs. In this thesis, I first develop an efficient method based on a novel tensor decomposition algorithm to attenuate the random noise of both PP-wave and PS-wave data in the pre-stack domain. Secondly, I introduce a robust registration method to match PP-wave and PS-wave events without the need of a known $v_p/v_s$ velocity ratio.

My work also contributes to imaging. I have developed an efficient LSRTM method by compressing the Hessian matrix as the superposition of Kronecker products. The proposed method has the potential to be extended to elastic cases and realistic 3D survey.

In Chapter 1, I review the topics of multicomponent seismic data acquisition, processing and interpretation. Afterwards, I point out the challenges associated with converted-wave data.

In Chapter 2, I discuss the notations of multi-dimensional tensor algebra utilized in this thesis and briefly review three low-rank tensor decomposition algorithms commonly used in the seismic data processing.

In Chapter 3, I demonstrate that tensor algebra can provide a robust framework for multi-

dimensional seismic data processing. A low-rank tensor can represent a noise-free seismic data volume. An additive random noise will increase the rank of the tensor. Hence, tensor rank-reduction techniques can be used to filter random noise. My filtering method adopts the Candecomp / Parafac (CP) decomposition to approximates an N-dimensional seismic data volume as the superposition of rank-one tensors. Similar to the Singular Value Decomposition (SVD) for matrices, a low-rank CP decomposition can capture the signal and exclude random noise in situations where a low-rank tensor can represent the ideal noise-free seismic volume. The alternating least squares (CP-ALS) method is adopted to compute the CP decomposition with a provided target rank. This method involves solving a series of highly over-determined linear least-squares subproblems. To improve the efficiency of the CP-ALS algorithm, I uniformly randomly sample equations of the linear least-squares subproblems to reduce the size of the problem significantly. The computational overhead is further reduced by avoiding unfolding and folding large dense tensors. I investigate the applicability of the randomized CP decomposition (RAND-CP-ALS) for incoherent noise attenuation via experiments conducted on a synthetic dataset and field data seismic volumes. I also compare the proposed algorithm (RAND-CP-ALS) against multi-dimensional singular spectrum analysis (MSSA) and classical $f-xy$ prediction filtering. I conclude the proposed approach can achieve slightly better denoising performance in terms of signal-to-noise ratio enhancement than traditional methods, but with a less computational cost. The proposed algorithm in this chapter is implemented in $t$-$\mathbf{x}$ domain. In the future, the method can be extended to $f$-$\mathbf{x}$ domain, which enjoy particular advantages over $t$-$\mathbf{x}$ domain. For example, 1) I can apply 4D tensor decomposition to each frequency component of 5D pre-stack seismic data independently. It is convenient for designing high-performance algorithms which are parallel over frequencies. 2) The low-rank properties of linear and parabolic events are preserved naturally in the frequency domain as their time delay can be represented as the outer product of phase-shift terms in the frequency domain.

In Chapter 4, I formulate the problem of multicomponent seismic data interpolation in the framework of low-rank tensor completion. Specifically, high-order SVD is implemented independently to NMO-corrected PP-wave and PS-wave to recover missing traces and attenuate random noise. Windowing strategy is utilized to enhance the performance of low-rank completion algorithm. The effectiveness of the proposed method is demonstrated on a dataset from West-Canadian sedimentary basin.

In Chapter 5, I map PS-wave data to PP-wave time domain, which is a critical step before joint PP-wave and PS-wave data interpretation and inversion, by minimizing the difference between PP-wave and warped PS-wave data. 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. To avoid undesirable foldings or rapid changes in the warped PS-wave image, I require a warping function that is monotonic and smooth. I propose to invert the $V_p/V_s$ ratio directly from PP and PS-wave data instead of estimating it from the warping function. Seismic data registration is posed as a constrained non-linear optimization problem. Furthermore, I propose to represent the $V_p/V_s$ ratio by spline functions and to adopt a parameterization that guarantees monotonic warping functions. My parameterization in terms of splines significantly reduces the number of unknowns of my problem and the convergence to a smooth monotonic solution is guaranteed. The proposed registration method requires a close initial solution as the starting point of the non-linear optimization solver. I can estimate the initial solution with dynamic image warping method (Compton and Hale, 2014) and refine the registration result iteratively with the proposed method. One difficulty hinders the application of joint pre-stack inversion is the relatively narrow band of registered PS-wave data with respect to PP-wave data. In the future, I will explore a collaborative deconvolution method to enhances the resolution of PS-wave data.

In Chapter 6, I propose an efficient least-squares reverse-time migration (LSRTM) method, which has become increasingly popular for complex wavefield imaging due to its ability to equalize image amplitudes, attenuate migration artifacts, handle incomplete and noisy data, and to improve spatial resolution. The major drawback of LSRTM is the considerable computational cost incurred by performing migration/demigration at each iteration of the optimization. To ameliorate the computational cost, I introduce a fast method to solve the LSRTM problem in the image domain. The proposed method is based on a new factorization that approximates the Hessian using a superposition of Kronecker products. The Kronecker factors are small matrices relative to the size of the Hessian. Crucially, the factorization can honour the characteristic block-band structure of the Hessian. I introduce a computationally efficient algorithm to estimate the Kronecker factors via low-rank matrix completion. The completion algorithm utilizes only a small percentage of preferentially sampled elements of the Hessian matrix. Element sampling requires the computation of source and receiver Green's functions but avoids explicitly constructing the entire Hessian. The proposed Kronecker-based factorization leads to an imaging technique that I name Kronecker-LSRTM (KLSRTM). The iterative solution of the image-domain KLSRTM is fast because I replace computationally expensive migration/demigration operations, with fast matrix multiplications involving small matrices. I first validate the efficacy of the proposed method by explicitly computing the Hessian for a small problem. Subsequent 2D numerical tests compare LSRTM with KLSRTM for several benchmark models. I observe that KLSRTM achieves near-identical images to LSRTM at a significantly reduced computational cost ($\sim$5-15$\times$ faster); however, KLSRTM has an increased, yet manageable, memory cost. Future research can extend the proposed method to a more practical 3D LSRTM problem in which the Kronecker factors are estimated using low-rank tensor completion (Kolda

and Bader, 2009b). The main challenge of 3D KLSRTM is the high memory cost associated with saving the tremendous amount of samples of the Hessian matrix. Researchers need to design algorithms to reduce the memory cost. One way to achieve this goal is to estimate Kronecker factors on-the-fly, which does not require the storing of Hessian samples. For example, the Green's function can be easily obtained based on the known travel time table in least-squares Kirchhoff migration, so as the Hessian samples. At last, I would like to emphasize that the least-squares reverse time migration, which is based on the two-way wave equation, is used as an example in this chapter. However, the proposed factorization method is not limited to the Hessian matrix associated with this particular case. Instead, it can be extended to other least-squares migration methods, such as one-way wave equation migration and Kirchhoff migration.

# Bibliography


Abma, R., and J. Claerbout, 1995, Lateral prediction for noise attenuation by t-x and f-x techniques: Geophysics, **60**, 1887–1896.

Akaike, H., 1969, Fitting autoregressive models for prediction: Annals of the Institute of Statistical Mathematics, **21**, 243–247.

Anagaw, A. Y., and M. D. Sacchi, 2014, Comparison of multifrequency selection strategies for simultaneous-source full-waveform inversion: Geophysics, **79**, no. 5, R165–R181.

Avseth, P., T. Mukerji, and G. Mavko, 2010, Quantitative seismic interpretation: Applying rock physics tools to reduce interpretation risk: Cambridge university press.

Barkved, O., B. Bartman, J. Gaiser, R. Van Dok, T. Johns, P. Kristiansen, T. Probert, and M. Thompson, 2004, The many facets of multicomponent seismic data: Oilfield Review, **16**, no. 2, 42–56.

Battaglino, C., G. Ballard, and T. G. Kolda, 2018, A practical randomized CP tensor decomposition: SIAM Journal on Matrix Analysis and Applications, **39**, 876–901.

Baysal, E., D. D. Kosloff, and J. W. Sherwood, 1983, Reverse time migration: Geophysics, **48**, no. 11, 1514–1524.

Berkhout, A., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, **27**, 924–938.

Bourgeois, A., B. Jiang, and P. Lailly, 1989, Linearized inversion: a significant step beyond pre-stack migration: Geophysical Journal International, **99**, 435–445.

Buur, J., and T. Kühnel, 2008, Salt interpretation enabled by reverse-time migration: Geophysics, **73**, no. 5, VE211–VE216.

Cai, J.-F., E. J. Candès, and Z. Shen, 2010, A singular value thresholding algorithm for matrix completion: SIAM Journal on Optimization, **20**, 1956–1982.

Canales, L. L., 1984, Random noise reduction: 54th SEG Annual Meeting, Atlanta, Georgia, USA, Expanded Abstracts, 525–527.

Candès, E. J., and B. Recht, 2009, Exact matrix completion via convex optimization: Foundations of Computational mathematics, **9**, 717–772.

Candès, E. J., and T. Tao, 2010, The power of convex relaxation: Near-optimal matrix

completion: IEEE Transactions on Information Theory, **56**, 2053–2080.

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

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.

Cattell, R. B., 1944, parallel proportional profiles and other principles for determining the choice of factors by rotation: Psychometrika, **9**, 267–283.

Chang, W.-F., and G. A. McMechan, 1986, Reverse-time migration of offset vertical seismic profiling data using the excitation-time imaging condition: Geophysics, **51**, 67–84.

Chase, M. K., 1992, Random noise reduction by FXY prediction filtering: Exploration Geophysics, **23**, 51–55.

Chavent, G., and R.-E. Plessix, 1999, An optimal true-amplitude least-squares prestack depth-migration operator: Geophysics, **64**, 508–515.

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.

Cheng, J., N. Kazemi, and M. Sacchi, 2016, Least-squares migration via a gradient projection method-application to seismic data deblending: 78th Annual International Conference and Exhibition, EAGE, Extended Abstracts, Th P4 07.

Chung, W., and D. Corrigan, 1985, Gathering mode-converted shear waves: A model study, *in* SEG Technical Program Expanded Abstracts 1985: Society of Exploration Geophysicists, 602–604.

Clapp, M. L., 2005, Imaging under salt: Illumination compensation by regularized inversion: PhD thesis, Stanford University.

Comon, P., X. Luciani, and A. L. De Almeida, 2009, Tensor decompositions, alternating least squares and other tales: Journal of Chemometrics, **23**, 393–405.

Compton, S., and D. Hale, 2014, Estimating vp/vs ratios using smooth dynamic image warping: Geophysics, **79**, no. 6, V201–V215.

Dai, W., P. Fowler, and G. T. Schuster, 2012, Multi-source least-squares reverse time migration: Geophysical Prospecting, **60**, 681–695.

Daubechies, I., M. Defrise, and C. De Mol, 2004, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint: Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, **57**, 1413–1457.

Daubechies, I., R. DeVore, M. Fornasier, and C. S. Güntürk, 2010, Iteratively reweighted least squares minimization for sparse recovery: Communications on Pure and Applied

Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, **63**, 1–38.

Davis, T. A., 2004, Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method: ACM Transactions on Mathematical Software (TOMS), **30**, 196–199.

De Lathauwer, L., and B. De Moor, 1998, From matrix to tensor: Multilinear algebra and signal processing: Institute of Mathematics and Its Applications Conference Series, Citeseer, 1–16.

De Lathauwer, L., B. De Moor, and J. Vandewalle, 2000, A multilinear singular value decomposition: SIAM journal on Matrix Analysis and Applications, **21**, 1253–1278.

De Lathauwer, L., and J. Vandewalle, 2004, Dimensionality reduction in higher-order signal processing and rank-(r1, r2,, rn) reduction in multilinear algebra: Linear Algebra and its Applications, **391**, 31–55.

Dong, S., J. Cai, M. Guo, S. Suh, Z. Zhang, B. Wang, and e. Z. Li, 2012, Least-squares reverse time migration: Towards true amplitude imaging and improving the resolution, *in* SEG Technical Program Expanded Abstracts 2012: Society of Exploration Geophysicists, 1–5.

Dussaud, E., W. W. Symes, P. Williamson, L. Lemaistre, P. Singer, B. Denel, and A. Cherrett, 2008, Computational strategies for reverse-time migration, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 2267–2271.

Elboth, T., I. V. Presterud, and D. Hermansen, 2010, Time-frequency seismic data denoising: Geophysical Prospecting, **58**, 441–453.

Etgen, J., S. H. Gray, and Y. Zhang, 2009, An overview of depth imaging in exploration geophysics: Geophysics, **74**, no. 6, WCA5–WCA17.

Farfour, M., and W. J. Yoon, 2016, A review on multicomponent seismology: A potential seismic application for reservoir characterization: Journal of advanced research, **7**, 515–524.

Fletcher, R. P., D. Nichols, R. Bloor, and R. T. Coates, 2016, Least-squares migrationData domain versus image domain using point spread functions: The Leading Edge, **35**, 157–162.

Fomel, S., 2007, Local seismic attributes: Geophysics, **72**, no. 3, A29–A33.

Fomel, S., M. Backus, K. Fouad, B. Hardage, and G. Winters, 2005, A multistep approach to multicomponent seismic image registration with application to a west texas carbonate reservoir study: SEG Technical Program Expanded Abstracts, 1018–1021.

Fomel, S., and M. M. Backus, 2003, Multicomponent seismic data registration by least squares: SEG Technical Program Expanded Abstracts, 781–784.

Gaiser, J., 1999, Applications for vector coordinate systems of 3-d converted-wave data: The Leading Edge, **18**, 1290–1300.

Gaiser, J. E., 1996, Multicomponent VP/VS correlation analysis: Geophysics, **61**, no. 4,

1137–1149.

Gandy, S., B. Recht, and I. Yamada, 2011, Tensor completion and low-n-rank tensor recovery via convex optimization: Inverse Problems, **27**, 025010.

Gao, J., J. Cheng, and M. D. Sacchi, 2015, A new 5D seismic reconstruction method based on a parallel square matrix factorization algorithm: 85th SEG Annual Meeting, New Orleans, Louisiana, USA, Expanded Abstracts, 3784–3788.

Gao, J., M. D. Sacchi, and X. Chen, 2011, A fast rank reduction method for the reconstruction of 5D seismic volumes: 81st SEG Annual Meeting, San Antonio, Texas, USA, Expanded Abstracts, 3622–3627.

———, 2013, A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions: Geophysics, **78**, V21–V30.

Gao, W., and M. Sacchi, 2016, Collaborative deconvolution of PS-wave data: Part 2: SEG Technical Program Expanded Abstracts, 2279–2283.

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**, no. 6, 660–669.

Granli, J. R., B. Arntsen, A. Sollid, and E. Hilde, 1999, Imaging through gas-filled sediments using marine shear-wave data: Geophysics, **64**, 668–677.

Guitton, A., 2004, Amplitude and kinematic corrections of migrated images for nonunitary imaging operators: Geophysics, **69**, 1017–1024.

Guliyev, E., and T. L. Davis, 2007, Interpretation of Vp/Vs velocity ratio for improved tight-gas sandstone reservoir characterization, Rulison field, Colorado: SEG Technical Program Expanded Abstracts, 1451–1455.

Gülünay, N., 1986, FXDECON and complex Wiener prediction filter: 86th SEG Annual Meeting, Houston, Texas, USA, Expanded Abstracts, 279–281.

———, 2000, Noncausal spatial prediction filtering for random noise reduction on 3-D post-stack data: Gepphysics, **65**, 1641–1653.

Hale, D., 2009, A method for estimating apparent displacement vectors from time-lapse seismic images: Geophysics, **74**, no. 5, V99–V107.

———, 2013, Dynamic warping of seismic images: Geophysics, **78**, S105–S115.

Harshman, R. A., 1970, Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis: UCLA Workshop Papers in Phonetics, **16**.

He, Y., G. Hilburn, F. Sherrill, et al., 2015, Joint pp/ps tomography with dynamic warping: Presented at the 2015 SEG Annual Meeting, Society of Exploration Geophysicists.

Henrion, R., 1993, Body diagonalization of core matrices in three-way principal components analysis: Theoretical bounds and simulation: Journal of chemometrics, **7**, 477–494.

Herrmann, F. J., D. Wang, G. Hennenfent, and P. P. Moghaddam, 2007, Curvelet-based

seismic data processing: A multiscale and nonlinear approach: Geophysics, **73 (1)**, A1–A5.

Hill, D. L., P. G. Batchelor, M. Holden, and D. J. Hawkes, 2001, Medical image registration: Physics in medicine and biology, **46**, no. 3, R1.

Hitchcock, F. L., 1927, The expression of a tensor or a polyadic as a sum of products: Studies in Applied Mathematics, **6**, 164–189.

Hou, S., F. Zhang, and X. Li, 2016, Multicomponent seismic reconstruction via modified inner product operator and weighted k-svd, *in* SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 4087–4091.

Hu, J., G. T. Schuster, and P. A. Valasek, 2001, Poststack migration deconvolution: Geophysics, **66**, 939–952.

Hustedt, B., S. Operto, and J. Virieux, 2004, Mixed-grid and staggered-grid finite-difference methods for frequency-domain acoustic wave modelling: Geophysical Journal International, **157**, 1269–1296.

Ibrahim, A., and M. D. Sacchi, 2013, Simultaneous source separation using a robust Radon transform: Geophysics, **79 (1)**, V1–V11.

Jaggi, M., and M. Sulovskỳ, 2010, A simple algorithm for nuclear norm regularized problems: Proceedings of the 27th international conference on machine learning (ICML-10), 471–478.

Jain, P., P. Netrapalli, and S. Sanghavi, 2013, Low-rank matrix completion using alternating minimization: Proceedings of the 45th annual ACM symposium on Theory of computing, ACM, 665–674.

Kaplan, S. T., P. S. Routh, and M. D. Sacchi, 2010, Derivation of forward and adjoint operators for least-squares shot-profile split-step migration: Geophysics, **75**, no. 6, S225–S235.

Klein, S., M. Staring, K. Murphy, M. Viergever, J. P. Pluim, et al., 2010, Elastix: a toolbox for intensity-based medical image registration: IEEE Transactions on Medical Imaging, **29**, no. 1, 196–205.

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

———, 2009b, 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 GeoConvention, Calgary, Canada.

Kreimer, N., and M. D. Sacchi, 2012a, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77 (3)**, V113–V122.

———, 2012b, 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.

Kroonenberg, P. M., and J. De Leeuw, 1980, Principal component analysis of three-mode data by means of alternating least squares algorithms: Psychometrika, **45**, 69–97.

Kühl, H., and M. D. Sacchi, 2003, Least-squares wave-equation migration for AVP/AVA inversion: Geophysics, **68**, 262–273.

Lailly, P., and J. Bednar, 1983, The seismic inverse problem as a sequence of before stack migrations: Conference on inverse scattering: theory and application, Siam Philadelphia, PA, 206–220.

Lambaré, G., J. Virieux, R. Madariaga, and S. Jin, 1992, Iterative asymptotic inversion in the acoustic approximation: Geophysics, **57**, 1138–1154.

Levin, S. A., 1984, Principle of reverse-time migration: Geophysics, **49**, 581–583.

Liner, C. L., and R. G. Clapp, 2004, Nonlinear pairwise alignment of seismic traces: The Leading Edge, **23**, no. 11, 1146–1150.

Lines, L., Y. Zou, A. Zhang, K. Hall, J. Embleton, B. Palmiere, C. Reine, P. Bessette, P. Cary, and D. Secord, 2005, Vp/Vs characterization of a heavy-oil reservoir: The Leading Edge, , no. 24, 1134–1136.

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

Liu, J., P. Musialski, P. Wonka, and J. Ye, 2013, Tensor completion for estimating missing values in visual data: IEEE transactions on pattern analysis and machine intelligence, **35**, 208–220.

Liu, N., B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien, 2005, Text representation: From vector to tensor: Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE, 4–pp.

Lu, J., Z. Yang, Y. Wang, and Y. Shi, 2015, Joint PP and PS AVA seismic inversion using exact Zoeppritz equations: Geophysics, **80**, no. 5, R239–R250.

Lubin, M., and I. Dunning, 2015, Computing in operations research using Julia: INFORMS Journal on Computing, **27**, 238–248.

Maintz, J. A., and M. A. Viergever, 1998, A survey of medical image registration: Medical image analysis, **2**, no. 1, 1–36.

Margrave, G. F., 2007, Methods of seismic data processing: Course Notes for Geophysics 557, University of Calgary.

Martin, G. S., K. J. Marfurt, and S. Larsen, 2002, Marmousi-2: An updated model for the investigation of AVO in structurally complex areas: SEG Technical Program Expanded Abstracts, 1979–1982.

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, 2019, A subsampled truncated-newton method for multiparameter full-waveform inversion: Geophysics, **84**, no. 3, R333–R340.

McMechan, G. A., 1983, Migration by extrapolation of time-dependent boundary values: Geophysical Prospecting, **31**, 413–420.

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.

Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208–221.

Nihei, K. T., and X. Li, 2007, Frequency response modelling of seismic waves using finite difference time domain with phase sensitive detection (TDPSD): Geophysical Journal International, **169**, 1069–1078.

Nocedal, J., and S. J. Wright, 2006, Numerical optimization 2nd.

Operto, S., J. Virieux, P. Amestoy, J.-Y. L'Excellent, L. Giraud, and H. B. H. Ali, 2007, 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study: Geophysics, **72**, no. 5, SM195–SM211.

Oropeza, V., and M. Sacchi, 2011a, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76 (3)**, V25–V32.

———, 2011b, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.

Østmo, S., W. Mulder, and R.-E. Plessix, 2002, Finite-difference iterative migration by linearized waveform inversion in the frequency domain: 72th Annual International Meeting, SEG, Expanded Abstracts, 1384–1387.

Paige, C. C., and M. A. Saunders, 1982, Lsqr: An algorithm for sparse linear equations and sparse least squares: ACM Transactions on Mathematical Software (TOMS), **8**, 43–71.

Perz, M., S. Chopra, R. Sharma, P. Cary, X. Li, W. Ohlhauser, K. Pike, B. Creaser, and M. H. Nemati, 2016, Characterizing a cardium waterflood via 3-c–3d land surface seismic: The washout creek experience, *in* SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 2770–2774.

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.

———, 2015, Tensor deflation for CANDECOMP/PARAFAC - Part II: Initialization and error analysis: IEEE Transactions on Signal Processing, **63**, 5939–5950.

Philippe, T., B. Thierry, and U. Michael, 2000, Image interpolation and resampling: Handbook of medical imaging, processing and analysis, **1**, no. 1, 393–420.

Pitsianis, N. P., 1997, The Kronecker product in approximation and fast transform generation: Cornell University.

Plessix, R.-E., and W. Mulder, 2004, Frequency-domain finite-difference amplitude-

preserving migration: Geophysical Journal International, **157**, 975–987.

Pratt, R. G., C. Shin, and G. Hick, 1998, Gauss–Newton and full Newton methods in frequency–space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Richardson, A., and C. Feller, 2019, Seismic data denoising and deblending using deep learning: arXiv preprint arXiv:1907.01497.

Rickett, J. E., 2003, Illumination-based normalization for wave-equation depth migration: Geophysics, **68**, 1371–1379.

Rokhlin, V., and M. Tygert, 2008, A fast randomized algorithm for overdetermined linear least-squares regression: Proceedings of the National Academy of Sciences, **105**, 13212–13217.

Sacchi, M. D., J. Gao, A. Stanton, and J. Cheng, 2015, Tensor factorization and its application to multidimensional seismic data recovery: 85th SEG Annual Meeting, New Orleans, Louisiana, USA, Expanded Abstracts, 4827–4831.

Sacchi, M. D., and H. Kuehl, 2001, ARMA formulation of f-x prediction error filters and projection filters: Journal of Seismic Exploration, **9**, 185–198.

Sacchi, M. D., and T. J. Ulrych, 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, 1169–1177.

Sacchi, M. D., T. J. Ulrych, and C. J. Walker, 1998, Interpolation and extrapolation using a high-resolution discrete Fourier transform: IEEE Transactions on Signal Processing, **46**, 31–38.

Schacke, K., 2004, On the Kronecker product: Master's thesis, University of Waterloo.

Shewchuk, J. R., et al., 1994, An introduction to the conjugate gradient method without the agonizing pain.

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.

Sidiropoulos, N. D., R. Bro, and G. B. Giannakis, 2000, Parallel factor analysis in sensor array processing: IEEE transactions on Signal Processing, **48**, 2377–2388.

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.

Sirgue, L., J. T. Etgen, and U. Albertin, 2008, 3D Frequency Domain Waveform Inversion Using Time Domain Finite Difference Methods: Presented at the 79th Annual International conference and Exhibition, EAGE, Extended Abstracts.

Sjoeberg, T. A., L.-J. Gelius, and I. Lecomte, 2003, 2-D deconvolution of seismic image blur: 73th Annual International Meeting, SEG, Expanded Abstracts, 1055–1058.

Smilde, A. K., Y. Wang, and B. R. Kowalski, 1994, Theory of medium-rank second-order calibration with restricted-tucker models: Journal of Chemometrics, **8**, 21–36.

Soubaras, R., 1994, Signal-preserving random noise attenuation by the f-x projection: 64th SEG Annual Meeting, Los Angeles, California, USA, Expanded Abstracts, 1576–1579.

———, 1995, Prestack random and impulsive noise attenuation by f-x projection filtering: 65th SEG Annual Meeting, Houston, Texas, USA, Expanded Abstracts, 711–714.

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

Starck, J.-L., E. J. Candès, and D. L. Donoho, 2002, The curvelet transform for image denoising: IEEE Transactions on Image Processing, **11**, 670–684.

Stewart, R. R., J. E. Gaiser, R. J. Brown, and D. C. Lawton, 2002, Converted-wave seismic exploration: Methods: Geophysics, **67**, 1348–1363.

———, 2003, Converted-wave seismic exploration: Applications: Geophysics, **68**, no. 1, 40–57.

Sun, J., D. Tao, and C. Faloutsos, 2006, Beyond streams and graphs: dynamic tensor analysis: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 374–383.

Sun, J.-T., H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, 2005, Cubesvd: a novel approach to personalized web search: Proceedings of the 14th international conference on World Wide Web, 382–390.

Tang, Y., 2009, Target-oriented wave-equation least-squares migration/inversion with phase-encoded hessian: Geophysics, **74**, no. 6, WCA95–WCA107.

Tarantola, A., 1984a, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

———, 1984b, Linearized inversion of seismic reflection data: Geophysical prospecting, **32**, 998–1015.

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, 2010a, Rank-reduction-based trace interpolation: 80th SEG Annual Meeting, Denver, Colorado, USA, Expanded Abstracts, 3829–3833.

———, 2010b, 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.

Tu, N., and F. J. Herrmann, 2015, Fast imaging with surface-related multiples by sparse inversion: Geophysical Journal International, **201**, 304–317.

Tucker, L. R., 1966, Some mathematical notes on three-mode factor analysis: Psychometrika, **31**, 279–311.

Ursenbach, C., P. Cary, and M. Perz, 2013, Limits on resolution enhancement for PS data mapped to PP time: The Leading Edge, **32**, no. 1, 64–71.

Valenciano, A. A., B. Biondi, and A. Guitton, 2006, Target-oriented wave-equation inversion: Geophysics, **71**, no. 4, A35–A38.

Van Dok, R., and P. Kristiansen, 2003, Event registration and Vp/Vs correlation analysis in 4C processing: SEG Technical Program Expanded Abstracts, 785–788.

Van Loan, C. F., 2000, The ubiquitous Kronecker product: Journal of computational and applied mathematics, **123**, 85–100.

Van Loan, C. F., and N. Pitsianis, 1993, Approximation with Kronecker products, *in* Linear algebra for large scale and real-time applications: Springer, 293–314.

Vasilescu, M. A. O., and D. Terzopoulos, 2002, Multilinear image analysis for facial recognition: Object recognition supported by user interaction for service robots, IEEE, 511–514.

———, 2003, Multilinear subspace analysis of image ensembles: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., IEEE, II–93.

Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889–901.

Wang, E., and J. Nealon, 2019, Applying machine learning to 3D seismic image denoising and enhancement: Interpretation, **7 (3)**, SE131–SE139.

Wang, F., and S. Chen, 2019, Residual learning of deep convolutional neural network for seismic random noise attenuation: IEEE Geoscience and Remote Sensing Letters, **16**, 1314–1318.

Wang, J., M. Ng, and M. Perz, 2010, Seismic data interpolation by greedy local Radon transform: Geophysics, **75 (6)**, WB225–WB234.

Wang, Y., H.-Y. Tung, A. J. Smola, and A. Anandkumar, 2015a, Fast and guaranteed tensor decomposition via sketching: Advances in Neural Information Processing Systems, **28**, 991–999.

Wang, Z., M.-J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye, 2015b, Orthogonal rank-one matrix pursuit for low rank matrix completion: SIAM Journal on Scientific Computing, **37**, A488–A514.

Whitmore, N. D., 1983, Iterative depth migration by backward time propagation: 53th Annual International Meeting, SEG, Expanded Abstracts, 382–385.

Wong, M., B. L. Biondi, and S. Ronen, 2015, Imaging with primaries and free-surface multiples by joint least-squares reverse time migration: Geophysics, **80**, no. 6, S223–S235.

Wright, S., and J. Nocedal, 1999, Numerical optimization: Springer Science, **35**, 67–68.

Xu, L., and M. D. Sacchi, 2017, Preconditioned acoustic least-squares two-way wave-equation migration with exact adjoint operator: Geophysics, **83**, S1–S13.

Xu, Y., R. Hao, W. Yin, and Z. Su, 2015, Parallel matrix factorization for low-rank tensor completion: American Institute of Mathematical Sciences, **9**, 601–624.

Xue, Z., Y. Chen, S. Fomel, and J. Sun, 2015, Seismic imaging of incomplete data and

simultaneous-source data using least-squares reverse time migration with shaping regularization: Geophysics, **81**, no. 1, S11–S20.

Yilmaz, Ö., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Society of exploration geophysicists.

Youn, O. K., and H.-w. Zhou, 2001, Depth imaging with multiples: Geophysics, **66**, 246–255.

Yu, J., J. Hu, G. T. Schuster, and R. Estill, 2006, Prestack migration deconvolution: Geophysics, **71**, no. 2, S53–S62.

Yuan, J. J., G. Nathan, A. Calvert, and R. Bloor, 2008, Automated C-wave registration by simulated annealing: SEG Technical Program Expanded Abstracts, 1043–1047.

Zhang, D., and G. T. Schuster, 2013, Least-squares reverse time migration of multiples: Geophysics, **79**, S11–S21.

Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang, 2017, Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising: IEEE Transactions on Image Processing, **26**, 3142–3155.

Zhang, Y., L. Duan, and Y. Xie, 2014, A stable and practical implementation of least-squares reverse time migration: Geophysics, **80**, no. 1, V23–V31.

Zhang, Y., and J. Sun, 2009, Practical issues of reverse time migration: True amplitude gathers, noise removal and harmonic-source encoding: Beijing 2009 International Geophysical Conference and Exposition, Beijing, China, 204–204.

Zuleta, L. M., and D. C. Lawton, 2012, Vp/Vs characterization of a shale gas basin, northeast British Columbia, Canada: SEG Technical Program Expanded Abstracts, 1–5.

Zwartjes, P., and M. Sacchi, 2006, Fourier reconstruction of nonuniformly sampled, aliased seismic data: Geophysics, **72**, V21–V32.

# APPENDIX A

## Relationship between the $V_p/V_s$ ratio and warping function

To derive equation 5.5, I use a simple three-layer model with two reflectors. I assume that thickness of first two layers in PP-wave time is $\Delta t$, so the two-way PP-wave travel time for the first and second reflections are $\Delta t$ and $2\Delta t$, respectively. The P-wave and S-wave velocities in the first two layers are denoted by $vp_1$, $vs_1$, $vp_2$ and $vs_2$. The corresponding PS-wave travel time can be expressed as

$$w(\Delta t) = \frac{vp_1 \cdot \Delta t/2}{vp_1} + \frac{vp_1 \cdot \Delta t/2}{vs_1}\,, \tag{A.1}$$

and

$$w(2\Delta t) = \frac{vp_1 \cdot \Delta t/2}{vp_1} + \frac{vp_2 \cdot \Delta t/2}{vp_2} + \frac{vp_2 \cdot \Delta t/2}{vs_2} + \frac{vp_1 \cdot \Delta t/2}{vs_1}\,. \tag{A.2}$$

With this simple derivation, one can write the following expression

$$\frac{w(2\Delta t) - w(\Delta t)}{2\Delta t - \Delta t} = \frac{1}{2} + \frac{1}{2}\frac{vp_2}{vs_2}\,, \tag{A.3}$$

which for the continuous case is equivalent to equation 5.5.

# APPENDIX B

## Discrete second derivative operator

I enforce smoothness on the warping function by applying a regularization formed with second order derivatives (equation 5.20). In this section, I describe a simple technique to specify sparse discrete second derivative operators $\mathbf{L}_x$ and $\mathbf{L}_t$. I start defining second order discrete derivatives in time and space

$$\begin{aligned} \frac{\partial^2 w(t,x)}{\partial t^2} &\approx \frac{w(t+\Delta t,x) - 2w(t,x) + w(t-\Delta t,x)}{\Delta t^2}\,, \\ \frac{\partial^2 w(t,x)}{\partial x^2} &\approx \frac{w(t,x+\Delta x) - 2w(t,x) + w(t,x-\Delta x)}{\Delta x^2}\,, \end{aligned} \tag{B.1}$$

without loosing generality we can consider both $\Delta t = \Delta x = 1$ and write

$$\begin{aligned} \frac{\partial^2}{\partial t^2} &\approx \mathbf{L}_t = \mathbf{I}_n \otimes \mathbf{D}_2\,, \\ \frac{\partial^2}{\partial x^2} &\approx \mathbf{L}_x = \mathbf{D}_2 \otimes \mathbf{I}_m\,, \end{aligned} \tag{B.2}$$

where $\mathbf{I}_n$ is an identity matrix of size $n \times n$, $n$ is the number of traces, similarly $\mathbf{I}_m$ is also an identity matrix of size $m \times m$, $m$ is the number of samples per trace. $\mathbf{D}_2$ is the matrix form of second order derivatives, and $\otimes$ indicates Kronecker product

$$\mathbf{D_2} = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & -1 & 1 \end{bmatrix}. \tag{B.3}$$

Now, $\mathbf{L}_x$ and $\mathbf{L}_t$ are sparse operators that can be directly applied to a vectorized 2D warping function to express second order smoothing in space and time, respectively.

# APPENDIX C

## Direct estimation of Kronecker product

Direct estimation refers to estimating the Kronecker factors when the complete Hessian matrix is known. In this scenario, the factors can be computed through a low-rank approximation (e.g., truncated SVD) of the rearranged Hessian. The process of approximating a block matrix via the superposition of Kronecker products is illustrated with a toy example. Consider a $4 \times 4$ block matrix $\mathbf{H}$ composed of $2 \times 2$ blocks. The matrix $\mathbf{H}$ is the Kronecker product of two $2 \times 2$ matrices $\mathbf{A}$ and $\mathbf{B}$, written explicitly they read as

$$\mathbf{A} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} b_1 & b_3 \\ b_2 & b_4 \end{bmatrix}, \tag{C.1}$$

where $a_i$ and $b_i$ denote the elements of the two matrices. In terms of $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{H}$ has the form

$$\mathbf{H} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_3 \\ b_2 & b_4 \end{bmatrix} = \begin{bmatrix} a_1 \begin{pmatrix} b_1 & b_3 \\ b_2 & b_4 \end{pmatrix} & a_3 \begin{pmatrix} b_1 & b_3 \\ b_2 & b_4 \end{pmatrix} \\ a_2 \begin{pmatrix} b_1 & b_3 \\ b_2 & b_4 \end{pmatrix} & a_4 \begin{pmatrix} b_1 & b_3 \\ b_2 & b_4 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}, \tag{C.2}$$

where $\mathbf{H}_{ij}$ indicates one block of $\mathbf{H}$. To determine the connection between $\mathbf{H}_{ij}$ and the pair $\mathbf{A}$ and $\mathbf{B}$, I rearrange the elements of $\mathbf{H}$ in a two step process. First, the blocks of $\mathbf{H}$ are vectorized into column vectors. Secondly, the column vectors are concatenated horizontally

to form a new matrix $\tilde{\mathbf{H}}$. The procedure is expressed as follows:

$$\tilde{\mathbf{H}} = \begin{bmatrix} vec(\mathbf{H}_{11}) & vec(\mathbf{H}_{21}) & vec(\mathbf{H}_{12}) & vec(\mathbf{H}_{22}) \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_1 & a_3b_1 & a_4b_1 \\ a_1b_2 & a_2b_2 & a_3b_2 & a_4b_2 \\ a_1b_3 & a_2b_3 & a_3b_3 & a_4b_3 \\ a_1b_4 & a_2b_4 & a_3b_4 & a_4b_4 \end{bmatrix}. \quad \text{(C.3)}$$

The rearrangement operation is denoted by the following mapping (Pitsianis, 1997)

$$\tilde{\mathbf{H}} = \mathcal{R}(\mathbf{H}). \quad \text{(C.4)}$$

To provide the general rule for this rearrangement operation, I use four indices $i, j, k, l$ to indicate one element in the matrix $\mathbf{H}$ (equation 12). The indices $i, j$ indicate the block an element belongs to, whereas $k, l$ are the indices of an element within that block. For a single element, I use the notation $H_{ij_{kl}}$. For example, $H_{22_{21}}$ is the element at the $2^{nd}$ row, $1^{st}$ column in the block $\mathbf{H}_{22}$ and is equal to $a_4 \cdot b_2$. I use only two indices $i_1, i_2$ to indicate the elements in the matrix $\tilde{\mathbf{H}}$ and assume $\mathbf{A}$ and $\mathbf{B}$ are square matrices of size $n_x \times n_x$ and $n_z \times n_z$, respectively. The relationship between indices of $\mathbf{H}$ and $\tilde{\mathbf{H}}$ are given by

$$\begin{aligned} i_1 &= (l-1)\, n_z + k, \\ i_2 &= (j-1)\, n_x + i. \end{aligned} \quad \text{(C.5)}$$

For more general cases, such as when $\mathbf{A}$ and $\mathbf{B}$ are non-square matrices, interested readers are referred to Van Loan and Pitsianis (1993). In the preceding toy example, the rearranged Hessian $\tilde{\mathbf{H}}$ is a rank one matrix as it can be expressed as the outer-product of the vectorized versions of the matrices $\mathbf{B}$ and $\mathbf{A}$,

$$\tilde{\mathbf{H}} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} = vec(\mathbf{B})\, vec(\mathbf{A})^T. \quad \text{(C.6)}$$

The problem of estimating the optimal Kronecker product approximation to a block matrix is defined as

$$\underset{\mathbf{A},\mathbf{B}}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{A} \otimes \mathbf{B}\|_F^2, \quad \text{(C.7)}$$

and is equivalent to finding the optimal rank-one matrix approximation (Pitsianis, 1997) via

$$\underset{\mathbf{a},\mathbf{b}}{\operatorname{argmin}} \|\tilde{\mathbf{H}} - \mathbf{b}\mathbf{a}^T\|_F^2. \quad \text{(C.8)}$$

The Kronecker factors are obtained by reshaping the vectors obtained by solving equation C.8. A rank one matrix is not sufficient to capture the complexity of the Hessian matrix; therefore, the problem is redefined using a superposition of Kronecker products in equations 6.12 and 6.13. The Kronecker factors can be retrieved by reshaping the first $k$ singular vectors of the rearranged Hessian into matrices. This process is summarized in algorithm 1.

**Algorithm 4** SKP

**function** $\{\mathbf{A}_i, \mathbf{B}_i\} = \textsc{Sum_of_Kronecker_Products}(\mathbf{H}, k, nz, nx)$
    Rearrange the block matrix $\mathbf{H}$: $\tilde{\mathbf{H}} = \mathcal{R}(\mathbf{H}, nz, nx)$
    Evaluate the truncated SVD of $\tilde{\mathbf{H}}$: $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = \texttt{svd}(\tilde{\mathbf{H}})$
    **for** $i = 1 : k$ **do**
        $\mathbf{A}_i = \texttt{reshape}(\mathbf{V}[:, i] \cdot \sqrt{\mathbf{S}[i, i]}, nx, nx)$
        $\mathbf{B}_i = \texttt{reshape}(\mathbf{U}[:, i] \cdot \sqrt{\mathbf{S}[i, i]}, nz, nz)$
    **end for**
    Output $\mathbf{A}_i$, $\mathbf{B}_i$
**end function**

# APPENDIX D

## Kronecker factors estimation using matrix completion

This appendix provides the algorithm for estimating the Kronecker factors from randomly selected samples of the Hessian matrix $\mathbf{H}$. I indicate random samples of $\mathbf{H}$ with $\mathbf{H}_\Omega$, where $\Omega$ indicates the sampled set. This algorithm is called economical orthogonal rank-one matrix completion (EOR1MP) (Wang et al., 2015b). Besides the random samples, the input to the algorithm includes the number of factors, which is indicated by $k$, and the size of the Kronecker factors $(n_z, n_x)$. In this algorithm, I organize the samples as a sparse matrix. The matrix $\mathbf{X}^i$ contains the approximations to observed samples after the $i^{th}$ iteration. The vector $\boldsymbol{\theta}$ has a length of $k$. At step 1, the top left and right singular vectors $(\mathbf{u}_i, \mathbf{v}_i)$ are estimated with Power method from the current residual matrix $\mathbf{R}^i$ (Jaggi and Sulovskỳ, 2010). The singular vectors are saved to recover the Kronecker factors as square matrices. The optimal weights $\alpha_1, \alpha_2$ in step 2 are computed from the least-squares solution of a linear inverse problem in which the weights are the unknowns and the vectorized non-zero elements of the sparse matrix $\mathbf{H}_\Omega$ form the data. The non-zero elements of $\mathbf{X}^{i-1}$ and $\mathbf{M}^i$ are vectorized and concatenated to form a two-column matrix.

**Algorithm 5** EOR1MP

**function** $\{\mathbf{A}_i, \mathbf{B}_i\} = \texttt{EOR1MP}(\mathbf{H}_\Omega, k, n_z, n_x)$
  **Initialize**: Set $\mathbf{X}^0 = \mathbf{0}$, $\boldsymbol{\theta}^0 = \mathbf{0}$
  **for** $i = 1 : k$ **do**
    **Step 1**: Find a pair of top left and right singular vectors $(\mathbf{u}^i, \mathbf{v}^i)$ of the observed residual matrix $\mathbf{R}^i = \mathbf{H}_\Omega - \mathbf{X}^{i-1}$ by the Power method. Set $\mathbf{M}^i = \mathbf{u}^i \cdot \mathbf{v}^{iT}$
    **Step 2**: Compute the optimal weights $\alpha_1$, $\alpha_2$ for $\mathbf{X}^{i-1}$ and $\mathbf{M}^i$ by solving

$$\underset{\alpha_1, \alpha_2}{\operatorname{argmin}} \|\alpha_1 \mathbf{X}^{i-1} + \alpha_2 (\mathbf{M}^i)_\Omega - \mathbf{H}_\Omega\|_F^2$$

    **Step 3**: Set $\mathbf{X}^i = \alpha_1 \mathbf{X}^{i-1} + \alpha_2 (\mathbf{M}^i)_\Omega$; $\theta_i^i = \alpha_2$ and $\theta_j^i = \theta_j^{i-1} \cdot \alpha_1$ for $j = 1, 2 \cdots i-1$.
  **end for**
  **for** $i = 1 : k$ **do**
    $\mathbf{A}_i = \texttt{reshape}(\mathbf{u}^i, n_z, n_z) \cdot \theta_i^k$
    $\mathbf{B}_i = \texttt{reshape}(\mathbf{v}^i, n_x, n_x)$
  **end for**
  **Output** $\mathbf{A}_i$, $\mathbf{B}_i$
**end function**

# APPENDIX E

## Constrained kronecker factors estimation

I illustrate the process of estimating Kronecker factors with constraints (equation 3.12) using a toy example. Suppose I have a $9 \times 9$ block band matrix $\mathbf{C}$, which is the Kronecker product of two square band matrices size of $3 \times 3$ and bandwidth 2. Mathematically, this is expressed as:

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B}, \ \mathbf{A}, \mathbf{B} \in \mathbb{M}_3^2. \tag{E.1}$$

I further define the two band matrices as

$$\mathbf{A} = \begin{pmatrix} a_1 & a_3 & 0 \\ a_2 & a_4 & a_6 \\ 0 & a_5 & a_7 \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix}, \tag{E.2}$$

so the matrix $\mathbf{C}$ is expressed as

$$\mathbf{C} = \begin{pmatrix} a_1 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} & a_3 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} & \mathbf{0} \\ a_2 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} & a_4 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} & a_6 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} \\ \mathbf{0} & a_5 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} & a_7 \begin{pmatrix} b_1 & b_3 & 0 \\ b_2 & b_4 & b_6 \\ 0 & b_5 & b_7 \end{pmatrix} \end{pmatrix}. \tag{E.3}$$

Based on the properties of Kronecker products, the rearranged matrix $\tilde{\mathbf{C}}$ can be expressed as

$$\tilde{\mathbf{C}} = \mathcal{R}(\mathbf{C}) = vec(\mathbf{B})\, vec(\mathbf{A})^T, \tag{E.4}$$

which, written element-wise, is equivalent to

$$\begin{pmatrix} a_1b_1 & a_2b_1 & 0 & a_3b_1 & a_4b_1 & a_5b_1 & 0 & a_6b_1 & a_7b_1 \\ a_1b_2 & a_2b_2 & 0 & a_3b_2 & a_4b_2 & a_5b_2 & 0 & a_6b_2 & a_7b_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1b_3 & a_2b_3 & 0 & a_3b_3 & a_4b_3 & a_5b_3 & 0 & a_6b_3 & a_7b_3 \\ a_1b_4 & a_2b_4 & 0 & a_3b_4 & a_4b_4 & a_5b_4 & 0 & a_6b_4 & a_7b_4 \\ a_1b_5 & a_2b_5 & 0 & a_3b_5 & a_4b_5 & a_5b_5 & 0 & a_6b_5 & a_7b_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1b_6 & a_2b_6 & 0 & a_3b_6 & a_4b_6 & a_5b_6 & 0 & a_6b_6 & a_7b_6 \\ a_1b_7 & a_2b_7 & 0 & a_3b_7 & a_4b_7 & a_5b_7 & 0 & a_6b_7 & a_7b_7 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \\ b_3 \\ b_4 \\ b_5 \\ 0 \\ b_6 \\ b_7 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & 0 & a_3 & a_4 & a_5 & 0 & a_6 & a_7 \end{pmatrix}. \tag{E.5}$$

I can also represent the vectorization of any matrix in the set of $\mathbb{M}_3^2$ as the multiplication of a condensed vector composed of the matrix diagonals, by a projection matrix

$$\mathbf{a} = vec(\mathbf{A}) = \begin{pmatrix} a_1 \\ a_2 \\ 0 \\ a_3 \\ a_4 \\ a_5 \\ 0 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \mathbf{Q_a}\hat{\mathbf{a}}. \tag{E.6}$$

The matrix $\mathbf{Q_a}$ inserts zeros into the condensed vector at the appropriate locations, $\hat{\mathbf{a}}$ is the new vector consisting of the diagonals of $\mathbf{A}$. Similarly, the vectorization of $\mathbf{B}$ can be written as

$$\mathbf{b} = vec(\mathbf{B}) = \mathbf{Q}_b\hat{\mathbf{b}}, \tag{E.7}$$

where $\hat{\mathbf{b}} = \begin{pmatrix} b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & b_7 \end{pmatrix}^T$. $\mathbf{Q}_b$ is equal to $\mathbf{Q}_a$ in this special example as the matrices $\mathbf{A}$ and $\mathbf{B}$ have the same size and bandwidth. The algorithm can be extended to more general cases when the size and bandwidth of $\mathbf{A}$ and $\mathbf{B}$ are different. Substituting equations E.6 and E.7 into equation E.4, I obtain

$$\mathcal{R}(\mathbf{C}) = \mathbf{Q}_b\hat{\mathbf{b}}\hat{\mathbf{a}}^T\mathbf{Q}_a^T. \tag{E.8}$$

A left and right multiplication of equation E.8 by $\mathbf{Q}_b^T$ and $\mathbf{Q}_a$ yields

$$\mathbf{Q}_b^T \mathcal{R}(\mathbf{C}) \mathbf{Q}_a = \mathbf{Q}_b^T \mathbf{Q}_b \hat{\mathbf{b}} \hat{\mathbf{a}}^T \mathbf{Q}_a^T \mathbf{Q}_a. \tag{E.9}$$

Since $\mathbf{Q}_b^T \mathbf{Q}_b = \mathbf{I}$ and $\mathbf{Q}_a^T \mathbf{Q}_a = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, the previous equation simplifies to

$$\mathbf{Q}_b^T \mathcal{R}(\mathbf{C}) \mathbf{Q}_a = \hat{\mathbf{b}} \hat{\mathbf{a}}^T. \tag{E.10}$$

An interesting result arises from the product $\mathbf{Q}_b^T \mathcal{R}(\mathbf{C}) \mathbf{Q}_a$, expressed explicitly it is

$$\mathbf{Q}_b^T \mathcal{R}(\mathbf{C}) \mathbf{Q}_a = \begin{pmatrix} a_1b_1 & a_2b_1 & a_3b_1 & a_4b_1 & a_5b_1 & a_6b_1 & a_7b_1 \\ a_1b_2 & a_2b_2 & a_3b_2 & a_4b_2 & a_5b_2 & a_6b_2 & a_7b_2 \\ a_1b_3 & a_2b_3 & a_3b_3 & a_4b_3 & a_5b_3 & a_6b_3 & a_7b_3 \\ a_1b_4 & a_2b_4 & a_3b_4 & a_4b_4 & a_5b_4 & a_6b_4 & a_7b_4 \\ a_1b_5 & a_2b_5 & a_3b_5 & a_4b_5 & a_5b_5 & a_6b_5 & a_7b_5 \\ a_1b_6 & a_2b_6 & a_3b_6 & a_4b_6 & a_5b_6 & a_6b_6 & a_7b_6 \\ a_1b_7 & a_2b_7 & a_3b_7 & a_4b_7 & a_5b_7 & a_6b_7 & a_7b_7 \end{pmatrix} = \hat{\mathbf{b}} \hat{\mathbf{a}}^T. \tag{E.11}$$

The left and right multiplication of $\mathcal{R}(\mathbf{C})$ with $\mathbf{Q}_b^T$ and $\mathbf{Q}_a$ selects the non-zero rows and columns from the rearranged matrix. The constrained optimization problem expressed in equation 3.13 can be converted to an unconstrained problem of the form

$$\underset{\hat{\mathbf{a}},\, \hat{\mathbf{b}}}{\operatorname{argmin}} \, \| \mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a - \hat{\mathbf{b}} \hat{\mathbf{a}}^T \|_F^2. \tag{E.12}$$

The equivalent expression for a matrix approximated by a superposition of Kronecker products is given by

$$\underset{\hat{\mathbf{A}},\, \hat{\mathbf{B}}}{\operatorname{argmin}} \, \| \mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a - \hat{\mathbf{B}} \hat{\mathbf{A}}^T \|_F^2, \tag{E.13}$$

where $\hat{\mathbf{A}} = (\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \cdots, \hat{\mathbf{a}}_k)$ and $\hat{\mathbf{B}} = (\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \cdots, \hat{\mathbf{b}}_k)$. The band Kronecker factor matrix $\mathbf{A}_i$ can be recovered by reshaping $\mathbf{Q}_a \hat{\mathbf{a}}_i$ into a square matrix. For additional constraints e.g., symmetry, orthogonality, Toeplitz structure, readers are referred to Pitsianis (1997). I summarize the computation of the band Kronecker factor matrix in Algorithm 6. This algorithm is only intended to provide insight. In real applications, I never compute the Kronecker factors through a direct matrix decomposition. Uniform random sampling is applied to the elements of $\mathbf{Q}_b^T \mathcal{R}(\mathbf{H}) \mathbf{Q}_a$ to compute $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ via the matrix completion technique described in Appendix D.

**Algorithm 6** SBKP

**function** $\{\mathbf{A}_i, \mathbf{B}_i\} = \texttt{SUM\_OF\_BAND\_KRONECKER}(\mathbf{H}, k, n_z, n_x, n_{r_z}, n_{r_x})$
  Rearrange block matrix $\tilde{\mathbf{H}} = \mathcal{R}(\mathbf{H}, nz, nx)$
  Build matrix $\mathbf{Q}_a = \texttt{build\_matrix}(n_x, n_{r_x})$
  Build matrix $\mathbf{Q}_b = \texttt{build\_matrix}(n_z, n_{r_z})$
  Compute $\hat{\mathbf{H}} = \mathbf{Q}_b^T \tilde{\mathbf{H}} \mathbf{Q}_a$
  Singular value decomposition $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = svd(\hat{\mathbf{H}})$
  **for** $i = 1 : k$ **do**
    $\hat{\mathbf{a}}_i = \mathbf{V}[:, i] \cdot \sqrt{\mathbf{S}[i, i]}$
    $\hat{\mathbf{b}}_i = \mathbf{U}[:, i] \cdot \sqrt{\mathbf{S}[i, i]}$
    $\mathbf{a}_i = \mathbf{Q}_a \cdot \hat{\mathbf{a}}_i$
    $\mathbf{b}_i = \mathbf{Q}_b \cdot \hat{\mathbf{b}}_i$
    $\mathbf{A}_i = reshape(\mathbf{a}_i, nx, nx)$
    $\mathbf{B}_i = reshape(\mathbf{b}_i, nz, nz)$
  **end for**
  Output $\mathbf{A}_i, \mathbf{B}_i$
**end function**

# APPENDIX F

## Reduced Hessian: Implementation details

In Appendix E, I established that the reduced Hessian $\hat{\mathbf{H}} = \mathbf{Q}_b^T \tilde{\mathbf{H}} \mathbf{Q}_a$ is obtained by selecting rows and columns of the rearranged Hessian $\tilde{\mathbf{H}}$. Because of the memory cost, I never build the full Hessian $\mathbf{H}$ explicitly. Likewise, the rearranged Hessian $\tilde{\mathbf{H}}$, $\mathbf{Q}_a$, $\mathbf{Q}_b$ and reduced Hessian $\hat{\mathbf{H}}$ are also not computed explicitly. As discussed earlier, the Kronecker factors are estimated from the random samples of the reduced Hessian $\hat{\mathbf{H}}$ (preferential samples of $\mathbf{H}$) via low rank matrix completion. As a result, only random samples of the reduced Hessian need to be computed. The restriction (and sampling) of $\mathbf{H}$ amounts to finding a mapping relationship between the indices of $\mathbf{H}$ and $\hat{\mathbf{H}}$. The desired samples can then be computed using equation 6.15. The remainder of this section explores the mapping relationship between $\mathbf{H}$ and $\hat{\mathbf{H}}$ in detail.

Figure D-1a displays the connection between the three forms of the Hessian matrix. For one identical element, its index is represented as $[ir, ic]$, $[\tilde{ir}, \tilde{ic}]$,$[\hat{ir}, \hat{ic}]$ for the Hessian, rearranged Hessian and the reduced Hessian, respectively.

I assume the index of one image point $\mathbf{x}$ is $(iz, ix)$ and the index of another image point $\mathbf{x} + \mathbf{r}$, in its vicinity, is $(iz + ir_z, ix + ir_x)$ with $|ir_z| \leqslant n_{r_z}$ and $|ir_x| \leqslant n_{r_x}$. Half of user-defined window sizes are denoted by $n_{r_z}$ and $n_{r_x}$. The index of the Hessian element $H[ir, ic]$ corresponding to these two image points is

$$\begin{aligned} ir &= (ix - 1) \cdot n_z + iz, \\ ic &= (ix + ir_x - 1) \cdot n_z + iz + ir_z, \end{aligned} \tag{F.1}$$

where $ir$ and $ic$ are the row and column indices of $\mathbf{H}$, respectively. This element moves to

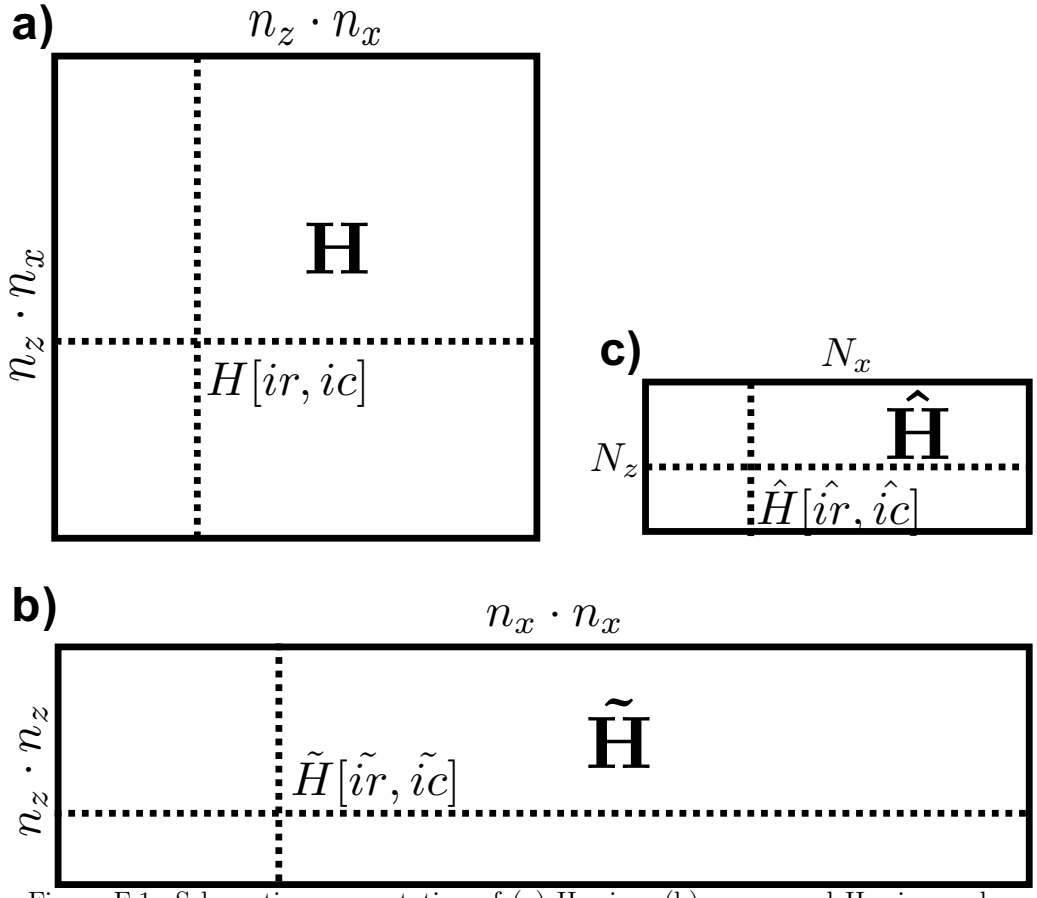


Figure F.1: Schematic representation of (a) Hessian, (b) rearranged Hessian and the (c) reduced Hessian matrices.

the location $(\tilde{ir}, \tilde{ic})$ in the rearranged Hessian $\tilde{\mathbf{H}}$ with

$$\begin{aligned} \tilde{ir} &= (iz + ir_z - 1) \cdot n_z + iz, \\ \tilde{ic} &= (ix + ir_x - 1) \cdot n_x + ix. \end{aligned} \tag{F.2}$$

Mapping the elements of Hessian $\mathbf{H}$ to the elements of reduced Hessian $\hat{\mathbf{H}}$ is realized with the help of two auxiliary mapping matrices $\mathbf{T}_x$ and $\mathbf{T}_y$. One is for the row index and the other is for the column index. I use a toy example to illustrate how to build the mapping

matrices. Consider a square $4 \times 4$ band matrix with a bandwidth of $b = 1$,

$$\mathbf{A} = \begin{bmatrix} a_1 & a_3 & 0 & 0 \\ a_2 & a_4 & a_6 & 0 \\ 0 & a_5 & a_7 & a_9 \\ 0 & 0 & a_8 & a_{10} \end{bmatrix}. \tag{F.3}$$

The vectorization of $\mathbf{A}$ can be expressed as

$$\mathbf{a} = vec(\mathbf{A}) = \begin{pmatrix} a_1 \\ a_2 \\ 0 \\ 0 \\ a_3 \\ a_4 \\ a_5 \\ 0 \\ 0 \\ a_6 \\ a_7 \\ a_8 \\ 0 \\ 0 \\ a_9 \\ a_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{pmatrix} = \mathbf{Q}_{\mathbf{a}} \hat{\mathbf{a}}, \tag{F.4}$$

I create the following mapping matrix

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 3 \\ 2 & 4 & 6 \\ 5 & 7 & 9 \\ 8 & 10 & 0 \end{bmatrix}. \tag{F.5}$$

The mapping matrix records the locations that the banded elements map to. The size of the mapping matrix is $n \times (2b+1)$. The $(b+1)^{th}$ column saves the locations of the diagonal elements, which are stored in the central column of matrix $\mathbf{T}$. The location of the $r^{th}$ off-diagonal elements are stored in the $(b+1+r)^{th}$ columns of $\mathbf{T}$, $r$ is the offset of the off-diagonal to the main diagonal and satisfies $|r| \leqslant b$. For example, one element $\mathbf{A}[i, i+r]$ is mapped to $\hat{\mathbf{a}}[\hat{i}]$ and $\hat{i} = \mathbf{T}[i, b+1+r]$.

In this application, I need to create two auxiliary matrices $\mathbf{T}_z$ and $\mathbf{T}_x$ for the index mapping.

The matrix $\mathbf{T}_z$ is related to the vertical model size $n_z$ and the vertical half window size $n_{r_z}$. The matrix $\mathbf{T}_x$ relates to the horizontal model size $n_x$ and the horizontal half window size $n_{r_x}$. Their sizes are $n_z \times (2n_{r_z}+1)$ and $n_x \times (2n_{r_x}+1)$. I get the index of the element in the reduced Hessian $\hat{\mathbf{H}}$ via the following equation

$$\begin{aligned} \hat{ir} &= \mathbf{T}_z[iz, n_{r_z}+1+ir_z] \\ \hat{ic} &= \mathbf{T}_x[ix, n_{r_x}+1+ir_x] \end{aligned}. \tag{F.6}$$

I can also map elements from the reduced Hessian back to the Hessian. To facilitate the mapping, I define another matrix $\mathbf{F}$ based on the content of $\mathbf{T}$. The matrix $\mathbf{F}$ is given as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 2 & -1 \\ 1 & 1 \\ 2 & 0 \\ 3 & -1 \\ 2 & 1 \\ 3 & 0 \\ 4 & -1 \\ 3 & 1 \\ 4 & 0 \end{bmatrix}. \tag{F.7}$$

The size of $\mathbf{F}$ is $p \times 2$ where $p$ is the maximum element in matrix $\mathbf{T}$. The first column saves the row index of the element when it maps back to a banded matrix. The second column saves the offset of an element to the main diagonal. The following describes how to map the vector $\hat{\mathbf{a}}$ back to the banded matrix $\mathbf{A}$. Take the element $\hat{a}[\hat{i}]$ as an example, the row index is given as $i = \mathbf{F}[\hat{i}, 1]$ and the offset is $r = \mathbf{F}[\hat{i}, 2]$. So the column index $j$ of this element is computed as $j = i + r$. In terms of the Hessian matrix, I need to build two mapping matrices $\mathbf{F}_z$ and $\mathbf{F}_x$ for the vertical and horizontal direction, respectively. For one element of the reduced Hessian $\hat{\mathbf{H}}[\hat{ir}, \hat{ic}]$, I can get the index of the image point and the offset with respect to it, they are given as

$$\begin{aligned} iz &= \mathbf{F}_z[\hat{ir}, 1] \ , \ ir_z = \mathbf{F}_z[\hat{ir}, 2] \\ ix &= \mathbf{F}_x[\hat{ic}, 1] \ , \ ir_x = \mathbf{F}_x[\hat{ic}, 2] \end{aligned}. \tag{F.8}$$

So the index of the corresponding Hessian element can be computed according to equation F.1.