Hypercomplex processing of vector-valued seismic signals

by

Breno Figueiredo Bahia

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

Vector field data are becoming increasingly more common in several fields of signal processing, such as electrical engineering and geophysics. Processing vector-valued signals, however, has imposed a series of challenges on its practitioners. Many approaches can be taken to carry out vector field data processing, one of which is adopting hypercomplex numbers for its representation. In particular, quaternion signal processing is a growing research area that has been gaining attention among researchers due to its natural ability to represent multicomponent data. In that light, this thesis shows that quaternions offer a consistent way to represent and fully exploit the information in vector field data. It explores how to extend techniques currently used in scalar seismic processing to the case of multicomponent data in a natural way and introduces the augmented second-order statistics of quaternion signals to harness the second-order information within the vector-valued seismic data. Two techniques are developed for processing multicomponent seismic data: the Quaternion Singular Spectrum Analysis (QSSA) and the quaternion frequency-space deconvolution (QFXDECON). These methods have been applied to synthetic and field multicomponent denoising and reconstruction, showing clear gains in their formulation. Finally, denoising methods such as the above can extend their applicability through the Regularization by Denoising (RED) framework. The RED cost function is used to formulate the simultaneous source separation problem and provides one way to leverage high-quality scalar and vector-valued denoising methods in deblending. The software developed in this thesis is mostly based on the Julia language, and it can be found in https://github.com/bbahia-phd/HCDSP.

Preface

A version of chapter 3 of this thesis has been published as a journal article: Bahia, B., and Sacchi, M. D. (2019). Quaternionic rank-reduction methods for vector-field seismic data processing. Digital Signal Processing, 87, 178-189.

A version of chapter 4 of this thesis has been published as a journal article: Bahia, B., and Sacchi, M. D. (2020). Widely linear denoising of multicomponent seismic data. Geophysical Prospecting, 68(2), 431-445.

A version of chapter 5 of this thesis has been published as a journal article: Bahia, B., Lin, R., and Sacchi, M. (2021). Regularization by denoising for simultaneous source separation. Geophysics, 86(6), P69-P83.

For the publications above, I was the author responsible for designing and developing the algorithms, data examples, and their written portions. Dr. Sacchi was the supervisory author, being involved in concept formulation, project guidance, and manuscript editing.

To my mother, who gave me everything.

Acknowledgements

The opportunity I had to work with Professor Mauricio Sacchi was life-changing, and my admiration for him only grew during my time at the Signal Analysis and Imaging Group (SAIG). I am forever grateful for the unique chance to learn from Professor Sacchi, who constantly provided guidance and support to learn and develop my skills at my own pace. With incredible patience, he also pushed me to go beyond my expectations with myself. Even the moments of doubt and frustration were possible to deconvolve with him. He put together a group that is hard to keep up with; here, I made friends who, much like our supervisor, are very inspiring to me. Their company, friendship, and collaboration were crucial during these years, especially if winter would get cold by any chance. Their input was important for my professional and personal development, and I am thankful for all our moments. They kept me going forward while away from family and other close friends, whom I miss greatly. My mother and sister are part of my daily motivation to keep learning and growing to the best possible. They constantly provide me unconditional love, support, and the privilege of pursuing my dreams and aspirations. I also cannot express more profound gratitude for having Priscila in my life. Her incentive and partnership made it much easier to believe in me and move forward while filling my days with the most beautiful and joyful laughter. I am also thankful for all my mentors throughout these years, who also gave me great incentives for my work. In particular, my period as an intern with PGS was exceptional for my development and growth. I also thank the committee members for taking their time to review, evaluate, and further improve this thesis. Good people surround me, and I feel fortunate for the life I have been enjoying.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Seismic Exploration

The conventional seismic exploration method uses compressional (P) waves to obtain meaningful images of the Earth's subsurface. Reflection seismology uses scalar seismic sources, such as dynamite shots or air guns, to generate signals that propagating into the subsurface, which are reflected and recorded back at the surface by many receivers placed in a given geometry. Reflections happen at geological interfaces, where there is a contrast in rock properties, together with energy transmission to deeper layers, as shown in Figure 1.1(a) for an acoustic medium. Each source-receiver pair defines additional sorting coordinates, such as midpoint and offset, routinely used for processing the acquired data (Yilmaz, 2001).

The conventional processing of P-wave seismic data is well-known and usually includes static corrections, coherent and random noise attenuation, deconvolution, stacking, velocity analysis, and migration. Static corrections aim at minimizing the effect of the slow weathering layer on the near surface, and a noise attenuation stage follows it to obtain useful signals. Only P-wave reflections are considered signals in P-wave seismology, and other waves must be removed. This usually includes direct arrivals, surface waves, and multiple reflections. The seismic pulse is then decompressed by deconvolution and follows with common-midpoint (CMP) sorting, velocity analysis, and subsequent averaging of the traces with same CMP (or stacking). Even though stacking is a powerful technique for random noise attenuation, post-stack processing is not unusual before migration places the recorded reflections at their correct depth position in the subsurface, thus generating a reliable reflectivity map for interpretation. In addition, it is possible to perform seismic migration before stacking to deal with lateral velocity variations, in a step coined prestack migration (Yilmaz, 2001).

1.1.1 Multicomponent Seismology

Multicomponent seismology, on the other hand, is defined as the concurrent and combined usage of compressional and shear-wave (S) energy to provide direct insights about subsurface elastic properties (*e.g.*, elastic moduli, density, and porosity) that are more difficult to obtain in the conventional (P-wave only) analysis. Each of these seismic modes is described by its particle-displacement vectors and provides complementary sensing of the subsurface that can be used in applications such as seismic imaging through gas clouds and inversion of reservoir properties (Tatham and McCormack, 1991; Garotta, 1999; Gaiser, 2016). Specifically, while P-waves distort the media along its propagating direction, S-waves have particle motion orthogonal to their ray (Figure 1.1(b)). Four waves are associated with a single incident P-wave through mode conversion for elastic isotropic media: the reflected and transmitted P-waves, along with their converted SV-wave companions. Fully recording the wavefield generated by conventional P-wave sources requires vectorial receivers to sense its additional components. The data is vector-valued and usually corresponds to the vertical and two

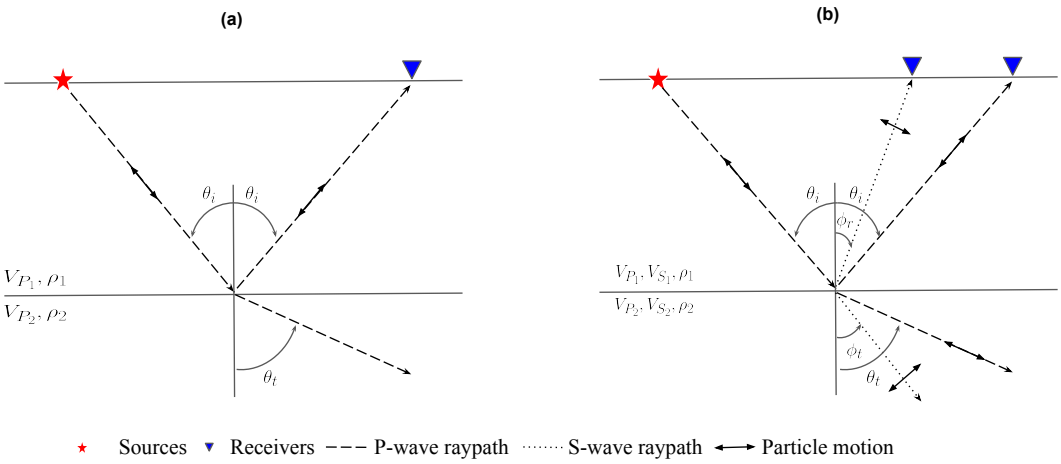

Figure 1.1: Illustrative representation of multicomponent seismic data acquisition.

A complete multicomponent characterization of the subsurface involves both sources and sensors, which must be vectorial. In other words, they respectively generate and sense variations along three-orthogonal directions. For onshore acquisition (Figures 1.2(a) and (b)), multicomponent data can have up to nine components (9C) if both vectorial sources and receivers are used; these data provide a complete sampling of the seismic modes (P, SV, and SH) when compared to three- or single-component data. In the marine acquisition

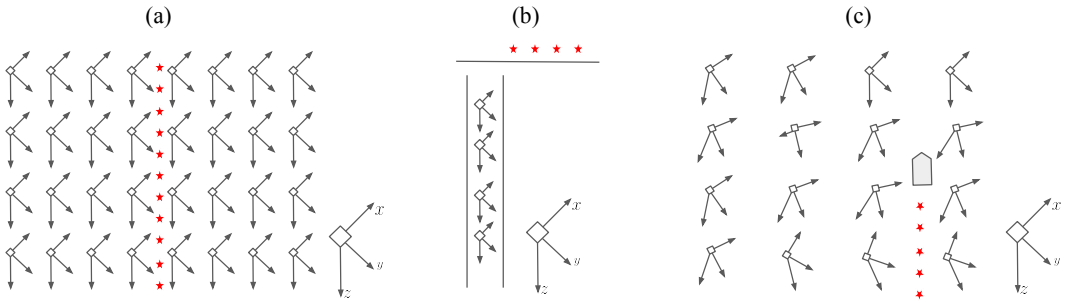

Figure 1.2: Illustrative representation of 3C multicomponent (a) onshore (b) vertical seismic profiling (VSP) and (c) marine seismic data acquisition where a vessel tows arrays of airgun sources. Sources are represented by red stars. After Gaiser (1999).

In field coordinates, the multicomponent recorded data is a mixture of surface, P-, and S-wave signals projected on each of the receiver axes. At this point, it is usual practice to refer to multicomponent data as the vector components of the displacement field here represented by u_x, u_y, and u_z. These components can also be called in-line, cross-line, and vertical components of motion, where the in-line component is defined according to some reference, such as the receiver line. Assuming a 3D acquisition as illustrated in Figure 1.2(a), sources placed at positions (s_x, s_y) and multicomponent receivers placed at (r_x, r_y) define a data volume that depends on time (t) and four spatial dimensions for each component, such as $u_x(t, s_x, s_y, r_x, r_y), u_y(t, s_x, s_y, r_x, r_y)$, and $u_z(t, s_x, s_y, r_x, r_y)$, also defining the source-receiver sorting domain. For these data, one of the most basic steps in its processing is aligning its components to preferred reference frames through rotations.

Usually, the components are rotated to a shot-centred system where the in-line and cross-line become the radial and transverse components, respectively, and can be represented by $u_r(t, s_x, s_y, r_x, r_y)$, and $u_t(t, s_x, s_y, r_x, r_y)$. This step aligns the radial component along the particle displacement of S-waves. Under the assumption of a vertical angle of incidence, it should isolate different wave modes along different components. While isotropic media u_t should have no recorded energy, anisotropic media induce phenomena like shear-wave splitting, which can be observed in the transverse component. Because of the asymmetric raypath shown in Figure 1.1(b), CMP processing is not suitable for processing converted SV-waves, which has its processing based on the common-conversion point (CCP) sorting. Processing then follows, with each component being treated in a componentwise fashion.

Componentwise processing is based on scalar techniques routinely used for P-wave data processing. Arguably, adding the data's horizontal components should not significantly change the established seismic data processing workflow. However, the componentwise processing of vector-valued data can be restrictive, especially when underlying assumptions for scalar processing do not hold. For instance, the vertical incidence is an approximation mostly valid at near offsets only, and rotating the displacement components generally does not fully isolate the wavefields as expected. Moreover, componentwise processing disregards any vector relationship between data components. Whereas the vectorial character of the data is liable for the complexities of working with multicomponent datasets, it is also the critical characteristic providing an opportunity for improvement over conventional scalar processing. Specifically, this thesis discusses random Gaussian noise attenuation, trace reconstruction, and simultaneous sources separation for vector-field seismic data. A brief scalar-based review of the topics is given in the following sections for completeness.

Random Noise Attenuation

Seismic random noise is usually described as the incoherent portion of the observed wavefield, representing a long-standing issue in seismic processing. There are many sources of random noise, such as poor receiver coupling, wind, and electrical cables nearby. The main task of denoising methods is to remove as much random noise as possible while preserving the useful, predictable from trace to trace, portion of the signal. There are several ways to attenuate noise from seismic data, as briefly introduced below.

Traditional seismic data denoising methods include the band-pass and frequency-wavenumber $(f-k)$ filtering, which are based on the assumption that signal and noise are separable in the Fourier representation of the data (Yilmaz, 2001). The frequency-space $(f-x)$ deconvolution (Canales, 1984; Gülünay, 1986) and projection (Soubaras, 1994) filters, on the other hand, are based on the predictability of linear events in the frequency domain. They are based on autoregressive models that explain the predictable portion of the data in a least-squares sense. These filters can also be formulated as $t-x$ prediction filters (Abma and Claerbout, 1995). Rank-reduction techniques are also very effective in random noise suppression, where matrix decomposition techniques, such as the Karhunen-Loéve transform (KLT) (Jones and Levy, 1987) and Singular Value Decomposition (SVD) (Freire and Ulrych, 1988), suppress the incoherent portions of the signals. Rank-reduction methods in the $f-x$ domain include the Cadzow (Trickett, 2008) and the Singular Spectrum Analysis (Sacchi, 2009) filters, which are also based on the plane-wave model to justify the low-rankness of Hankel matrices built from monochromatic frequency slices of the data. Finally, methods borrowed from the field of image processing include nonlocal means (Bonar and Sacchi, 2012), vector median filters (Liu, 2013), and various modern methods based on machine

learning (Saad and Chen, 2020).

Missing Data Recovery

Another pervasive issue, like random noise, is missing observations. Seismic acquisition commonly has irregular patterns because of geographical and logistical constraints, and reconstruction algorithms attempt to estimate missing traces following prior models similar to trace-to-trace predictability. Random noise attenuation and missing trace recovery are often performed simultaneously since their formulation shares the same prior model. Imposing sparsity or low-rankness constraints in transform domains are common approaches to seismic data reconstruction, and some examples are discussed below.

The methods for random noise suppression mentioned above are also useful for data recovery. Their underlying model, usually that of plane waves, allows for data synthesis algorithms that progressively reconstruct data. Prediction filters, in both $t-x$ (Crawley et al., 1999) and $f-x$ (Spitz, 1991; Porsani, 1999) domains, can handle data interpolation beyond aliasing. Similarly, the different formulations of the Singular Spectrum Analysis (SSA) filter can be used for multidimensional reconstruction (Oropeza and Sacchi, 2011). The SSA has evolved to be a fast and memory-efficient filter (Cheng and Sacchi, 2017) that can handle non-Gaussian noise (Chen and Sacchi, 2014; Bahia and Sacchi, 2019a) and irregular geometries (Carozzi and Sacchi, 2021). Rank-reduction methods for multidimensional seismic data reconstruction can be performed via tensor completion (Kreimer and Sacchi, 2012) also for non-Gaussian noise distributions (Carozzi and Sacchi, 2019). The multidimensional tensor is unfolded in multi ways, where parallel matrix factorization is used to obtain its low-rank approximation. Also related, sparsity-inducing methods also have remarkable performance in seismic data reconstruction. These methods have close ties with compressive sensing (Donoho, 2006), as shown by Hennenfent and Herrmann (2008) and Herrmann (2010). Similarly, the Projection Onto Convex Sets (POCS) reconstruction (Abma and Kabir, 2006) exploits sparse models in its formulation through the frequency-wavenumber thresholding. The underlying idea behind sparsity constraints is that a limited number of basis functions can represent observed signals. There are many options for sparse representation of the data, including Fourier (Sacchi and Ulrych, 1996), Radon (Trad et al., 2003), and Curvelet (Herrmann et al., 2008) domains.

Simultaneous Source Separation

Seismic data acquisition has seen fundamental changes to maximize its cost-effectiveness and efficiency. To date, significant improvements have been published in the light of blended acquisition strategies in which several sources are fired at short random time intervals (Beasley

et al., 1998; Berkhout, 2008). In addition to its economic benefits, simultaneous-source (blended) surveys allow for denser source sampling, which leads to higher quality subsurface images. The most common option to process blended data is to first separate (or deblend) its sources as if they were acquired via conventional acquisition. The most straightforward deblending process entails extracting the contributions from individual sources in a blended record. It is usually named pseudo-deblending (Berkhout, 2008), or combing (Abma and Foster, 2020) since there is no attempt to reduce interference and the data still contains the so-called blending noise. The different approaches usually taken in deblending can be divided into two categories: deblending by denoising and deblending by inversion.

Unlike random Gaussian noise, blending interferences have amplitudes similar to the signal. Given its spatial incoherency in sorting coordinates such as common-receiver, common-offset, or common-midpoint domains, deblending via denoising approaches operates directly on the combed (pseudo-deblended) data. Different filtering techniques, such as the classical frequency-space prediction error filters (Spitz et al., 2008), low-rank filtering (Maraschini et al., 2012), and vector median filters (Huo et al., 2012), are therefore useful for source separation. However, deblending by denoising has limitations that can be minimized via deblending by inversion. The goal is to obtain the separated shots that honor the blended records while having features imposed by a regularizer. Usual choices for the regularization term are sparsity-inducing transforms, such as wavelet (Lin and Herrmann, 2009), seislet (Chen et al., 2014), Fourier (Abma et al., 2015), and Radon (Ibrahim and Trad, 2020) transforms. Similarly to reconstruction, prior models associated with low-rank matrices also provide alternative methods for deblending (Cheng and Sacchi, 2015; Kumar et al., 2015). While these methods exploit the randomness in the source firing times, different source encoding schemes lead to different deblending methods. Robertsson et al. (2016), for instance, uses a periodic sequence of source signatures to perform wavefield separation through signal apparition in the wavenumber domain. These and other methods are responsible for the success of simultaneous sources in the seismic industry.

There is a close connection between the applications discussed above, as they can be solved through similar models and methods. These problems also affect vector-valued data, but the abovementioned methodologies are mainly developed for scalar data processing. In multicomponent seismology, the components of the displacement vector-field are highly correlated. Such correlation can be exploited during processing to harness the full potential of these datasets. Therefore, this thesis defines the seismic data based on the displacement field's x, y, z vector components rather than rotating them. Specifically, it studies how to numerically represent these multivariate seismic measurements and exploit this data to its full potential in the applications discussed above from a vectorial perspective.

1.2 Vector-valued Seismic Data Processing

It is straightforward to notice that vectorial receivers offer a three-fold increase in data volume. However, one should not expect an equivalent increment in its processing complexity since one must now consider and preserve the vector fidelity of the data. In other words, processing the data components should follow an approach compatible with its vector nature. One reason to seek such compatibility is that much of the value of multicomponent seismic data lives in the differences and similarities among different components. It is common to read that a shift from scalar towards vector data processing should happen in the multicomponent seismology literature. This thesis discusses one possible way for doing so.

In that light, one of the attractive features of the componentwise processing approach is that the numerical representation of the data is trivially solved. Each component has its data samples represented as single numbers (real scalars) without considering the correlation of the data components, which represents a "waste" of valuable information. Such a representation could suffice in the case of a scalar quantity, such as P-wave datasets, and it is thus helpful in benchmarking. For vector-valued data, however, where more numbers are needed, the question of how to represent such data in a computer promptly arises.

1.2.1 Review of Previous Work

The numerical representation of vector-valued signals can be accomplished in different ways. One of such is to represent the vector-valued sample using the long-vector notation as

$$\hat{\mathbf{x}}_1[n] = \begin{bmatrix} \mathbf{x}_x[n] \\ \mathbf{x}_y[n] \\ \mathbf{x}_z[n] \end{bmatrix} \tag{1.1}$$

or

$$\hat{\mathbf{x}}_2[n] = \begin{bmatrix} \mathbf{x}_x[n] & \mathbf{x}_y[n] & \mathbf{x}_z[n] \end{bmatrix}, \tag{1.2}$$

where n represents a sample index of an arbitrary discretized vectorial quantity $\mathbf{x}$ with x, y, and z components composed of real ($\mathbb{R}$) or imaginary ($\mathbb{C}$) numbers. Examples in multicomponent seismology include Naghizadeh and Sacchi (2012) and Kamil et al. (2015), who have employed equation 1.1 to represent 3C ocean-bottom cable seismic data already in the frequency domain, and Zhang and van der Baan (2018) who used 1.2 to represent microseismic multicomponent samples in the time domain. Even though many contributions have emerged from this non-unique representation of vector signals, it lacks clarity. It is not clear, for instance, what sort of relation between components can be preserved or exploited during their processing. Moreover, many of data processing techiniques are based on

frequency-domain representations of the data, obtained through the Fourier transform. The long-vector notation might be restrictive in obtaining such a Fourier representation for these signals. Although it is still possible to Fourier transform each of the components independently, that introduces a scalar step in the vectorial processing of these signals. Therefore, while the long-vector notation is useful, the complex and hypercomplex representations of vector signals are known as natural and less restrictive signal representations.

It is important to realize that multicomponent seismology is one example of applied science that deals with x, y, and z components of a given signal. Other fields, such as communications and atmospheric sciences, also encounter vector-valued signals in their applications. For instance, while in the field of communications, the practitioner deals with x and y components of the electromagnetic wavefield (Schreier and Scharf, 2010), atmospheric scientists deal with the components of wind velocity profiles (Took and Mandic, 2009). It is then possible to benefit from their experience in how to numerically represent and process their data, where one shall encounter the concept of complex signals. For instance, an arbitrary two-component signal $(\mathbf{s}_x, \mathbf{s}_y) \in \mathbb{R}^{N\times 1}$ is naturally represented by $\hat{\mathbf{s}} \in \mathbb{C}^{N\times 1}$

$$\hat{\mathbf{s}}[n] = \mathbf{s}_x[n] + \mathbf{s}_y[n]i, \tag{1.3}$$

where $i^2 = -1$ is the imaginary unit. Complex-valued signals arise in several applications, and are mainly divided into inherent and designed (due to convenience of representation) complex signals. The former signals are real-world problems naturally described by complex numbers. The latter represent signals that, although real-world processes measured as real numbers, are best expressed and processed through complex analysis.

1.2.2 Complex-valued Signal Processing

Complex-valued signals show up in different signal processing applications. Their processing fuelled the development of complex-valued algorithms for the natural treatment of these quantities directly in the complex domain. Some basic concepts are straightforward to extend, as the statistical expectation operator ($E(\hat{\mathbf{s}}) = E(\mathbf{s}_x) + E(\mathbf{s}_y)i$), and these are well-discussed in textbooks such as Mandic and Goh (2009) and Schreier and Scharf (2010) which provide theoretical rigour to complex-valued data analysis. It is still usual to find that most complex-valued techniques represent a direct extension of their real-valued counterparts based on real domain statistics. Such an approach can be optimal only under the assumption that the signal is *proper*. Proper signals, also called second-order circular signals, are completely characterized by the standard covariance matrix ($E\{\hat{\mathbf{s}}\hat{\mathbf{s}}^H\}$) because of their vanishing complementary covariance matrices ($E\{\hat{\mathbf{s}}\hat{\mathbf{s}}^T\}$). In the case of non-circular (or improper) complex signals, on the other hand, it is possible to exploit the correlation

between data components if the standard covariance matrix is completed with the pseudo-covariance (complementary) matrices (Picinbono, 1994). This is referred to as augmented statistics because the signals are analyzed on an augmented basis that fully describes the second-order information of the data since both the standard covariance matrix and the complementary covariance matrix characterize the complex signal.

1.2.3 Quaternion-valued signal processing

For the case of vector-valued data with more than two components, as the examples given with 3C and 4C multicomponent seismology, the quaternion domain represents a valuable alternative for their processing. Quaternions are hypercomplex numbers invented by Sir William R. Hamilton (Hamilton, 1844, 1866) which, although defining a non-commutative algebra, allow for the numerical representation of 3C and 4C signals in a concise way. For example, using the quaternion signal model, a 2D-4C seismic dataset consisting of pressure ($\mathbf{P} \in \mathbb{R}^{N_t \times N_x}$) and three displacement components ($(\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z) \in \mathbb{R}^{N_t \times N_x}$) can be written as the full quaternion matrix $\hat{\mathbf{U}} \in \mathbb{H}^{N_t \times N_x}$ (see chapter 2)

$$\hat{\mathbf{U}}[it, ix] = \mathbf{P}[it, ix] + \mathbf{U}_x i[it, ix] + \mathbf{U}_y j[it, ix] + \mathbf{U}_z k[it, ix], \tag{1.4}$$

where $i^2 = j^2 = k^2 = ijk = -1$ are imaginary units, and $[it, ix]$ represent discrete time and space index. The natural ability of quaternions to represent multivariate signals has already been recognized, and it has been increasingly used among the signal processing community. The increased interest retools the field with new techniques based on its algebra, such as the quaternionic versions of the QR decomposition (Bunse-Gerstner et al., 1989), Fourier Transform (QFT) (Ell, 1992), and Singular Value Decomposition (QSVD) (Le Bihan and Mars, 2004). Such extensions allow the formulation of further signal processing techniques in the quaternion domain, and illustrate a shift from scalar towards vector signal processing.

Applications of quaternions in the geophysical literature can be traced back to the work of Davies et al. (1989) with integral transforms for vector fields. They can also be used to formulate Gabor filters helpful in detecting subtle differences in time-lapse analysis (Witten and Shragge, 2006) and for enhancing the anisotropic inversion of cross-dipole sonic logs (Zeng et al., 2018). The QFT is also a key tool in deblending scalar data via signal apparation (Andersson et al., 2017). In the context of multicomponent seismology, Krieger and Grigoli (2015) derive an optimal reorientation of multicomponent receivers based on the natural ability of quaternions to represent rotations by minimizing the energy along the transverse component. From the quaternion signal processing perspective, exciting applications have also been presented. For instance, Grandi et al. (2007) performs multicomponent velocity analysis using the quaternion signal model in the common mid-point domain, si-

multaneously obtaining velocity trends for different wave modes. Menanno and Mazzotti (2012) extended the classic Wiener deconvolution of seismic traces, spiky and predictive, to the case of multicomponent data through the quaternion model and least-squares theory. Stanton and Sacchi (2013) extend the projection-onto-convex-sets (POCS) reconstruction to the case of vector-valued multidimensional data by defining the radial and transverse components as a full quaternion in the frequency domain. The method enhanced S-wave splitting analysis and was expanded by Li et al. (2021) through the complexified QFT based on the whole frequency domain 3C dataset rather than 2C only. Sajeva and Menanno (2017) provides a two-component displacement model for Rayleigh waves and uses the QSVD to extract these modes from the data through rank-reduction after signal alignment. Similarly, Bahia et al. (2020) separates linear dispersive noise from reflection displacements using a 2C quaternionic model and sparsity constraints to perform the inversion. Finally, Zhao et al. (2020) introduces learned quaternion dictionaries for sparse representation of vector-valued signals, which were helpful in 3C denoising applications.

Using complex and hypercomplex variables and analysis to model real physical phenomena is an undeniable reality, but that also begs for clarity. It seems reasonable to wonder if it is correct to take advantage of the imaginary to represent real measurements. However, using a "unusual" algebra to analyze multivariate data also brings insights that usually do not emerge in a conventional (real number) analysis. Augmented statistics, for example, is one significant aspect that can be exploited using the complex or hypercomplex signal representation. Similarly to the complex case, it has been shown that exploiting the correlation between data components in a quaternion signal is also possible if complementary covariance matrices are considered in its processing. However, such manipulation with its conjugate is not possible for quaternion signals. In other words, if $\mathbf{U}$ is a quaternion signal, $\mathbf{U}$ and its conjugate ($\mathbf{U}^*$) are not enough to completely describe it, and a nontrivial extension of augmented statistics has been developed for this case (Took and Mandic, 2011). Similarly, the augmented quaternionic covariance provides a complete description of the second-order statistics of quaternion signals. It paves the way to the definition of a *Q-proper* quaternion signal, as discussed in later chapters of this thesis and summarized below.

1.3 Contributions of this Thesis

It is important to stress that this thesis does not suggest that quaternion-based techniques should replace existing seismic data processing methods. Quite the contrary, it calls attention to the possibilities that live in the theory of quaternion-based signal processing. In this light, this thesis contributes to the field of vector-valued seismic data processing in data recovery and regularization, random Gaussian noise attenuation, and simultaneous sources

separation from a quaternionic perspective. The hope is to benefit from existing methodologies in treating scalar seismic data and the novelties offered by quaternion-based signal processing to propose suitable strategies to process vector-valued seismic data. As such, the main contributions of this thesis are summarized as follows:

- Introduce a vector-valued seismic data processing framework that is physically meaningful and, therefore, fully vectorial. The quaternion signal model and quaternion-based signal processing provide the right environment for the development of a vector-based processing framework. Further, this thesis contains the initial attempts to introduce the use of augmented statistics in vector-valued seismic data processing to harness the acquired data fully.
- Explore the developments in quaternionic matrix decomposition and rank-reduction techniques in processing vector-valued seismic data. More specifically, the developments in Quaternion Singular Spectrum Analysis (QSSA) were extended to the case of vector-valued seismic data for applications in denoising and reconstruction. As rank-reduction is a known computationally-intensive problem, efficient quaternionic formulations for rank-reduction of structured matrices are also presented.
- Building upon the low-rank methods proposed above, strictly- and widely linear extensions of predictive filters are also presented in the case of vector-valued seismic data. These represent vector-valued versions of the classic frequency-space deconvolution (FXDECON) and make the most out of the multicomponent data as it fully exploits second-order information from the quaternionic signal. Denoising examples show that the methods have promising signal preservation capabilities while superior to their scalar counterpart.
- To further exploit the vast option of denoising techniques available, possibly including the ones presented in this thesis, the Regularization by Denoising (RED) framework was employed in solving the simultaneous source separation problem. The method can incorporate many denoising methods, and its performance was analyzed in seismic data deblending for both single and multicomponent common-receiver gathers.

1.4 Thesis Overview

Chapter 2 provides an overview of quaternion numbers, quaternionic signals, and their properties. It thus discusses relevant aspects of quaternion-based signal processing, such as augmented second-order statistics, and also includes the QFT and QSVD, which are crucial to the developments in this thesis.

Chapter 3 introduces the QSSA as a tool for vector-valued denoising and reconstruction of multicomponent (3C) seismic data, which is represented as a quaternion signal in the time domain. Its $f - x$ representation is obtained through the QFT, and the low-rank character of the Hankel matrices built from the quaternionic frequency slices is shown to be preserved. Augmented second-order statistics in the QSSA (AQSSA) provide a clear way to use second-order information from quaternionic signals, improving results over the SSA and QSSA. Following the scalar case, it is possible to implement the QSSA filter efficiently. Such implementation is based on the spectral convolution theorem, which has a nontrivial extension to the quaternion domain. These efficient extensions were vital in reconstructing multidimensional vector-valued seismic data through rank-reduction.

Chapter 4 uses quaternion auto-regressive (Q-AR) models, again in conjunction with the QFT, to develop a hypercomplex version of the $f - x$ prediction filter named the QFXDECON. For complete second-order signal characterization, where the correlation between data components can be fully exploited, augmented statistics is introduced through a widely-linear predictive model of the quaternionic frequency slices. The contribution is introduced as the WL-QFXDECON and includes additional useful information from the data components in its formulation. The methods retain the FXDECON's simplicity and efficiency and exemplify the possibility of developing vector-valued data processing methods that are physically consistent and make the most out of the acquired data. Its underlying augmentation step creates a four-fold redundancy in the data samples for signal and noise alike, which provides better signal preservation in synthetic and field data examples.

Chapter 5 discusses general inverse problems and briefly focuses on the related Plug-and-Play priors and Regularization by Denoising (RED) frameworks. RED is a unique framework that explicitly employs denoisers in its cost function, but its gradient is only valid under some conditions imposed on the denoisers. These conditions are summarized and discussed in this chapter, along with first-order optimization methods used to minimize its cost function. The RED framework is then introduced for scalar and vector-valued simultaneous sources separation applications. The formulation leads to deblending algorithms that can use many denoising techniques, including those developed in Chapters 3 and 4. The denoisers are used to implicitly impose desired priors, such as sparsity or low-rankness in transform domains, which are favorable to applications such as deblending and reconstruction. The denoisers tested in this chapter are the frequency-wavenumber thresholding (FKT) and the SSA in the scalar case and their respective quaternionic extensions in the case of vector-valued data. The synthetic and field numerically blended examples show RED competitive performance with leading methods for both scalar and vector cases.

Chapter 6 concludes this thesis and provides a summary of its contributions and limitations that can be explored in future research.

CHAPTER 2

Quaternions

2.1 Introduction

Quaternions are hypercomplex numbers of four dimensions (4D) introduced by Sir William Rowan Hamilton in an effort to extend complex numbers to higher dimensions. Its inherent 4D structure provides an intuitive framework for representing three- and four-component signals, yielding a unified processing approach for vector data processing. The following introduces the necessary definitions of quaternion algebra to provide enough background for defining quaternion signals and their subsequent processing techniques. To this point, there are thorough descriptions of quaternions available in the literature, such as the work of Hamilton himself (Hamilton, 1844, 1866) and Ward (2012), and focusing on signal and image processing, Hitzer and Sangwine (2013) and Ell et al. (2014).

2.2 Quaternion Algebra

2.2.1 Fundamentals

The following description is based on Ell et al. (2014) for their use of quaternions to represent the three components (red, green, and blue) of color images. A quaternion variable, $q \in \mathbb{H}$, can be defined as a four-dimensional hypercomplex number with Cartesian form given by

$$q = a + bi + cj + dk, \tag{2.1}$$

where the 4-tuple $\{a, b, c, d\} \in \mathbb{R}$ are its components. A quaternion can be decomposed as a scalar plus a vector part following

$$q = S(q) + V(q), \tag{2.2}$$

where its scalar part corresponds to its only purely real component

$$S(q) = a, \tag{2.3}$$

and its vector part, which consists of three imaginary components, as follows

$$V(q) = bi + cj + dk = q - S(q). \tag{2.4}$$

A quaternion is said to be pure if $S(q) = 0$, and the set of pure quaternions can be denoted as $V(\mathbb{H})$. Quaternions with null vector part are simply identified as elements of $\mathbb{R}$, which makes it noticeable that the sets of real and complex numbers are special cases of quaternions, i.e., $\mathbb{R} \in \mathbb{C} \in \mathbb{H}$. Similarly to the complex case, the imaginary units (i, j, k) follow the rules

$$i^2 = j^2 = k^2 = ijk = -1, \tag{2.5}$$

and additional relations can be drawn from 2.5 through left- and right-sided multiplications of ijk with other units (i, j, k), which are given by

$$\begin{aligned} ij &= -ji = k, \\ ki &= -ik = j, \\ jk &= -kj = i. \end{aligned} \tag{2.6}$$

Notice how the imaginary units themselves are pure quaternions, e.g. $i = 0 + 1i + 0j + 0k$, and represent the orthogonal unit vectors in three dimensions (3D).

The algebraic operations of addition and multiplication of quaternion numbers are defined in analogy to its complex counterpart. For instance, if $q = a+bi+cj+dk$ and $p = e+fi+gj+hk$, their sum is given by

$$\begin{aligned} q + p = & \quad S(p) + V(p) + S(q) + V(q) \\ = & \quad (a + e) + (b + f)i + (c + g)j + (d + h)k, \end{aligned}$$

and their product can be written as

$$\begin{aligned} qp = & \quad (S(q)+V(q))(S(p)+V(p)) \\ = & \quad S(q)S(p)+S(q)V(p)+V(q)S(p)+V(q)V(p). \end{aligned} \tag{2.7}$$

Using the relations in 2.5 and 2.6, the product $V(q)V(p)$ can be rewritten as

$$\begin{aligned} V(q)V(p) = & \quad (bi+cj+dk)(fi+gj+hk) \\ = & \quad bi(fi+gj+hk)+cj(fi+gj+hk)+dk(fi+gj+hk) \\ = & \quad -bf-cg-dh+i(ch-dg)+j(df-bh)+k(bg-cf), \end{aligned} \tag{2.8}$$

where it is possible to recognize the inner (or scalar) product $\langle V(q), V(p)\rangle = bf+cg+dh$ and the vector cross product $V(q)\otimes V(p) = i(ch-dg)+j(df-bh)+k(bg-cf)$. It is also possible to conclude that the product of two pure quaternions usually results in a full quaternion. More importantly, the quaternion product contains a vector product which is known to be anticommutative. Its final expression is written as

$$qp = S(q)S(p) - \langle V(q), V(p)\rangle + S(q)V(p) + V(q)S(p) + V(q)\otimes V(p), \tag{2.9}$$

leading to the main difference between quaternion and usual algebra: multiplication in $\mathbb{H}$ is, in general, noncommutative ($qp \neq pq$). Despite its noncommutativity, the quaternion product is associative and distributive, as used to develop the expression in 2.8. Finally, quaternions commute with real numbers, as $V(q)S(p) = S(p)V(q)$ for instance, noticing that the operators $S(\cdot)$ and $V(\cdot)$ extract the scalar and vector part of a quaternion, respectively.

Even though noncommutativity might have important implications in the development of quaternion-based applications, once it is kept in mind and adequately accounted for, the implementation of quaternion-based operations and techniques can be arguably straightforward. In that light, when manipulating quaternion numbers, it is common to hope that classical definitions for real or complex numbers are trivial extensions to the quaternion case. However, there are specific properties for the quaternion domain which do not arise in its complex counterpart, and some extensions, albeit not straightforward nor trivial, are worthy of being studied. The relevant ones for the scope of this thesis are presented next.

2.2.2 Properties

Some properties of quaternions that are useful for future discussion of this thesis include the conjugate of a quaternion, obtained by negating its imaginary part

$$q^* = S(q) - V(q) = a - bi - cj - dk, \tag{2.10}$$

thus paving the way to the definition of its norm ($\|\cdot\|$) and modulus ($|\cdot|$)

$$|q| = \|q\|^{1/2} = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{2.11}$$

If $|q| = 1$, q is a unitary quaternion, as the example of the (i, j, k) imaginary units. A related concept to conjugation that arises in the quaternion domain but has no counterpat in the complex case is the concept of involutions, or self-inverse mappings, which will show themselves to be crucial in future chapters. Unlike conjugation, quaternion involutions are defined with respect to an arbitrary pure unitary quaternion μ, and it is written in general form as

$$q^\mu = -\mu q \mu,$$

with $\mu \in V(\mathbb{H})$ and $|\mu| = 1$. The canonical case corresponds to setting μ as one of the standard basis elements (i, j, k), and represent rotations along a single imaginary axis

$$q^i = -iqi = a + bi - cj - dk, \tag{2.12}$$

$$q^j = -jqj = a - bi + cj - dk, \text{ and} \tag{2.13}$$

$$q^k = -kqk = a - bi - cj + dk. \tag{2.14}$$

Notice how it is possible to linearly relate a quaternion, its conjugate and involutions by

$$q^* = \frac{1}{2}(q^i + q^j + q^k - q), \tag{2.15}$$

as well as to uniquely express the four real components of q through the expressions

$$a = \frac{1}{2}(q + q^*) \qquad b = \frac{1}{2i}(q - q^{i*}) \tag{2.16}$$

$$c = \frac{1}{2j}(q - q^{j*}) \qquad d = \frac{1}{2k}(q - q^{k*}). \tag{2.17}$$

2.2.3 Representations

In addition to the Cartesian form of a quaternion given in equation 2.1, there are other representations of quaternions that are particularly useful in the development and study

of hypercomplex techniques. Examples are polar and Cayley-Dickson (CD) forms of a quaternion, which are presented next.

Polar Form

The polar form of a quaternion is equivalent of the Euler formula, and is given by

$$q = |q|e^{\mu_q \phi_q} = |q|(\cos\phi_q + \mu_q \sin\phi_q) \tag{2.18}$$

where its modulus ($|q|$) is given by 2.11, the phase ($\phi_q \in \mathbb{R}^+$)

$$\phi_q = \arctan\left(\frac{|V(q)|}{S(q)}\right), \tag{2.19}$$

where $\phi_q = \pi/2$ if $S(q) = 0$, and the eigenaxis ($\mu_q \in V(\mathbb{H})$) is written as

$$\mu_q = \frac{V(q)}{|V(q)|}, \tag{2.20}$$

where, since $|V(q)| = \sqrt{b^2 + c^2 + d^2}$, μ_q is a pure unit quaternion. For one such arbitrary (pure and unitary) quaternion, the Euler representation given in equation 2.18 would be written as

$$e^{\mu\phi} = \cos\phi + \mu\sin\phi, \tag{2.21}$$

where its real part is given by $\cos\phi$ and vector part $\mu\sin\phi$. Since taking a quaternion conjugate means negating its vector part, i.e. $\cos\phi - \mu\sin\phi = e^{-\mu\phi}$, this can be achieved by either negating its phase (ϕ) or reverting its axis (μ). Finally, it is also possible to obtain the expressions $\cos\phi = \frac{1}{2}\left(e^{\mu\phi} + e^{-\mu\phi}\right)$ and $\sin\phi = -\frac{\mu}{2}\left(e^{\mu\phi} - e^{-\mu\phi}\right)$.

Other polar forms of quaternions can be devised, and although not the focus of this thesis, some of these representations found useful applications in signal and image processing. Since quaternions constitute a 4D algebra, their polar forms represent a signal in terms of magnitude and phase. While the magnitude is a single real number, the phase can only be fully described by three real numbers. For instance, one representation worthy of being mentioned is connected to the Euler angle parametrization of quaternions following the expression

$$q = |q|e^{\eta i}e^{\kappa j}e^{\phi k}, \tag{2.22}$$

where the angles of q, $\eta \in [0, 2\pi)$, $\kappa \in [0, \pi)$, $\phi \in [0, 2\pi)$, are identified as three phases. Although useful, such polar representation of quaternions goes beyond the purposes of this thesis, which mainly focuses on the Euler representation of quaternions as a hypercomplex exponential given in equation 2.18. In the next subsection, another useful representation

of quaternions is introduced, namely the Cayley-Dickson representation of a quaternion number.

Cayley-Dickson Form

In their Cartesian form, quaternions are represented by four real components, and a similar argument can be made to represent a quaternion as a pair of complex numbers. That representation is known as the Cayley-Dickson (CD) form of a quaternion, and it splits the quaternion into two different (orthogonal) complex planes where q is written as

$$q = z_1 + z_2 j, \tag{2.23}$$

where $z_1 = a + bi$ and $z_2 = c + di$. Notice that $\{z_1, z_2\} \in \mathbb{C}_i$, the set of complex numbers defined with $i^2 = -1$. Therefore, the CD form allows the interpretation of a quaternion in terms of complex numbers. It is very convenient because it facilitates understanding quaternions and their properties through better-understood complex algebra.

The CD representation can be generalized to the symplectic decomposition of quaternions in which a quaternion q is still written as a pair of complex numbers. However, q is first projected in a new basis of interest $\{1, \mu, \nu, \mu\nu\} \in \mathbb{H}$ to obtain q' as

$$q' = a' + b'\mu + c'\nu + d'\mu\nu, \tag{2.24}$$

with decomposition is given similarly to that in equation 2.23 as

$$q' = (a' + b'\mu) + (c' + d'\mu)\nu. \tag{2.25}$$

The complex units $\{\mu, \nu, \mu\nu\}$ follow the same rules as in 2.5 and 2.6

$$\mu^2 = \nu^2 = (\mu\nu)^2 = -1, \tag{2.26}$$

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

$$\begin{aligned} a' &= a \\ \begin{pmatrix} b' \\ c' \\ d' \end{pmatrix} &= \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 \\ \nu_1 & \nu_2 & \nu_3 \\ \mu\nu_1 & \mu\nu_2 & \mu\nu_3 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \end{pmatrix}. \end{aligned} \tag{2.27}$$

The quaternion given by 2.25 is decomposed into its simplex and perplex parts

$$q' = q_s + q_p\nu, \tag{2.28}$$

with $q_s = (a' + b'\mu)$ and $q_p = (c' + d'\mu)$. Both simplex and perplex parts of a quaternion in its symplectic (and CD) form are isomorphic to complex numbers. This is significant because, since $\{q_s, q_p\} \in \mathbb{C}_\mu$, many properties and insights from complex algebra are used to facilitate the development of quaternion-based techniques (Ell and Sangwine, 2007).

Other Representations

There are other possible representations of quaternions that do not fit the scope of this work. For instance, it is possible to define a polar Cayley-Dickson representation of quaternions with complex-valued modulus and phase. These representations are said to be "scalar" because they still work around representing a single number via parametrization by, for instance, modulus and phase. That is in contrast with matrix representations where, for instance, a quaternion number q can be written as the 4×4 real matrix $M_{\mathbb{R}}(q)$

$$M_{\mathbb{R}}(q) = \begin{pmatrix} a & b & c & d \\ -b & a & -d & c \\ -c & d & a & -b \\ -d & -c & b & a \end{pmatrix}, \tag{2.29}$$

or through its 2×2 complex matrix representation $M_{\mathbb{C}}(q)$

$$M_{\mathbb{C}}(q) = \begin{pmatrix} z_1 & -z_2^* \\ z_2 & z_1^* \end{pmatrix}. \tag{2.30}$$

Therefore, the matrix representation makes it possible to explore quaternion algebra from a known perspective as the quaternion product is equivalent to the regular matrix product of their matrix representations. However, there are practical aspects to the matrix representation of quaternions that make its adoption less attractive. For instance, while the representation in equation 2.29 requires storing 16 real values, the cartesian form in equation 2.1 requires only four. Even more impactful is the increase in computational demand; while the quaternion product in equation 2.9 already increases the number of real-valued multiplications and additions to 16 and 12, respectively, its real-valued matrix representation increases it to 64 and 48. In both cases, an increase by a factor of four rules out the adoption of matrix representations in the practical developments of this thesis. On the other hand, the matrix representations of quaternions have their theoretical advantages when studying, for

instance, matrices whose elements consist of quaternion numbers, as introduced in the next section. As it will become clear, mappings between quaternion-valued and complex-valued matrices are helpful when developing quaternion-based methods.

2.2.4 Matrices and Vectors of Quaternions

The definitions above facilitate the description of vectors and matrices of quaternions used to represent signals recorded by vector sensors. However, it is necessary to clarify that the elements of given matrices and vectors can be drawn from different sets, meaning that matrices can arrange real, complex, or quaternion numbers. While matrices of real or complex numbers are better known and understood, matrices of quaternion numbers do not lag much behind, with studies going back to Wolf (1936). Of particular importance for this section are the work of Zhang (1997) and Le Bihan and Mars (2004), for instance, which bridge a considerable gap in understanding quaternion matrices mainly through their complex representation. Again, while several classical definitions can be easily extended to the quaternion domain, for instance, scalar product, norm, and the distance between two vectors, some operations for real and complex arrays do not hold in the quaternion domain.

Intuitively, a matrix $\mathbf{X} \in \mathbb{H}^{N \times M}$ has elements that belong to the set of quaternion numbers ($x_{i,j} \in \mathbb{H}$, for $i = [1, N]$ and $j = [1, M]$). One can write $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_M]$, with $\mathbf{x}_i \in \mathbb{H}^{N \times 1}$, $i = [1, M]$, defining a length-N column vector of quaternions. Similarly to the complex case, the scalar product between two quaternion vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^H \mathbf{y}, \tag{2.31}$$

where the superscript ^{H} denotes the conjugate transpose of a quaternion vector. The ℓ_2 norm of a quaternion vector is then given by

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^H \mathbf{x}}, \tag{2.32}$$

which allows for the definition of a distance measure between quaternion vectors $\mathbf{x}$ and $\mathbf{y}$ as

$$\|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{(\mathbf{x} - \mathbf{y})^H (\mathbf{x} - \mathbf{y})}. \tag{2.33}$$

These definitions are straightforward extensions of real or complex matrix algebra routinely used in signal processing. Further, it is also possible to exploit the representations of quaternions discussed in section 2.2.3 in the context of vectors and matrices of quaternions.

In particular, the complex representation (CD form) of vectors and matrices of quaternions is rather useful. The CD representation of a quaternion vector $\mathbf{x} \in \mathbb{H}^{N \times 1}$ is given by

$\mathbf{x} = \mathbf{x}_s + \mathbf{x}_p j$, where $\{\mathbf{x}_s, \mathbf{x}_p\} \in \mathbb{C}_i^{N\times 1}$, and its complex representation $\chi_{\mathbf{x}} \in \mathbb{C}^{2N}$ follows

$$\chi_{\mathbf{x}} = \begin{pmatrix} \mathbf{x}_s \\ -\mathbf{x}_p^* \end{pmatrix}. \tag{2.34}$$

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

$$\chi_{\mathbf{X}} = \begin{pmatrix} \mathbf{X}_s & \mathbf{X}_p \\ -\mathbf{X}_p^* & \mathbf{X}_s^* \end{pmatrix}. \tag{2.35}$$

Such notation of quaternion matrices makes it possible, for instance, to employ complex decomposition algorithms to manipulate quaternion matrices (Le Bihan and Mars, 2004). The ability to decompose or transform quaternion matrices is fundamental to the definition of quaternionic models to represent vector signals and their processing.

2.3 Quaternion Signal Processing

The discussion in Chapter 1 illustrated possible approaches to processing vector signals. As stated before, this thesis avoids using componentwise or long-vector notation for vector signal processing. Instead, quaternion arrays naturally define vector signals, allowing for simultaneous and concise processing.

For the case of color images, as an example, where each pixel of the image represents three orthogonal components, here denoted by $(\mathbf{R}, \mathbf{G}, \mathbf{B}) \in \mathbb{R}^{N\times M}$, at N and M discrete positions, its possible to encode the three corresponding signals into the components of a pure quaternion matrix $\mathbf{Q} \in V(\mathbb{H}^{N\times M})$ such as

$$\mathbf{Q}[n, m] = \mathbf{R}[n, m]i + \mathbf{G}[n, m]j + \mathbf{B}[n, m]k, \tag{2.36}$$

with $n = [1, N]$ and $m = [1, M]$. As it is possible to notice, $\mathbf{Q}$ holistically represents the color image, naturally accounting for the mutual information between data components. It is also possible to use quaternion arrays to accommodate two pairs of complex signals, as it follows from its symplectic decomposition, and process them simultaneously. In this case, an arbitrary two-component signal, such as $\tilde{\mathbf{U}}_x$ and $\tilde{\mathbf{U}}_y \in \mathbb{C}_i^{N\times M}$, can be represented by means of the Cayley-Dickson form of full quaternion matrices ($S(\mathbf{Q}) \neq \mathbf{0}$) as

$$\mathbf{Q}[n, m] = \tilde{\mathbf{U}}_x[n, m] + \tilde{\mathbf{U}}_y[n, m]j. \tag{2.37}$$

Above, for instance, $\{\tilde{\mathbf{U}}_x, \tilde{\mathbf{U}}_y\} \in \mathbb{C}_i^{N\times M}$ could have been obtained through the classical Fourier transform of the real components, say $\mathbf{U}_x$ and $\mathbf{U}_y$, for which the better understood classic Fourier pairs separately hold for each of the signal components.

The quaternionic signal model can handle multicomponent signals, but it calls for the generalization of classical tools in signal processing, such as the Fourier transform, to their hypercomplex versions. Such extensions also include theoretical developments in statistical signal processing of quaternion-valued data, which, as might be expected, draws essential lessons from the developments made in processing complex-valued signals. The most important, perhaps, is that the standard second-order quaternionic characterization of vector signals does not fully characterize the quaternionic signal, and it needs to be augmented by complementary covariance matrices obtained as a function of its involutions. The following sections discuss the quaternionic extension of signal processing techniques commonly used in their real and complex versions. In particular, the following discussion focuses on the statistical description of quaternion signals, its corresponding versions of auto-regressive models, Fourier Transforms, and matrix factorization techniques.

2.3.1 Augmented Second-order Statistics

Since quaternions are a (hypercomplex) extension of complex numbers, it is natural that most of the quaternion signal processing theory is rooted in its complex counterpart. That includes the developments on second-order statistics of complex-valued signals, with contributions to the theory of improper signals by Picinbono (1994, 1996) and Picinbono and Chevalier (1995), and the references cited therein. In the quaternionic context, the work by Took and Mandic (2011) provides the theoretical rigor on which this thesis is based. The authors point out that the standard covariance matrix of quaternion-valued processes does not guarantee maximum use of the available information in a quaternion signal.

In the case of complex signals, for instance, it is possible to exploit the correlation between data components if the standard covariance matrix is completed with the complementary covariance matrices (Picinbono, 1996). This is referred as augmented statistics because the signals are usually analyzed in an augmented basis where a length-N complex vector, say $\mathbf{x} \in \mathbb{C}^{N\times 1}$, is augmented with its complex conjugate as

$$\mathbf{x}^a = \begin{pmatrix} \mathbf{x} \\ \mathbf{x}^* \end{pmatrix}.$$

Analyzing the augmented covariance matrix obtained from $\mathbf{x}^a$, one obtains

$$\mathbf{C}^a = E\{\mathbf{x}^a\mathbf{x}^{aH}\} = E\left\{\begin{pmatrix} \mathbf{x}\mathbf{x}^H & \mathbf{x}\mathbf{x}^T \\ (\mathbf{x}\mathbf{x}^T)^* & (\mathbf{x}\mathbf{x}^H)^* \end{pmatrix}\right\}, \tag{2.38}$$

where $E\{\cdot\}$ represents the expectation operator. Notice that $\mathbf{C}^a$ fully describes the second-order information of the signal because both the standard covariance matrix ($E\{\mathbf{x}\mathbf{x}^H\}$) and the complementary covariance matrix ($E\{\mathbf{x}\mathbf{x}^T\}$) characterize the complex signal.

For quaternion signals, however, such manipulation with its conjugate is not possible, *i.e.*, $\mathbf{x}$ and $\mathbf{x}^*$ are not enough to completely describe $\mathbf{x}$, if $\mathbf{x}$ is a quaternion signal. Nevertheless, it has been shown that one can exploit the correlation between data components in a quaternion signal if complementary covariance matrices are considered in its processing (Took and Mandic, 2011). By noticing that to fully express the four components of a quaternion-valued signal one has to use its four involutions (see equations 2.17), Took and Mandic (2011) propose its processing in an augmented basis such as

$$\mathbf{x}^a = \begin{pmatrix} \mathbf{x} \\ \mathbf{x}^i \\ \mathbf{x}^j \\ \mathbf{x}^k \end{pmatrix} \tag{2.39}$$

if $\mathbf{x}$ is a quaternion vector. Similarly to the complex case, the covariance matrix of the augmented quaternion vector $\mathbf{x}^a$ is then given by

$$E\{\mathbf{x}^a\mathbf{x}^{aH}\} = E\left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{x}^i \\ \mathbf{x}^j \\ \mathbf{x}^k \end{pmatrix} \begin{pmatrix} \mathbf{x}^* & \mathbf{x}^{i*} & \mathbf{x}^{j*} & \mathbf{x}^{k*} \end{pmatrix} \right\} = \tag{2.40}$$

$$\begin{pmatrix} E\{\mathbf{x}\mathbf{x}^*\} & E\{\mathbf{x}\mathbf{x}^{i*}\} & E\{\mathbf{x}^{j*}\} & E\{\mathbf{x}\mathbf{x}^{k*}\} \\ E\{\mathbf{x}^i\mathbf{x}^*\} & E\{\mathbf{x}^i\mathbf{x}^{i*}\} & E\{\mathbf{x}^i\mathbf{x}^{j*}\} & E\{\mathbf{x}^i\mathbf{x}^{k*}\} \\ E\{\mathbf{x}^j\mathbf{x}^*\} & E\{\mathbf{x}^j\mathbf{x}^{i*}\} & E\{\mathbf{x}^j\mathbf{x}^{j*}\} & E\{\mathbf{x}^j\mathbf{x}^{k*}\} \\ E\{\mathbf{x}^k\mathbf{x}^*\} & E\{\mathbf{x}^k\mathbf{x}^{i*}\} & E\{\mathbf{x}^k\mathbf{x}^{j*}\} & E\{\mathbf{x}^k\mathbf{x}^{k*}\} \end{pmatrix}, \tag{2.41}$$

where $E\{\cdot\}$ denotes the statistical expectation operator. It is relevant to mention that the complementary matrices, given by $E\{\mathbf{x}^\alpha\mathbf{x}^{\beta H}\}$ ($\alpha, \beta = \{1, i, j, k\},\ \alpha \neq \beta$), capture the cross-correlation between the components of the vector signal. Thus, the augmented covariance (equation 2.41) provides a full description of the second-order statistics of $\mathbf{x}$. These matrices pave the way to the definition of a Q-proper, or only proper, quaternion signal.

The concept of a proper signal in the complex domain refers to those signals with vanishing complementary covariance matrices (Picinbono, 1994). This implies two important conditions for properness of complex-valued signals: the components of the signal are uncorrelated, and have the same variance. In the quaternion domain, by continuity, a proper signal should also satisfy these conditions. Therefore, a Q-proper signal has uncorrelated components (*i.e.*, $E\{\mathbf{x}^\alpha\mathbf{x}^{\beta H}\} = 0;\ \ \alpha, \beta = \{1, i, j, k\},\ \alpha \neq \beta$), and each of its components

has the same variance (Took and Mandic, 2011).

2.3.2 Linear Prediction and Autoregressive Modeling

Linear prediction theory allows determining the value of a signal at a sample n, say x_n, using linear combinations of its past K samples as

$$\hat{x}(n) = \sum_{k=1}^{K} x(n-k) f_k, \tag{2.42}$$

which is usually written in vector form, assuming a real signal, as

$$\hat{\mathbf{x}}(n) = \mathbf{f}^T \mathbf{x}(n-1), \tag{2.43}$$

for a vector $\mathbf{x}(n) = [x(n)\ x(n-1)\ ...\ x(n-K+1)]^T$, and a length-$M$ prediction filter $\mathbf{f}$. The standard extension of this linear model to complex or quaternion numbers is written as

$$\hat{\mathbf{x}}(n) = \mathbf{f}^H \mathbf{x}(n-1), \tag{2.44}$$

where it should be clear that the vectors, say $\mathbf{x}$ and $\mathbf{f}$, now contain either complex or quaternion numbers. However, following the discussion in subsection 2.3.1, it might not be sufficient to replace the real variables in equation 2.42 by quaternion variables. Rather, similarly to the complex case, augmented statistics suggests adopting linear models for quaternion signals based on the use of its involutions, thus called widely-linear, as follows

$$\hat{\mathbf{x}}(n) = \mathbf{f}^{aH} \mathbf{x}^a(n-1) = \mathbf{f}^H \mathbf{x}(n) + \mathbf{f}_i^H \mathbf{x}^i(n) + \mathbf{f}_j^H \mathbf{x}^j(n) + \mathbf{f}_k^H \mathbf{x}^k(n). \tag{2.45}$$

Here the notation in $\mathbf{f}_{i,j,k}$ only denotes the filter associated with the given involution rather than an index or, unlike $\mathbf{x}^{i,j,k}$, an involution itself.

A related concept to that of linear prediction is that of autoregressive (AR) models. For instance, since equation 2.46 is a regression problem based on its own past samples, this model can also be regarded as an zero-mean K-order AR model. More specifically, such a model satisfies the difference equation

$$x_n + x_{n-1} f_1 + x_{n-2} f_2 + ... + x_{n-K} f_K = e_n, \tag{2.46}$$

where e_n is an innovation term, here taken as white Gaussian noise, and f_k are the filter

coefficients to be determined in an optimal sense. Compactly, one can write

$$x_n = \sum_{k=1}^{K} x_{n-k} f_k + e_n, \tag{2.47}$$

which represents the convolution of the data sequence, x_{n-k}, with the filter coefficients f_k. Equation 2.47 characterizes a K-order *forward* AR model since it can be used to recursively obtain the predictable portions of the signal x_n from the current sample n based on its previous K samples. Similarly, one can define a *backward* recursion formula

$$x_n = \sum_{k=1}^{K} x_{n+k} b_k + e_n, \tag{2.48}$$

to predict the current sample n based on its future K samples, but now from the convolution of the data sequence, x_{n+k}, with the filter coefficients b_k.

Assuming a signal of length N, equations 2.47 and 2.48 can be written in matrix form as

$$\mathbf{d} = \mathbf{X}\mathbf{f} + \epsilon, \tag{2.49}$$

where $\mathbf{X}_{M\times K}$ is a transient-free convolution matrix built from the data samples, $\mathbf{d}_{M\times 1}$ is the data vector, $\mathbf{f}_{K\times 1}$ is the vector of filter coefficients (forward or backward), $\epsilon_{M\times 1}$ is the vector of innovation terms, and $M = N - K + 1$ (Ulrych and Sacchi, 2005).

Autoregressive models are of interest to many fields of science, and their extension to complex- and quaternion-defined processes was natural step in signal processing. The standard extension from real to complex- or quaternion-valued AR models also follows equations 2.47 and 2.48, but this is already known to be suboptimal for improper signals as useful additional information is being ignored. Similarly to the widely linear prediction model in equation 2.45, a widely linear AR model is defined (Ginzberg and Walden, 2013) by allowing additional linear operations on the quaternion involutions in equation 2.49 as

$$\mathbf{d} = \mathbf{X}\mathbf{f} + \mathbf{X}^i\mathbf{f}_i + \mathbf{X}^j\mathbf{f}_j + \mathbf{X}^k\mathbf{f}_k + \epsilon, \tag{2.50}$$

which can be recast as

$$\mathbf{d} = \mathbf{X}^a\mathbf{f}^a + \epsilon \tag{2.51}$$

with $\mathbf{X}^a = \begin{pmatrix}\mathbf{X} & \mathbf{X}^i & \mathbf{X}^j & \mathbf{X}^k\end{pmatrix}$ and $\mathbf{f}^a = \begin{pmatrix}\mathbf{f} & \mathbf{f}_i & \mathbf{f}_j & \mathbf{f}_k\end{pmatrix}^T$. Note that $\mathbf{X}, \mathbf{X}^i, \mathbf{X}^j$, and $\mathbf{X}^k$ are transient-free convolution matrices; then the product $\mathbf{X}^{aH}\mathbf{X}^a$ provides an estimator of the augmented covariance matrix of the type given in equation 2.41.

2.3.3 Matrix Factorization

Singular Value Decomposition

Quaternion singular value decomposition (QSVD) is a fundamental tool when treating vector signals. Proof of the existence of QSVD is given in Zhang (1997) or Loring (2012), and it is based on the polar decomposition of quaternion matrices. Possible implementations of the QSVD are discussed by Le Bihan and Mars (2004) and Sangwine and Le Bihan (2006), which further allows the definition of quaternionic signal processing techniques based on subspace methods. To be specific, Le Bihan and Mars (2004) compute the QSVD through the complex-adjoint representation of a quaternion matrix (equation 2.35), but its description follows the real or complex case, as discussed below.

Similarly to real- or complex-valued matrices, the rank of a quaternion matrix $\mathbf{X} \in \mathbb{H}^{N\times M}$ can be defined as the maximum number of its linearly independent columns or rows, which can be understood as the minimum number of rank-1 quaternion matrices that result in $\mathbf{X}$ by linear combination. Formaly, a rank-k quaternion matrix $\mathbf{X}$ has SVD written as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = \mathbf{U}\begin{pmatrix}\mathbf{\Sigma}_k & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^H \tag{2.52}$$

where $\mathbf{\Sigma}_k \in \mathbb{R}^{k\times k}$ contains the k singular values of $\mathbf{X}$, arranged in decreasing magnitude order, on its main diagonal. As in the real or complex cases, $\mathbf{U} \in \mathbb{H}^{N\times N}$ and $\mathbf{V} \in \mathbb{H}^{M\times M}$ are unitary quaternion matrices (so $S(\mathbf{U}\mathbf{U}^H) = S(\mathbf{V}\mathbf{V}^H) = \mathbf{I}$, and $V(\mathbf{U}\mathbf{U}^H) = V(\mathbf{V}\mathbf{V}^H) = \mathbf{0}$) which contain the left and right singular vectors of $\mathbf{X}$, respectively. Hence, as mentioned above, the QSVD can be used to write $\mathbf{X}$ as the sum of k rank-1 quaternion matrices as

$$\mathbf{X} = \sum_{n=1}^{k} \sigma_n \mathbf{u}_n \mathbf{v}_n^H, \tag{2.53}$$

where σ_n represents the n-th (real-valued) singular value associated with $\mathbf{u}_n$ (n-th column of $\mathbf{U}$) and $\mathbf{v}_n$ (n-th column of $\mathbf{V}$), the left and right singular vectors of $\mathbf{X}$, respectively. Other useful properties for discussing the QSVD in the following sections are (Zhang, 1997):

- The rank of a quaternion matrix $\mathbf{X}$ is k if $\mathbf{X}$ has k non-zero singular values, and the QSVD is thereby said to be rank revealing, as follows from equation 2.52.
- The rank of a quaternion matrix $\mathbf{X}$ is k if and only if the rank of its complex adjoint $\chi_{\mathbf{X}}$ is $2k$.

As already mentioned, it is possible to compute the QSVD using the equivalence between quaternion matrices ($\mathbb{H}^{N\times M}$) and its complex adjoint matrix $\chi_{\mathbf{X}} \in \mathbb{C}^{2N\times 2M}$ (Le Bihan and

Mars, 2004), which allows the SVD of a quaternion matrix $\mathbf{X}$ to be obtained from the SVD of $\chi_{\mathbf{X}}$ (equation 2.35) using classical algorithms. The (classic) SVD of $\chi_{\mathbf{X}}$ is given by

$$\chi_{\mathbf{X}} = \mathbf{U}^{\chi\mathbf{x}} \begin{pmatrix} \mathbf{\Sigma}_{2k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} (\mathbf{V}^{\chi\mathbf{x}})^H = \sum_{n'=1}^{2k} \sigma_{n'} \mathbf{u}_{n'}^{\chi\mathbf{x}} \mathbf{v}_{n'}^{\chi\mathbf{x}H}, \tag{2.54}$$

with left and right singular vectors given by $\mathbf{u}_n^{\chi x} \in \mathbb{C}^{2N}$ and $\mathbf{v}_n^{\chi x} \in \mathbb{C}^{2M}$, respectively. The diagonal matrix $\mathbf{\Sigma}_{2k}$ has a special structure where its singular values ($\sigma_{n'}$), which are also the singular values of $\mathbf{X}$, appear by pairs along its main diagonal written as

$$\mathbf{\Sigma}_{2k} = \mathtt{diag}(\sigma_1; \sigma_1; \sigma_2; \sigma_2; ...\sigma_k; \sigma_k). \tag{2.55}$$

Such redundancy happens given how $\chi_{\mathbf{X}}$ is built to preserve quaternion properties. Once computed the complex SVD of $\chi_{\mathbf{X}}$, obtaining the singular values and vectors of $\mathbf{X}$ is, therefore, a matter of associating the correct indexes: the n-th singular elements of $\mathbf{X}$ ($\sigma_n, \mathbf{u}_n, \mathbf{v}_n$) are associated with the $n' = (2n-1)$-th singular elements of $\chi_{\mathbf{X}}$ ($\sigma_{n'}, \mathbf{u}_{n'}^{\chi\mathbf{x}}, \mathbf{v}_{n'}^{\chi\mathbf{x}}$). Notice, moreover, that its singular vectors, $\mathbf{u}_{n'}^{\chi\mathbf{x}}$ and $\mathbf{v}_{n'}^{\chi\mathbf{x}}$, have the especific structure (Zhang, 1997)

$$\mathbf{u}_{n'}^{\chi\mathbf{x}} = \begin{pmatrix} \mathbf{u}_{n'_s}^{\chi\mathbf{x}} \\ -(\mathbf{u}_{n'_p}^{\chi\mathbf{x}})^* \end{pmatrix} \quad \text{and} \quad \mathbf{v}_{n'}^{\chi\mathbf{x}} = \begin{pmatrix} \mathbf{v}_{n'_s}^{\chi\mathbf{x}} \\ -(\mathbf{v}_{n'_p}^{\chi\mathbf{x}})^* \end{pmatrix}, \tag{2.56}$$

which are, in fact, quaternion vectors in their complex notation similarly to equation 2.34, and have to be rewritten in quaternionic form following

$$\mathbf{u}_n = \mathbf{u}_{n'_s}^{\chi\mathbf{x}} + \mathbf{u}_{n'_p}^{\chi\mathbf{x}} j \quad \text{and} \quad \mathbf{v}_n = \mathbf{v}_{n'_s}^{\chi\mathbf{x}} + \mathbf{v}_{n'_p}^{\chi\mathbf{x}} j. \tag{2.57}$$

Having the $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V})$ factors of a quaternion matrix $\mathbf{X}$, it is possible to obtain a rank-r ($r < k$) approximation of a given quaternion matrix, in a least-squares sense, through the truncation of its first r singular elements in equation 2.53 (Le Bihan and Mars, 2004).

Not only linear prediction and matrix decomposition have been extended to their quaternionic counterparts. Although there is a recent increase in the application of hypercomplex numbers in signal processing being documented, quaternion-based signal processing dates back to the 1980's, with applications of the quaternionic Fourier transforms in nuclear magnetic resonance imaging by Ernst et al. (1987) and Delsuc (1988). The following section introduces the quaternion Fourier Transform and some of its particularities.

2.3.4 Fourier Transform

The retooling offered by quaternion-based signal processing introduces noteworthy differences in the signal analysis due to the particularities of the quaternion algebra. In this view, an excellent example is the quaternion Fourier transform (QFT), another natural extension in the processing and analysis of quaternion-valued signals. A formal definition of hypercomplex spectral transformations can be found in Ell (1992) albeit earlier applications had already been documented, as aforementioned. Good references on the topic are Hitzer and Sangwine (2013) and Ell et al. (2014).

The advent of the QFT allows signal processing techniques performed in the frequency to be generalized to quaternion-defined datasets (Ell and Sangwine (2007); Ortolani et al. (2017); Bahia and Sacchi (2019b)). However, the fact that quaternion multiplication is noncommutative has important implications in the QFT definition. Specifically, due to noncommutativity, the positioning of the exponential kernel in the transform definition is meaningful; therefore, the QFT is not uniquely defined and requires the definition of a *side* for the transformation. The QFT also requires the definition of an arbitrary pure unit quaternion ($\mu^2 = -1$), called the *eigenaxis* of the transformation, which represents a direction in the signal space to which the transform is being applied (Ell and Sangwine, 2007). Therefore, it defines a preferential association of the components' axes. However, this preferential association is application dependent and can be defined accordingly by the practitioner.

Definition

Two 1D right-sided QFTs of a quaternion-valued function $f : \mathbb{R} \to \mathbb{H}$ can defined in the continuous case as follows (Ell et al., 2014)

$$\mathcal{F}^{R}_{\mp\mu}\{f\}(\omega) = F^{R}(\omega) = \kappa_{-} \int_{-\infty}^{+\infty} f(t) e^{\mp\mu\omega t} dt. \tag{2.58}$$

The QFT is referred to as *forward* or *reverse* transform if the eigenaxis is set to $-\mu$ or μ, respectively. The reverse transform is not the same as the inverse QFT, which given by

$$\mathcal{F}^{-R}_{\mp\mu}\{F^{R}\}(t) = f(t) = \kappa_{+} \int_{-\infty}^{+\infty} F^{R}(\omega) e^{\pm\mu\omega t} d\omega, \tag{2.59}$$

where μ is an arbitrary pure unit quaternion ($\mu^2 = -1$), and $\kappa_{+}\kappa_{-} = (2\pi)^{-1}$. This definition is said to be the *right-sided* QFT because the exponential kernel is placed to the right of the function to be transformed. In a similar fashion, therefore, one can define the *left-sided*

QFT by plancing the exponential kernel to the left side of $f(t)$ or $F(\omega)$ as

$$\mathcal{F}^{L}_{\mp\mu}\{f\}(\omega) = F^{L}(\omega) = \kappa_{-}\int_{-\infty}^{+\infty} e^{\mp\mu\omega t} f(t)dt. \tag{2.60}$$

with inverse given by

$$\mathcal{F}^{-L}_{\mp\mu}\{F^{L}\}(t) = f(t) = \kappa_{+}\int_{-\infty}^{+\infty} e^{\pm\mu\omega t} F^{L}(\omega)d\omega. \tag{2.61}$$

In the 1D case, these already represent four possible left-right and forward-reverse transform pairs. For vector signals of higher dimensions, such as color images, the number of QFT definitions is even greater because noncommutativity also implies that $e^{ia+jb} \neq e^{ia}e^{jb}$, and two different kernels can be placed on either side of the function to be transformed. This gives rise to sandwiched, factored, and unfactored multi-axis definitions of the QFT as shown in table 2.1 for the forward 2D case (Hitzer and Sangwine, 2013).

Table 2.1: Possible definitions of Quaternion Fourier transform for 2D signals $f(x, y)$.

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

Notice how defining generalized complex operators such as $\mu^2 = -1$ increases the number of possible QFTs even further[1]. Furthermore, the eigenaxis generalizes the complex operator ($i^2 = -1$) in the classic definition of the Fourier transform, and therefore the well-known FT is understood as a particular case of the QFT where $\mu = i$.

Properties

An essential aspect of Fourier analysis is rooted in its properties and one's ability to exploit them properly. While several basic properties are easily extended to the QFT of quaternion-valued signals, others deserve closer attention due to quaternion's noncommutative algebra. A fundamental property for this thesis is the spectral counterpart of the quaternion convolution theorem, which is discussed in this subsection.

Due to noncommutativity, it is usual to expect that the sideness of the transform has to be taken into account when using the QFT for processing quaternion signals. With that

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

in mind, some basic relationships are somewhat straightforward extensions from the classic Fourier transform pairs. For example, the time domain shift $f(t-\tau)$ is written as

$$f(t-\tau) = F^R_{-\mu}(\omega)e^{-\mu\omega\tau} \tag{2.62}$$

and

$$f(t-\tau) = e^{-\mu\omega\tau}F^L_{-\mu}(\omega) \tag{2.63}$$

according to the side of the transform as right or left sided, respectively. Other properties, as the convolution theorem, require more work when extended to the quaternion domain as shown by Ell and Sangwine (2007).

The convolution of two signals corresponds to their elementwise frequency-domain multiplication. This is known to be true for either real or complex-valued signals. In the quaternion case, however, the same is not possible because noncommutativity plays its role. Still, Ell and Sangwine (2007) show how to obtain a spectral-domain *formulae* for one-sided (left or right) convolutions of two quaternion functions f and g along an eigenaxis μ as

$$\mathcal{F}^L_\mu\{f * g\} = F^L_s \circ G^L + F^L_p\nu \circ G^{L_+} \tag{2.64}$$

and

$$\mathcal{F}^R_\mu\{f * g\} = F^R \circ G^R_s + F^{R_+} \circ G^R_p\nu, \tag{2.65}$$

where "$\circ$" represents the entrywise product of two arrays, "$*$" represents the convolution operation, the superscripts ^{L} (or ^{R}) and $^{L_+}$ (or $^{R_+}$) indicate left (or right) forward and reverse QFTs, respectively, and the subscripts $_s$ and $_p$ indicate that the signal was decomposed, respectively, into simplex and perplex parts. Therefore, one can also exploit QFTs to perform convolutions of quaternion-valued functions via the proper (regarding sideness and eigenaxis definition) version of the quaternion convolution theorem, i.e., using the correct inverse QFT of equations 2.64 or 2.65. For that, it is important to understand the implementation of the QFT, which is discussed next.

Implementation

The symplectic decomposition in equation 2.28 defines an important relationship between complex and quaternions numbers which answers what are the criteria for quaternion functions to be Fourier-transformable. For that, assume an eigenaxis μ for the transformation, and take the symplectic decomposition of the function to be transformed $f(t)$ along μ as

$$f(t) = f_s(t) + f_p(t)\nu, \tag{2.66}$$

where $f_s(t)$, $f_p(t) \in \mathbb{C}_\mu$, and ν is a pure unit quaternion perpendicular to μ. Following with the left-sided (similarly for the right-sided) QFT of $f(t)$ one obtains

$$F^L(\omega) = \kappa_- \int_{-\infty}^{+\infty} e^{-\mu\omega t} f_s(t) dt + \kappa_- \int_{-\infty}^{+\infty} e^{-\mu\omega t} f_p(t) dt\nu$$

$$F^L(\omega) = F_s(\omega) + F_p(\omega)\nu \tag{2.67}$$

where $F_s(\omega) = \kappa_- \int_{-\infty}^{+\infty} e^{-\mu\omega t} f_s(t) dt$ and $F_p(\omega) = \kappa_- \int_{-\infty}^{+\infty} e^{-\mu\omega t} f_p(t) dt$ showing how the QFT can be decomposed as the transform of two complex subfields. Therefore, given the equivalence between $\mathbb{C}_\mu$ and standard complex numbers, existence and invertibility of the QFT are inherited from the complex case (Ell et al., 2014). In addition, numerical implementation is also greatly facilitated by the symplectic decomposition, which allows the use of already existing Fourier transform libraries, such as the fast Fourier transform (FFT).

Regarding computational aspects of QFTs, it is first necessary to define the discrete QFT (DQFT), as it is done for the 1D left-sided case[2], assuming a length-N signal f

$$F[\omega] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-\mu 2\pi \frac{n\omega}{N}} f[n], \tag{2.68}$$

with inverse

$$f[n] = \frac{1}{\sqrt{N}} \sum_{\omega=0}^{N-1} e^{\mu 2\pi \frac{n\omega}{N}} F[\omega]. \tag{2.69}$$

While it is possible to write a direct code for the computation of the DQFT by evaluating the hypercomplex exponentials in equations 2.68 and 2.69, or even an FFT code for it, Ell and Sangwine (2007) propose to take advantage of already existing FFT codes by using, again, the symplectic decomposition of quaternions and equivalences with the complex case.

Equation 2.67 shows that it is possible to decompose the QFT along an eigenaxis μ as a pair of complex-valued Fourier transforms. Since $F_s(\omega)$ and $F_p(\omega) \in \mathbb{C}_\mu$, these transforms can be obtained via complex-valued FTs by replacing the generalized operator μ by the complex operator i in the function f to be transformed. Numerically, that is done by projecting the quaternion-valued f onto a new 4D basis $\{1, \mu, \nu, \mu\nu\}$ following equation 2.27, and reverting the change of basis after the transform has been computed. The change of basis allows interpreting μ as i in the complex-valued Fourier transforms employed in the QFT. Furthermore, according to Ell and Sangwine (2007), the choice of μ and ν is arbitrary, but these cannot be parallel to each other. If a user-defined ν is initially not perpendicular to μ, an updated value for ν can be found by taking the vector cross product between μ and

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

$\mu\nu$. Having defined the 4D basis in $\mathbb{H}$, the 1D DQFT can be computed as follows:

- Apply a change of basis from f to f' following equation 2.27.
- Decompose the function into its simplex and perplex parts following 2.28.
- Perform two 1D complex FFTs of f_s and f_p to obtain $F_s[\omega]$ and $F_p[\omega]$.
- Construct the full quaternion in frequency domain by $F'[\omega] = F_s[\omega] + F_p[\omega]\nu$.
- Reverse the change of basis from equation 2.27 to obtain $F[\omega]$ from $F'[\omega]$.

Therefore, from the generalization of FFTs to N-dimensional (ND) signals, the steps above can be generalized to define an ND QFT (Stanton and Sacchi, 2013).

Coefficient Layout

It is now possible to describe the Fourier analysis of vector signals defined with quaternion numbers. Figure 2.1(a)-(c) shows three components of a source line extracted from a common-receiver gather (CRG) synthesized in a 3D-3C vertical seismic profile (VSP) experiment. An array of $N_r = 31$ vector sensors are placed downhole to record three orthogonal displacement components, here denoted by $(\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z) \in \mathbb{R}^{N_t \times N_{s_x} \times N_{s_y}}$, at N_t discrete time samples and $N_s = N_{s_x} N_{s_y}$ discrete shot positions. In particular, there are $N_{s_x} = 205$ source lines, with $N_{s_y} = 205$ source positions each, and the source spacing equals the receiver spacing as $\Delta s_x = \Delta s_y = \Delta r = 16.67$ meters. Figure 2.1(d)-(f) shows the corresponding frequency-wavenumber (FK) spectrum obtained for each respective component in (a)-(c) on a classical basis where only half of the spectra is needed to fully characterize the signal.

From the quaternionic signal model in equation 2.36, the three components are encoded as the pure quaternion matrix $\mathbf{Q} \in V(\mathbb{H}^{N_t \times N_{s_x} \times N_{s_y}})$ such as

$$\mathbf{Q}[t, s_x, s_y] = \mathbf{U}_x[t, s_x, s_y]i + \mathbf{U}_y[t, s_x, s_y]j + \mathbf{U}_z[t, s_x, s_y]k, \tag{2.70}$$

with time $t = [1, N_t]$ and source indexes $s_x = [1, N_{s_x}]$ and $s_y = \left[1, N_{s_y}\right]$. In this case, $\mathbf{Q}$ represents the multicomponent seismic data, where $\mathbf{U}_x$, $\mathbf{U}_y$, and $\mathbf{U}_z$ are the two horizontal and vertical components, respectively, naturally accounting for the mutual information between data components. Application of QFT along the time domain yields another quaternion signal, usually a full quaternion, represented by the QFTed simplex and perplex parts of $\mathbf{Q}$

$$\tilde{\mathbf{Q}}[\omega, s_x, s_y] = \tilde{\mathbf{Q}}_s[\omega, s_x, s_y] + \tilde{\mathbf{Q}}_p[\omega, s_x, s_y]j, \tag{2.71}$$

with $\tilde{\mathbf{Q}} \in \mathbb{H}^{N_t \times N_{s_x} \times N_{s_y}}$ obtained through the QFT with respect to time. Inhereted from the complex case, the frequency domain coefficients in the hypercomplex spectrum have the same layout as the classical spectrum, which is divided into positive and negative frequencies, also including the DC and Nyquist terms, as usually defined (Ell et al., 2014). Therefore, a symmetry of the coefficients indexes exists, and the same frequency component is represented by the indexes N and $N_t - N$. As in the complex case, however, there are no conjugate symmetries between the frequency components of the QFT of a quaternion signal.

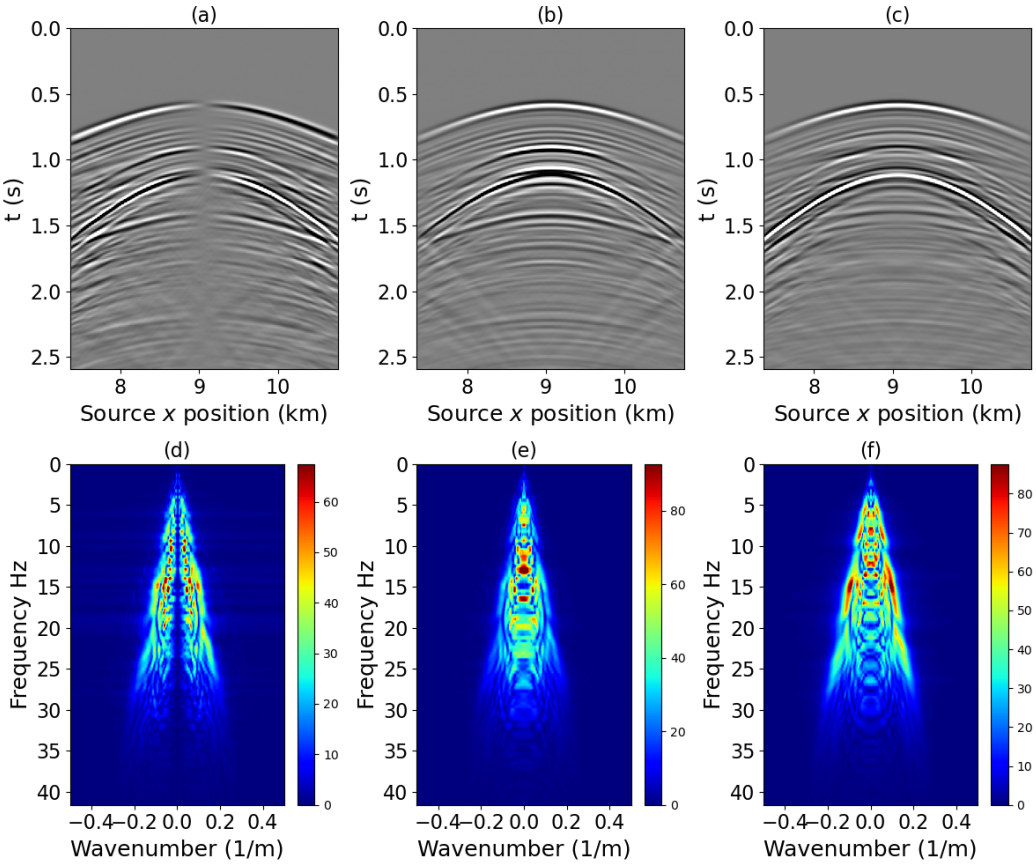

Figure 2.1: Synthetic (a) $\mathbf{U}_x$, (b) $\mathbf{U}_y$, and (c) $\mathbf{U}_z$, and their respective FK spectrum in (d)-(f). The data is obtained via elastic finite-difference modeling, and the FK spectra through classic (componentwise) Fourier analysis.

A signal such as $\tilde{\mathbf{Q}}$, which has been QFTed, is written in terms of cosinusoidal basis functions (the hypercomplex exponentials) and hypercomplex amplitudes. It is possible, therefore, to write the hypercomplex representation of the signal in terms of its amplitude and phase, analogously to classical processing, and the additional concept of orientation (the eigenaxis), as it follows from equation 2.18. The amplitude and FK spectra of the above seismic data are shown in figure 2.2, where both of these are obtained using the polar representation of quaternions, after applying the left-sided QFT with eigenaxis $\mu = \frac{1}{\sqrt{3}}(1, 1, 1)$ to the data.

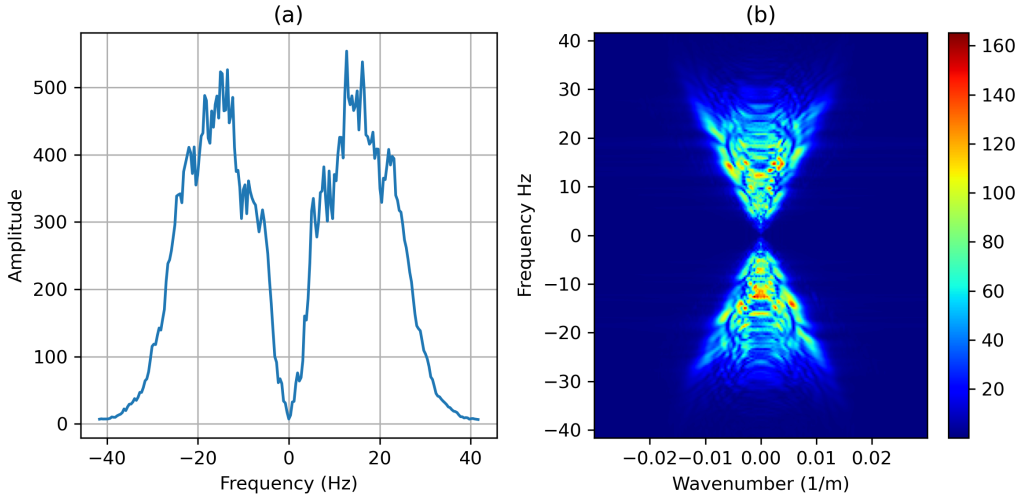

Figure 2.2: (a) Amplitude spectrum and (b) FK spectrum of the quaternion-based seismic signal.

On the one hand, the hypercomplex spectrum offers the advantage of representing the amplitude of all of its elements in a single term given by its modulus (equation 2.11). As it might be expected, this is not the same as summing the spectra shown in figure 2.1(d)-(f). In addition, the frequency content shown in figure 2.2(a) is similar for both positive and negative frequencies of the spectrum; notice how these are not equal since there are no conjugate symmetries for quaternion signals in the frequency domain. As an important implication, operations carried out in monochromatic frequency slices of seismic data, as in classic frequency-space ($f - x$) filtering techniques, have to be separately applied to each side of the spectrum when extended to quaternion-defined seismic data. Such is the case for the techniques presented in the following chapters: the quaternionic extension of the Singular Spectrum Analysis (SSA) and $f - x$ deconvolution (FXDECON) filters.

CHAPTER 3

Quaternionic Rank-Reduction Methods for Vector-field Seismic Data[1]

3.1 Introduction

The extension of well-known techniques to the quaternion domain, as discussed in Chapter 2, allows for the generalization of different signal processing methods to the case of quaternion signals. For instance, the truncated QSVD has been used in rank-reduction applications to perform random Gaussian noise attenuation (Le Bihan and Mars, 2004), multichannel analysis of electroencephalograms (Enshaeifar et al., 2016), and wavefield separation (Sajeva and Menanno, 2017). These applications illustrate the potential of quaternions in offering a physically meaningful alternative treatment to the case of vector-valued data.

Random noise attenuation of scalar seismic data, together with seismogram (trace) reconstruction, can be accomplished in different ways as, for instance, $f-x$ prediction (Canales, 1984) and projection (Soubaras, 1994) filters, and Fourier methods (Liu and Sacchi, 2004). Rank-reduction is yet another option in processing scalar seismic data, such as the Singular Spectrum Analysis (SSA) filter. SSA belongs to the group of $f-x$ methods in seismic data processing, and its applications mainly entail random noise attenuation and missing trace recovery (Oropeza and Sacchi, 2011; Gao et al., 2013).

The SSA theory was extended to the quaternion domain named QSSA (Enshaeifar et al., 2016). Based on the augmented second-order statistics of quaternion signals, it is possible to formulate the augmented QSSA (AQSSA), aiming to exploit the additional information

[1] A version of this chapter is published in Bahia, B., and Sacchi, M. D. (2019). Quaternionic rank-reduction methods for vector-field seismic data processing. Digital Signal Processing, 87, 178-189.

that lives in the correlation between signal components. This chapter illustrates applications of the QSSA and AQSSA in simultaneous denoising and reconstruction of multicomponent seismic data following a vector-consistent approach.

3.2 Theory

3.2.1 Singular Spectrum Analysis

The Singular Spectrum Analysis (SSA) is a data-driven subspace decomposition algorithm where the original signal is represented as the sum of components classified as belonging to signal or noise subspaces. This section follows Oropeza and Sacchi (2011) in the description of the SSA applied to seismic data processing. The SSA operator can be represented by

$$\mathcal{P}\{\cdot\} = \mathcal{A}\{\mathcal{R}\{\mathcal{H}\{\cdot\}, k\}\} \tag{3.1}$$

where the operations of Hankelization ($\mathcal{H}\{\cdot\}$), rank-reduction ($\mathcal{R}\{\cdot, k\}$), and antidiagonal averaging ($\mathcal{A}\{\cdot\}$) are explained in what follows.

Consider a regularly sampled 2D scalar dataset, $\mathbf{D}[t, n]$, with $t = [1, N_t]$ representing time samples, and $n = [1, N_1]$ the seismograms in the spatial dimension. One monochromatic frequency slice of the Fourier transformed data, $\tilde{\mathbf{D}}[\omega, n] = \mathcal{F}\{\mathbf{D}[t, n]\}$, can be represented by the following $N_1 \times 1$ vector $\mathbf{s}$

$$\mathbf{s} = \begin{pmatrix} s_1 & s_2 & \dots & s_{N_1} \end{pmatrix}^T, \tag{3.2}$$

where "^{T}" represents the transpose operator. Note that the dependency on ω has been omitted, and the analysis is performed for every frequency slice within a range $[\omega_{min}, \omega_{max}]$.

In the case of 2D datasets, SSA reduces to a 1D problem since each frequency slice is analyzed separately. Then, the Hankelization operation consists in the construction of a single level-1 $L_1 \times K_1$ Hankel matrix, $\mathbf{H}^{(1)} \in \mathbb{C}^{L_1 \times K_1}$. The columns of $\mathbf{H}^{(1)}$ are one-sample lagged versions of a length-L_1 window of the frequency slice $\mathbf{s}$. In the SSA literature, this matrix is named the trajectory matrix, and it is given by

$$\mathbf{H}^{(1)} = \mathcal{H}\{\mathbf{s}\} = \begin{pmatrix} s_1 & s_2 & \dots & s_{K_1} \\ s_2 & s_3 & \dots & s_{K_1+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{L_1} & s_{L_1+1} & \dots & s_{N_1} \end{pmatrix} \tag{3.3}$$

where $K_1 = N_1 - L_1 + 1$, with L_1 such that the matrix $\mathbf{H}^{(1)}$ is close to square.

In a similar fashion, for 3D datasets, as $\mathbf{D}[t, n, m]$, with $m = [1, N_2]$ representing the seismograms in the additional spatial dimension, the monochromatic frequency slice of the data can be represented by the $N_1 \times N_2$ matrix

$$\mathbf{S} = \begin{pmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,N_2} \\ s_{2,1} & s_{2,2} & \dots & s_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N_1,1} & s_{N_1,2} & \dots & s_{N_1,N_2} \end{pmatrix}. \tag{3.4}$$

The problem is now 2D, and the Hankelization operation consists in the construction of a level-1 trajectory matrix (equation 3.3) for each row $j = [1, N_1]$ of $\mathbf{S}$. In this case, one constructs a sequence of N_1 level-1 Hankel matrices, $\mathbf{H}_j^{(1)}$, which are then organized into a level-2 Hankel matrix $\mathbf{H}^{(2)}$, where $K_2 = N_2 - L_2 + 1$, and following the same idea as before

$$\mathbf{H}^{(2)} = \mathcal{H}\{\mathbf{S}\} = \begin{pmatrix} \mathbf{H}_1^{(1)} & \mathbf{H}_2^{(1)} & \dots & \mathbf{H}_{K_2}^{(1)} \\ \mathbf{H}_2^{(1)} & \mathbf{H}_3^{(1)} & \dots & \mathbf{H}_{K_2+1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{L_2}^{(1)} & \mathbf{H}_{L_2+1}^{(1)} & \dots & \mathbf{H}_{N_1}^{(1)} \end{pmatrix}. \tag{3.5}$$

Following this same methodology, one would be able to construct higher-level Hankel matrices ($\mathbf{H}^{(n)}$) for filtering seismic data that depend on more spatial dimensions (Gao et al., 2013). Note that $\mathbf{H}^{(n)}$ would be a block Hankel matrix of size $L_1 L_2 ... L_n \times K_1 K_2 ... K_n$, and its construction can be generally represented by the Hankelization operator as

$$\mathbf{H}^{(n)} = \mathcal{H}\{\mathbf{S}\}. \tag{3.6}$$

Furthermore, the window lengths (L_i with $i = [1, n]$) are chosen such that $\mathbf{H}^{(n)}$ is the closest to a square matrix, so $L_i = \lfloor \frac{N_i}{2} \rfloor + 1$, where $\lfloor \rfloor$ denotes the integer part of its argument. Also, notice that $\mathbf{H}^{(n)}$ can be used to compute the sample covariance matrix, $\tilde{\mathbf{C}}^{(n)}$, as

$$\tilde{\mathbf{C}}^{(n)} = \frac{1}{L} \mathbf{H}^{(n)} \mathbf{H}^{(n)H}, \tag{3.7}$$

where the superscript ^{H} denotes conjugate transposition of a matrix, and $L = L_1 L_2 ... L_n$. This matrix captures an averaged correlation of the elements in the level-n trajectory matrix, and it can be used in the rank-reduction step in SSA (discussed next), yielding the same orthonormal basis as $\mathbf{H}^{(n)}$, but with squared eigenvalues (Broomhead and King, 1986).

In the seismology scenario, a data window that contains k linear events in the time domain can be modeled by the superposition of k complex exponentials in $f - x$ (Canales, 1984; Oropeza and Sacchi, 2011). Using recursive models, one can show that the rank of the ideal

Hankel matrix (noise-free and complete data) equals the number of complex exponentials that form the data (Yang and Hua, 1996). Ideally, the trajectory matrix's rank should equal the number of linear events in the time domain of a seismic section. Random additive noise and missing observations increase the rank of the ideal Hankel matrix (Oropeza and Sacchi, 2011; Gao et al., 2013). Hence, rank-reduction techniques have been successfully employed in random noise suppression and missing observation recovery. SSA then follows with the low-rank approximation of the trajectory matrix, usually done via the truncated singular value decomposition of $\mathbf{H}^{(n)}$, represented by

$$\mathbf{H}_k^{(n)} = \mathcal{R}\{\mathbf{H}^{(n)}, k\} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^H, \tag{3.8}$$

where k is the number of linear events in the time domain, $\mathbf{\Sigma}_k$, $\mathbf{U}_k$ and $\mathbf{V}_k$ are the first k singular values and vectors of the trajectory matrix. According to the Eckart-Young theorem (Eckart and Young, 1936), this approximates $\mathbf{H}_k^{(n)}$ to $\mathbf{H}^{(\mathbf{n})}$ in a least-squares sense. Finally, the reconstructed data can be obtained from $\mathbf{H}_k^{(n)}$ by averaging the elements along its antidiagonals, which is represented by the operator $\mathcal{A}$

$$\hat{\mathbf{S}} = \mathcal{A}\{\mathbf{H}_k^{(n)}\}. \tag{3.9}$$

2D Denoise Example

In multicomponent seismology, three components might be simultaneously acquired (figure 3.1), referring to the two horizontal and the vertical components of the vector wavefield, here represented by the displacements $\{\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z\}$.

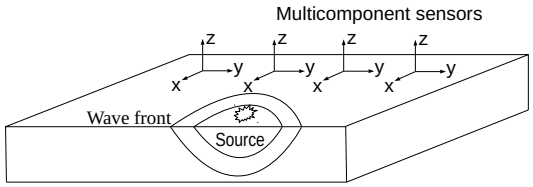

Figure 3.1: Illustrative representation of multicomponent seismic data acquisition.

The data shown in figure 3.2 exemplify these components. It corresponds to a spatial slice view ($\mathbf{U}_{x,y,z}[1:N_t, 10, 1:N_2]$) of a synthetic 3C-3D shot gather generated via finite-difference modeling assuming an isotropic elastic medium in a north-east coordinate frame.

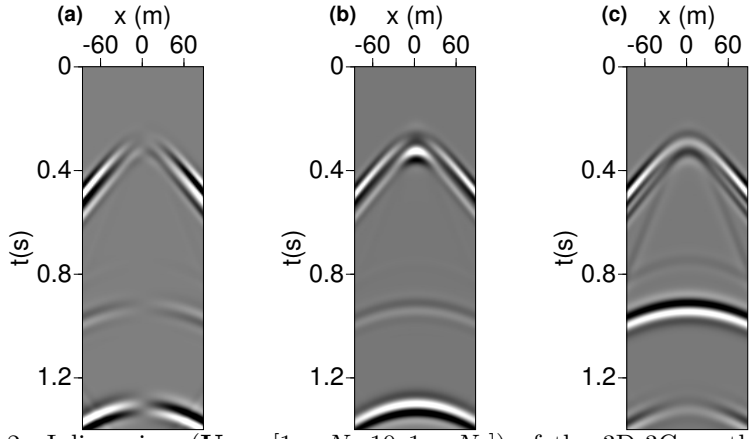

Figure 3.2: Inline view ($\mathbf{U}_{x,y,z}[1:N_t, 10, 1:N_2]$) of the 3D-3C synthetic data components (a) $\mathbf{U}_x$, (b) $\mathbf{U}_y$, and (c) $\mathbf{U}_z$.

When treating such data, one might adopt the usual approach of processing each component separately. For instance, one could apply SSA to each component to attenuate random noise via rank-reduction. To approximate the assumption of linear events in the time domain, the data are usually divided into overlapping windows (figure 3.3), and SSA is applied to each window separately. Of course, the components are likely to be corrupted by additive Gaussian noise having different levels of signal-to-noise ratio (SNR[2]). In this example, the SNRs are set to $[SNR_x, SNR_y, SNR_z] = [0.8, 1.0, 1.2]$.

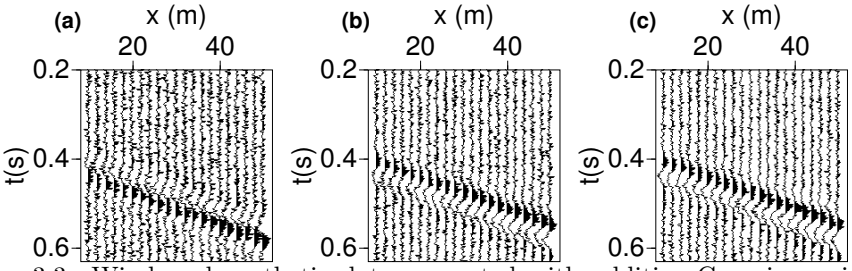

Figure 3.3: Windowed synthetic data corrupted with additive Gaussian noise in components (a) $\mathbf{U}_x$, (b) $\mathbf{U}_y$, and (c) $\mathbf{U}_z$.

[2]The SNR definition used here $SNR = \frac{A^2_{signal}}{A^2_{noise}}$., where A^2_{signal} and A^2_{noise} refers to the mean square amplitude of the signal and noise, respectively.

As only one quasi-linear event is observed in the window, the ideal rank of the trajectory matrix should be one for each component. Since random noise disrupts its low-rank character (figure 3.4), denoising can be posed as a rank-reduction problem. In general, however, the rank selection is more complicated due to, for instance, the presence of non-linear events and amplitude variations with source-receiver offset. One possible way to work around such issues is to increase k to two or three times the number of expected events in the window (Oropeza and Sacchi, 2011). Nonetheless, it is possible to denoise all components separately using rank-reduction with $k = 1$ (figure 3.5).

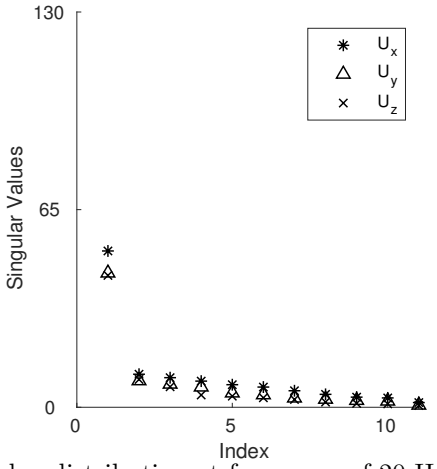

Figure 3.4: Singular value distribution at frequency of 20 Hz for each component.

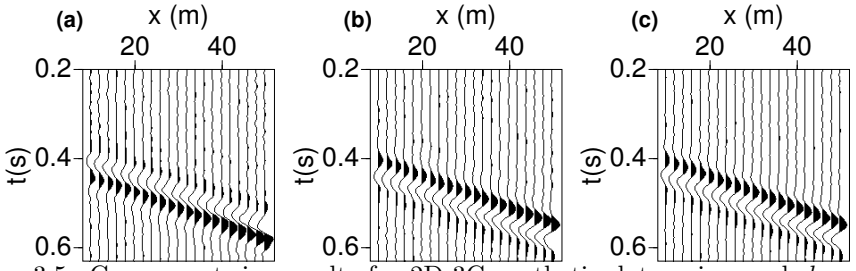

Figure 3.5: Componentwise results for 2D-3C synthetic data using rank $k = 1$ in the SVD for each component (a) $\mathbf{U}_x$, (b) $\mathbf{U}_y$, and (c) $\mathbf{U}_z$.

The componentwise approach takes no advantage of the complete information carried by the vectorial data. The following section discusses how to apply the quaternionic version of the SSA to the case of vector-valued seismic data, thus overcoming this issue.

3.2.2 Quaternion Singular Spectrum Analysis

The quaternionic extension of the Singular Spectrum Analysis (QSSA) was first given by Enshaeifar et al. (2016) who exploited the aforementioned advances in quaternion statistics (Section 2.3.1) to define two quaternionic versions of the SSA. The first, called QSSA, is the trivial extension of the SSA to the quaternion domain, with the proper replacement of real or complex techiniques by its quaternionic counterparts. In other words, the same operator in equation 3.1 can be used to describe the operations of Hankelization, rank-reduction and anti-diagional averaging, as explained above. Of course, the rank-reduction operation, so far given by the SVD of the trajectory matrix ($\mathbf{H}^{(n)}$ in equation 3.6), is now done through the QSVD (Section 2.3.3). The second quaternionic SSA, named augmented QSSA (AQSSA), aims at exploiting the complete second-order information within the quaternion signal by augmenting the trajectory matrix with its involutions as follows

$$\mathbf{H}^{(n)a} = \begin{pmatrix} \mathbf{H}^{(n)} \\ \mathbf{H}^{(n)i} \\ \mathbf{H}^{(n)j} \\ \mathbf{H}^{(n)k} \end{pmatrix}. \tag{3.10}$$

Note that the sample covariance matrix obtained from $\mathbf{H}^{(n)a}$ (equation 3.7) captures the complete second-order information in the quaternion signal similarly to equation 2.41. Such is the main difference between QSSA and AQSSA, which aims at explicitly exploiting the correlation between data components in a vector-consistent approach, as is illustrated next.

2D Denoise Example

When applying QSSA or AQSSA to quaternion-defined seismic signals, the same description given in section 3.2.1 follows. First, define the pure quaternion array

$$\mathbf{D}[t, n, m] = \mathbf{U}_x[t, n, m]i + \mathbf{U}_y[t, n, m]j + \mathbf{U}_z[t, n, m]k, \tag{3.11}$$

in the time-space $(t - x)$ domain, which has to be transformed to $f - x$ via the QFT. In this chapter, the QFT is applied from the left along $\mu = \frac{1}{\sqrt{3}}(i + j + k)$.

As there are no conjugate symmetries in the frequency domain for quaternion signals, the analysis is performed for every frequency slice within a range $[\omega_{min}, \omega_{max}]$ on both sides of the spectrum (*i.e.*, positive and negative frequencies). Along with the augmentation step in equation 3.10, that is a key difference between SSA and QSSA/AQSSA, as no significant implication in the sideness and eigenaxis definitions of the QFT was yet found. Perhaps more importantly, figure 3.6 shows that representing vector data as a quaternion signal does

not affect the rank of the trajectory matrix. Its low-rank character can be demonstrated through QAR models (Section 2.3.2), assuming processes with zero mean and no innovation.

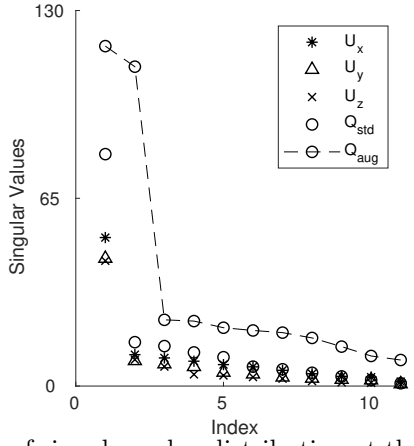

Figure 3.6: Comparison of singular value distribution at the frequency of 20 Hz for each component and the quaternion signals. The QSSA and AQSSA singular values are represented by Q_{std} and Q_{aug}, respectively.

The augmentation step in AQSSA increases the rank of the trajectory matrix from 1 to 2 in this case, possibly illustrating the additional information exploited by this approach. Also, the singular values found via QSVD are of higher amplitude than those found via SVD for each component (figure 3.6). Both approaches can simultaneously process multicomponent seismic data, as the results show in figure 3.7.

3.2.3 Efficient Formulations

These simple examples illustrate how the SSA is a powerful signal processing technique. However, one key point highlighted in several papers (e.g., Gao et al. (2013); Chiron et al. (2014); Cheng and Sacchi (2017)) refers to the high computational costs associated with the SVD of the trajectory matrix, as well as the memory requirements for its storage. Although forming the trajectory matrix can be taken as computationally efficient, performing the SVD of particularly big matrices is relatively slow, with a computational complexity of $O(L^2K)$ in the case of a $L \times K$ matrix. Therefore, despite its good performance, SSA might be an unfeasible algorithm for large datasets because of high computational costs.

One option to improve rank-reduction efficacy in the case of scalar data entails replacing the expensive SVD with other rank-reduction tools, as discussed in Appendix A. For instance, the structure of the trajectory matrix can be exploited to improve the efficacy of the Lanczos

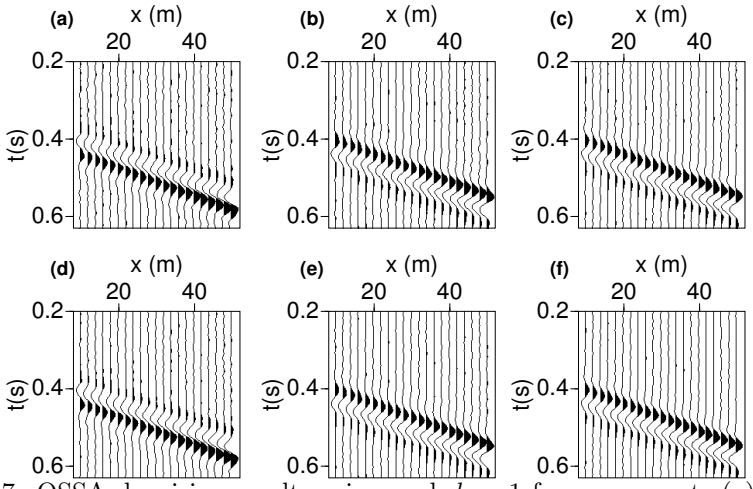

Figure 3.7: QSSA denoising results using rank $k = 1$ for components (a) $\mathbf{U}_x$, (b) $\mathbf{U}_y$, and (c) $\mathbf{U}_z$, and (d)-(f) AQSSA using $k = 2$. Results were obtained via QSVD.

bidiagonalization using FFTs to perform matrix-vector multiplications. Other publications addressing this topic point to randomization as a powerful tool to deal with such issues, as it usually outperforms its deterministic versions in accuracy and speed (Liberty et al., 2007; Halko et al., 2011). Some interesting applications of the randomized QR decomposition include 5D scalar seismic data reconstruction via parallel matrix factorization (Carozzi and Sacchi, 2017) and the efficient formulation of the SSA (Cheng and Sacchi, 2017).

Although the quaternion extensions of classic signal processing techniques bring the possibility of taking better advantage of the information in the data, they also carry associated drawbacks with them. Quaternion-based signal processing is known to be more computationally expensive than its real or complex counterparts since quaternion multiplication (equation 2.8) requires more floating-point operations than classic (real or complex) multiplications. That imposes a challenge in developing multidimensional vector-valued seismic data filtering via rank-reduction because computationally expensive matrix decomposition techniques, such as the SVD, might suffer more in their hypercomplex versions. Fortunately, published methods that improve rank-reduction efficacy in the case of scalar data can be readily extended to the quaternion domain. Specifically, the structure of the trajectory matrix can also be exploited to improve the efficacy of the quaternionic version of the Lanczos bidiagonalization, and randomized QR algorithms, where both benefit from the quaternion convolution theorem and QFTs discussed in section 2.3.4.

Figure 3.8 compares the processing time in QSSA based on QSVD, Lanczos bidiagonalization, and randomized quaternion QR (rQQR) decomposition for a single frequency slice and increasing array size. The rank used in the QSVD was $k = 1$, and the number of Lanczos

iterations and random projections in the rQQR was $k = 4$. Notice how the Lanczos bidiagonalization and rQQR provide substantial gains in processing times, close to the order of 10^3 seconds. One of the advantages worth noticing when avoiding the SVD in the quaternion domain for this specific application is that the particular transformations to real or complex matrices are avoided. Therefore, the algorithms are implemented directly in the quaternion domain, which reduces their computational complexity.

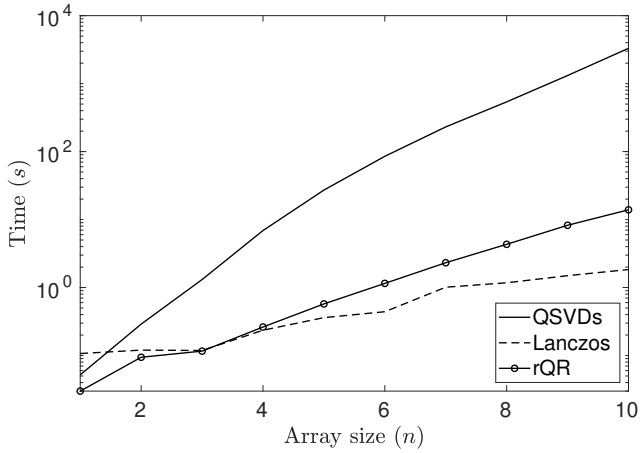

Figure 3.8: Comparison of the processing time for quaternionic rank-reduction algorithms using QSVDs (line), Lanczos bidiagonalization (dashed line), and rQQR (circles) in the QSSA. The Lanczos bidiagonalization and the rQQR were implemented using efficient matrix-vector multiplication via QFTs. The array size grows as $10n \times 10n$.

It is important to stress that, unlike the case of scalar datasets (Cheng and Sacchi, 2017), the rQQR approach could not outperform the processing times given by the Lanczos bidiagonalization because the convolution of two quaternion sequences requires two QFTs and one inverse QFT, as shown in equations 2.64 and 2.65. Since the implementation of the QFTs is based on two complex FFTs, the processing times for the random projections in the rQQR grow more rapidly than the matrix-vector multiplications in the Lanczos iterations for increasing array size. Nevertheless, the gains obtained with any of the efficient formulations of the QSSA make it possible to use these techniques iteratively for simultaneous denoising and reconstruction of multidimensional vector-valued seismic data, as discussed next.

3.3 Examples

This section tests the methods to reconstruct synthetic vector-valued seismic data. The prediction gain R_j is used to quantify (in dB) and evaluate the performance of the algorithms,

which is given by

$$R_j = 10 \log_{10} \left(\frac{\|\mathbf{U}_j^o\|_2^2}{\|\mathbf{U}_j - \mathbf{U}_j^o\|_2^2} \right), \tag{3.12}$$

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

3.3.1 Simultaneous Denoising and Reconstruction

In the case of missing seismograms, the same approach for data recovery can be used for either scalar or vector-valued data. For each frequency slice in a given range, the following POCS-like iterative reinsertion algorithm (Bro, 1997; Oropeza and Sacchi, 2011; Gao et al., 2013) performs data recovery for both scalar or quaternion-valued data

$$\mathbf{S}_i = \alpha \mathbf{S}_{obs} + (\mathbf{1} - \alpha \mathbf{T}) \circ \mathcal{P}\{\mathbf{S}_{i-1}\}, \tag{3.13}$$

where $\mathbf{S}_{obs}$ represents the observed frequency slice with missing entries, $\mathbf{S}_i$ represents the frequency slice at iteration number i, $\mathcal{P}\{\cdot\}$ correspond to the filter operator (SSA, QSSA or AQSSA) as given by equation 3.1, $\mathbf{T}$ is a sampling operator that identifies the presence of a trace in the spatial position (n, m) of the data

$$\mathbf{T}[n, m] = \begin{cases} 1, \text{if trace exists} \\ 0, \text{if trace is missing} \end{cases} \tag{3.14}$$

and $\mathbf{1}$ is a matrix of ones with the same dimensions as the input frequency slice. This way, $(\mathbf{1} - \alpha \mathbf{T})$ represents an imputation operator where $\alpha \in (0, 1]$ controls the reinsertion of reconstructed data. If $\alpha < 1$, denoising is simultaneously performed with data recovery via a weighted average of the original observations and the SSA results.

The Rank Reduction Tool

Since the SVD provides a least-squares solution in the rank-reduction problem, its reconstruction gains are expected to outperform the quaternionic Lanczos and randomized QR results. For scalar data, the QR decomposition is arguably more stable while also offering relaxation in estimating the desired rank in terms of random projections (see Appendix A). Figure 3.9 shows the reconstruction gains of the QSSA applied to a 3D-3C synthetic dataset with three constant dips under different realizations of noise and trace decimation (10% to 90%). The 'rank' (random projections or Lanczos iterations) varied from 1 to 50, and the imputation algorithm (equation 3.13) ran for 10 iterations with $\alpha = 0.5$.

Compared to the QSVD and Lanczos bidiagonalization, the rQQR also offers a relaxation in

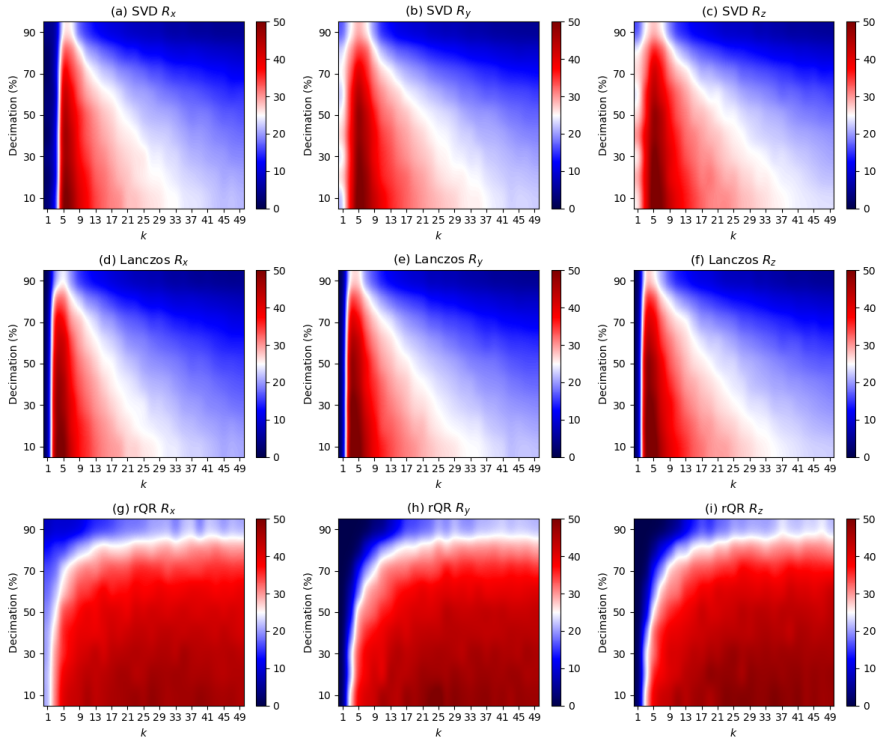

Figure 3.9: QSSA reconstruction gains for each data component x, y, z with varying levels of noise and data decimation as a function of $(\mathbf{a}) - (\mathbf{c})$ the rank in QSVD, $(\mathbf{d}) - (\mathbf{f})$ Lanczos iterations, and $(\mathbf{g}) - (\mathbf{i})$ random projections in the rQQR.

terms of rank selection for vector data, as figure 3.10 shows for an example with $p = 60\%$ of missing traces. For the remaining examples in this chapter, however, the Lanczos algorithm was favored since it is the most efficient and best performing algorithm among these options.

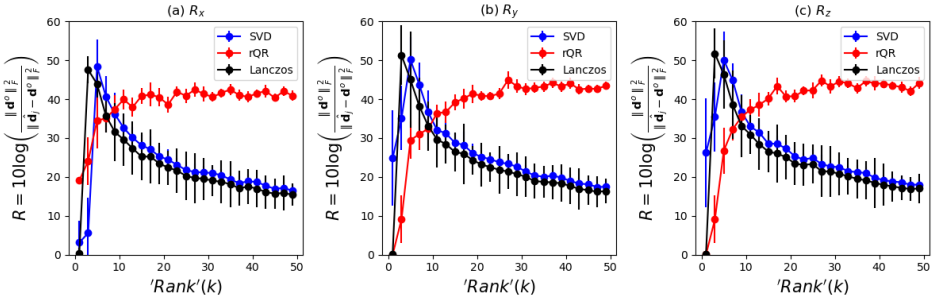

Figure 3.10: Reconstruction gains for varying levels of data decimation as a function of the rank in QSVD and random projections in rQQR for 60% decimated data.

The same synthetic 3D-3C data above was used for comparing the SSA and its quaternionic counterparts, QSSA and AQSSA, in seismic data reconstruction. The levels of data decimation also varied from 10% to 90%, and $\alpha = 0.5$ was used in as reinsertion parameter. The number of Lanczos iterations was set to $k = 8$ for the SSA and QSSA and $k = 15$ for the AQSSA, and this setup was reused over 20 realizations of the experiment. While the AQSSA could marginally outperform the QSSA, both methods improved over the componentwise application of the SSA as shown in Figure 3.11.

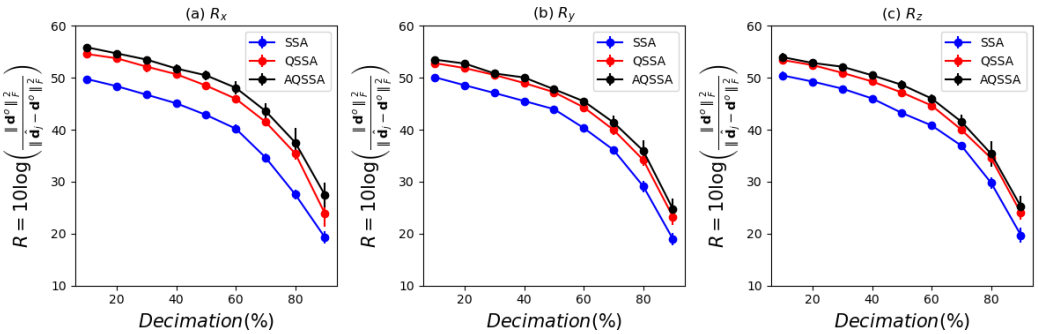

Figure 3.11: Reconstruction gains for varying levels of data decimation using $k = 8$ Lanczos iterations for the SSA and QSSA, and $k = 15$ for the AQSSA.

For a fixed level of trace decimation $p = 60\%$, the number of Lanczos iterations was allowed to vary from 1 to 22. Although outperforming the componentwise SSA, the QSSA and AQSSA have similar peak performance as shown in Figure 3.12. The figure also shows the increased subspace of the AQSSA since it requires additional iterations for better results. However, similarly to the SSA and QSSA, quality decreases for a larger subspace since noise is progressively inserted in the results.

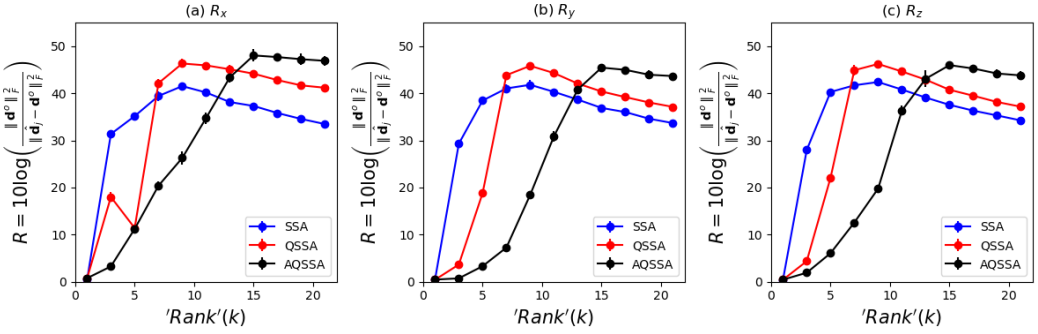

Figure 3.12: Reconstruction gains for varying levels of data decimation using $k = 8$ Lanczos iterations for the SSA and QSSA, and $k = 10$ for the AQSSA.

5D-3C Linear Events

This section illustrates one 5D-3C example with planar events, extending the ones used in figures 3.9 to 3.12. As explained in Chapter 1, a three-dimensional acquisition geometry generates data volumes that depend on time and four spatial dimensions for each component. In more detail, figure 3.13 shows traditional coordinate systems for sorting seismic data that lead to 5D representations, such as midpoint and offset domains, which are useful for applications such as denoising and reconstruction (Carozzi and Sacchi, 2019).

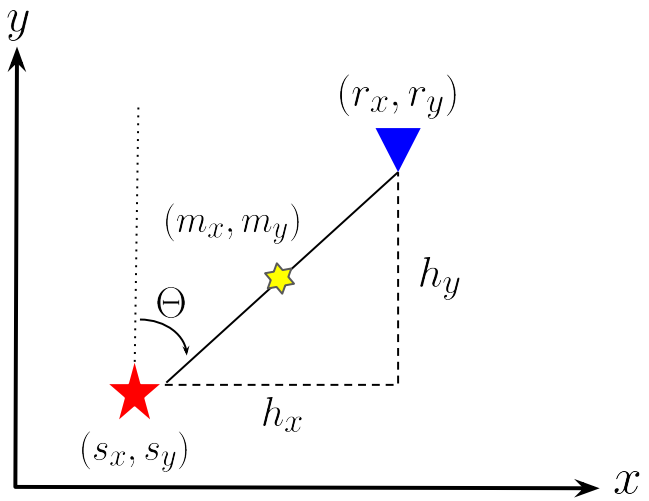

Figure 3.13: Map view of a seismic acquisition coordinate system defined by sources (s_x, s_y) and receivers (r_x, r_y). Additional coordinates, such as azimuth (Θ), midpoints (m_x, m_y), and offsets (h_x, h_y) are also shown (after Carozzi (2021)).

The midpoint and offsets coordinates are given, respectively, as

$$(m_x, m_y) = \left(\frac{s_x + r_x}{2}, \frac{s_y + r_y}{2} \right)$$

and

$$(h_x, h_y) = \left(\frac{s_x - r_x}{2}, \frac{s_y - r_y}{2} \right).$$

The azimuth can then be defined as $\Theta = \arctan\left(\frac{h_y}{h_x}\right)$, and the absolute offset as $h = \sqrt{{h_x}^2 + {h_y}^2}$. After gridding, the discrete data for the vertical component, $\mathbf{U}_z$, can be sorted in several domains, such as the midpoint-offset $\mathbf{U}_z[t, m_x, m_y, h_x, h_y]$, midpoint-offset-azimuth $\mathbf{U}_z\,[t, m_x, m_y, h, \Theta]$, and the shot-receiver domain $\mathbf{U}_z[t, s_x, s_y, r_x, r_y]$. Traces with common coordinates are usually averaged before being assigned to the same poition in a process named *binning*. Some bins, however, will have no traces assigned to it, and can be estimated through SSA reconstruction. Binning in itself creates further issues with

seismic data that go beyond the purposes of this thesis (see Carozzi and Sacchi (2021)), which assumes regular data throughout its tests. Moreover, since CMP binning is only appropriate for wave modes that have ideally symmetric raypaths around the midpoint, the datasets used in these examples are assumed to be sorted in the shot-receiver domain to attain maximum mode mixing in the field coordinate data components.

The ideal (noise-free and complete) data is shown in figure 3.14(a), and one example with $p = 90\%$ of missing traces is shown in figure 3.14(b). The idealized data, albeit unrealistic, has three dips, one dominating in each orthogonal component to illustrate pure P and SV modes in a radial-transverse coordinate frame. To approximate the case of "mode mixing" from one component onto the others, a rotation is used to map the data from its original frame to the (mixed) north-east coordinate frame. Each component of the data represents a regular tensor of size $100 \times 20 \times 20 \times 20 \times 20$, which were synthesized with a Ricker wavelet of central frequency $f_c = 20$ Hz with $\Delta t = 0.004$ ms. Each (pure) data component mainly contains a distinct dip with a magnitude of 1, and the projection artificially mixes these components, as shown below.

The SNRs in each component are still set to $[SNR_x, SNR_y, SNR_z] = [0.8, 1.0, 1.2]$. As above, the algorithm runs for 10 iterations with α linearly decreasing from 1 to 0 with iterations for simultaneous denoising and reconstruction. For comparison, each component is also reconstructed separately with the SSA filter, and all methods are applied in the frequency range of 0 to 60 Hz. The number of Lanczos iterations is set to $k = 10$ for SSA and QSSA, and $k = 12$ for the AQSSA given its increased signal subspace (figure 3.6). Figure 3.15 shows the reconstructed results for all components and methods. The reconstruction gain (equation 3.12) for each method, shown in table 3.1, indicates superiority in the AQSSA reconstruction. Both quaternionic approaches were more effective in denoising and reconstructing the multicomponent data despite their visual similarity.

Prediction gain (dB)			
Method	R_x	R_y	R_z
SSA	54.53	55.25	56.02
QSSA	63.07	59.32	59.69
AQSSA	65.11	61.09	62.26

Table 3.1: Reconstruction prediction gains for the 5D-3C linear events using the SSA, QSSA, and AQSSA.

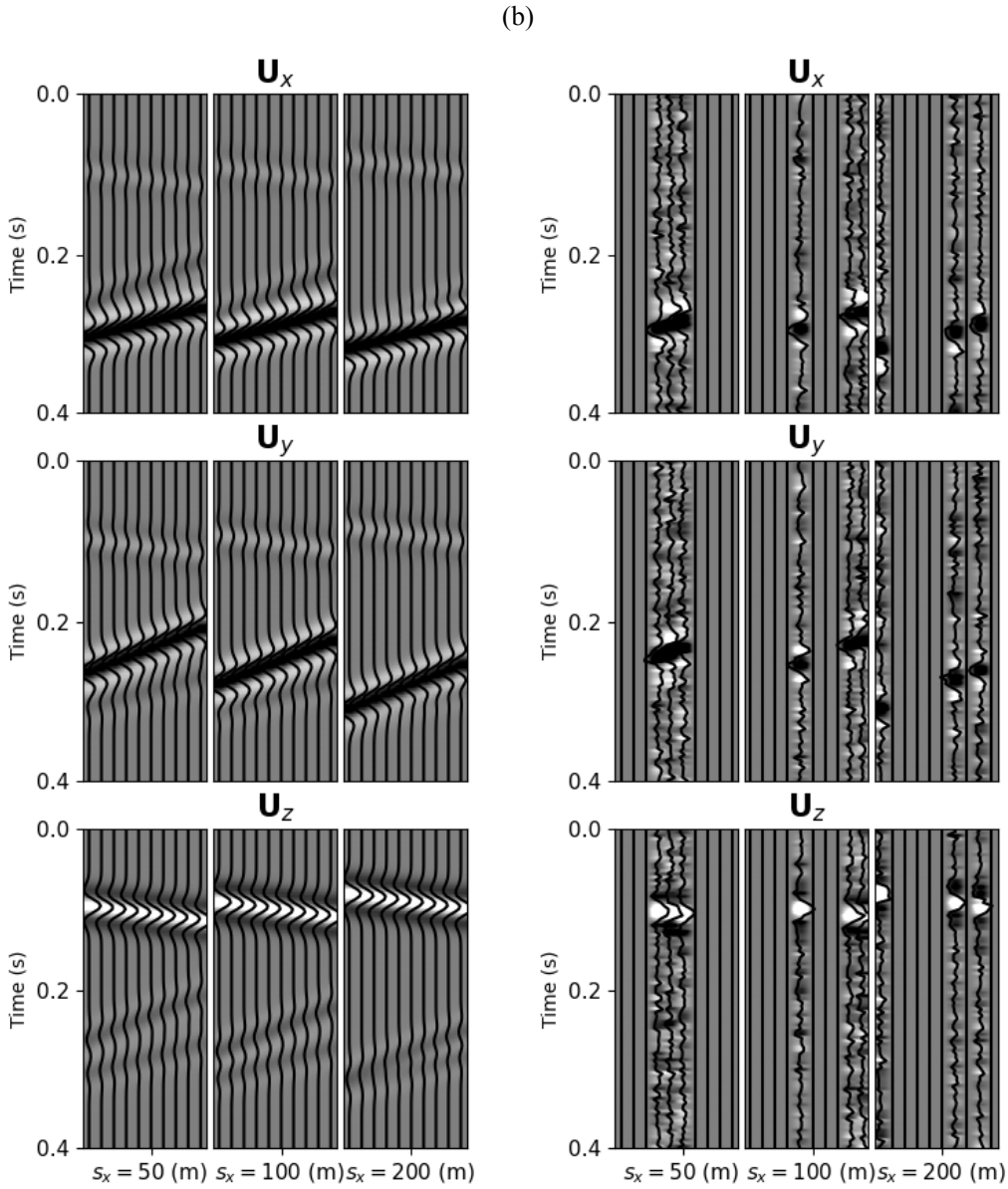

Figure 3.14: The 5D -3C ($\mathbf{U}_x$,$\mathbf{U}_y$,$\mathbf{U}_z$) linear events in its (a) ideal and (b) noisy and decimated forms, respectively.

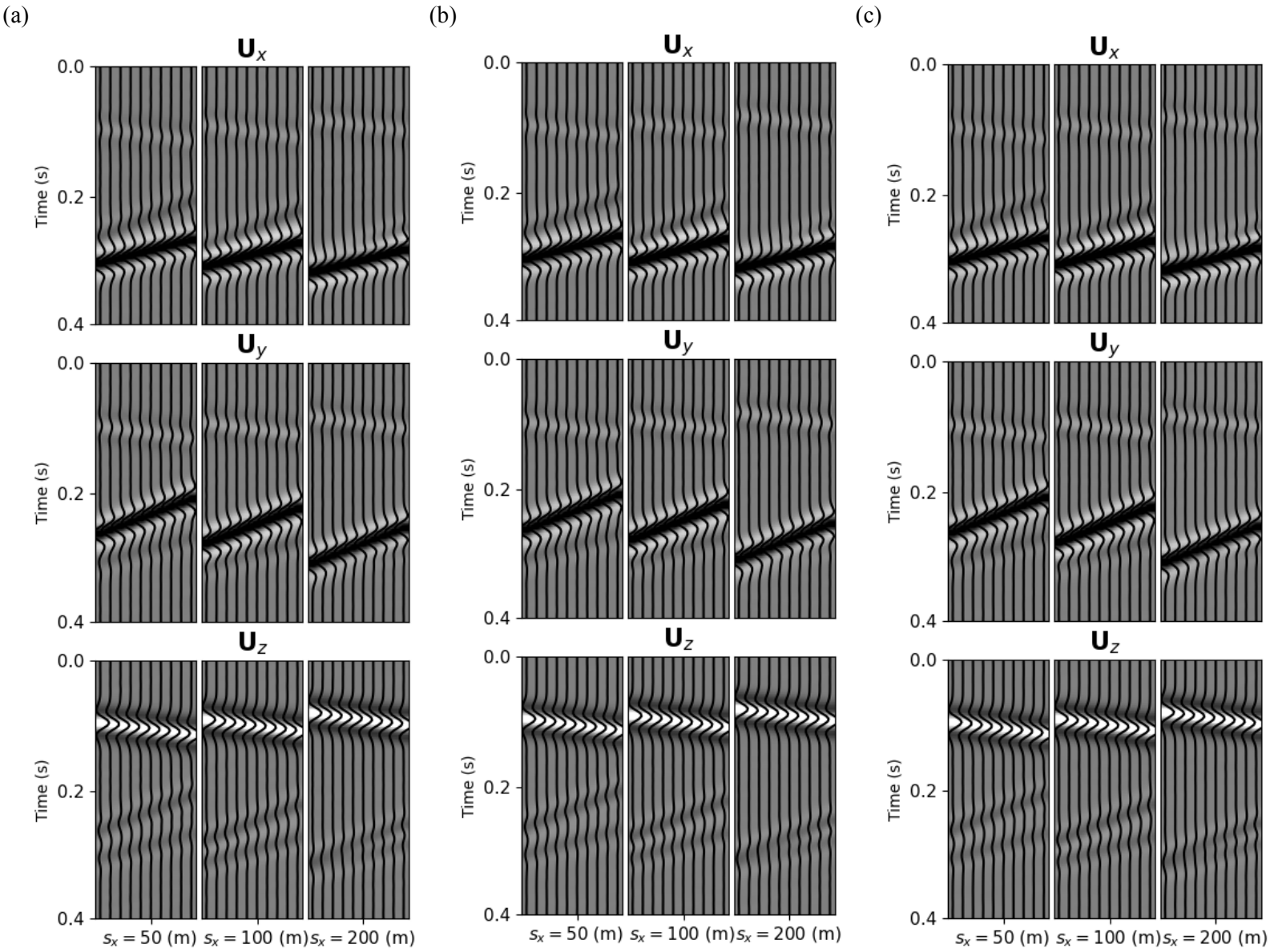

Figure 3.15: The reconstructed 5D-3C ($\mathbf{U}_x$,$\mathbf{U}_y$,$\mathbf{U}_z$) linear events with the (a) SSA, (b) QSSA, and (c) AQSSA filters. The reconstruction quality (dB) for each respective method method is $(54.53, 55.25, 56.02)$, $(63.07, 59.32, 59.69)$, and $(65.11, 61.09, 62.26)$ in x, y, z order.

Synthetic 3D-3C VSP example

A 3D-3C common-receiver gather from the VSP data shown in figure (figure 3.16(a)-(c)) is used for comparing the SSA and QSSA. Each component of the data was corrupted with Gaussian random noise following the same SNRs as above, and then decimated to have 60 % of its traces missing (figure 3.16(d)-(f)). The observed data, originally of size $218 \times 205 \times 205$, were patched into overlapping cubes of size $128 \times 32 \times 32$ with 50% overlap across each dimension. The algorithms run in the frequency range 0 to 50 Hz for all patches, with $k = 10$ random projections for SSA and QSSA, and $k = 15$ for AQSSA. The imputation algorithm runs for 50 iterations with a reinsertion parameter $\alpha = 0.5$.

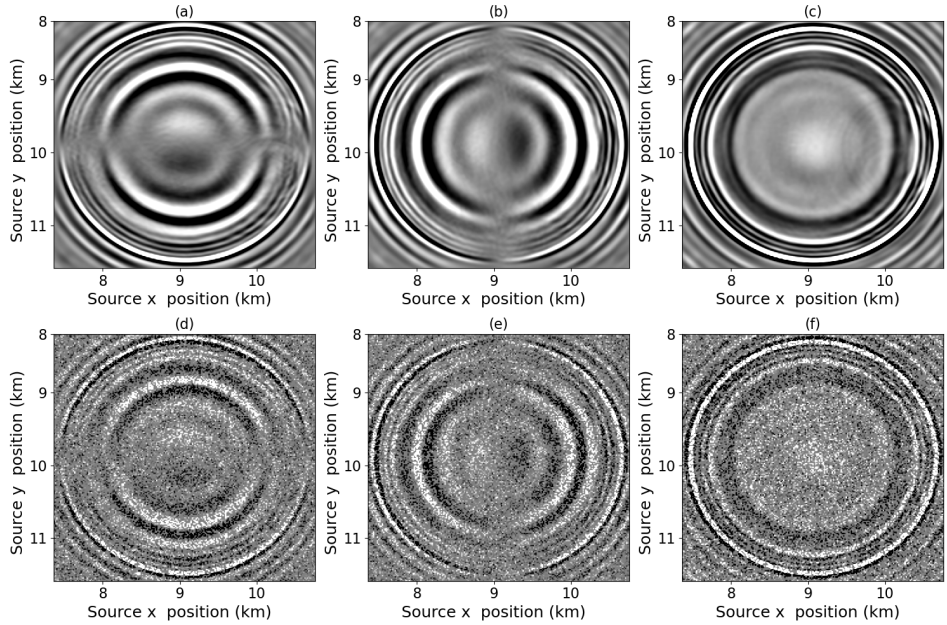

Figure 3.16: Time slice at 1.5s of noisy data decimated to have 50% of its traces missing.

The prediction gains (equation 3.12) shown in table 3.2 indicate that AQSSA outperforms SSA and QSSA in this test. Although the results obtained in this example are comparable, the AQSSA provides cleaner results than the SSA and QSSA, as shown in figure 3.17 and by the similar prediction gains. Figure 3.18 shows the difference between the reconstruction results and the ideal data (figure 3.16(a)-(c)), and all have similar levels of signal leakage in their residual panels. Even though the differences between the three approaches are arguably not significant, the quaternion approaches seem to provide good options for simultaneous trace recovery and random noise attenuation of vector-valued seismic signals.

Prediction gain (dB)			
Method	R_x	R_y	R_z
SSA	28.73	27.64	28.86
QSSA	29.32	28.14	29.18
AQSSA	30.43	29.88	30.17

Table 3.2: Reconstruction prediction gains for the 3D-3C VSP using the SSA, QSSA, and AQSSA.

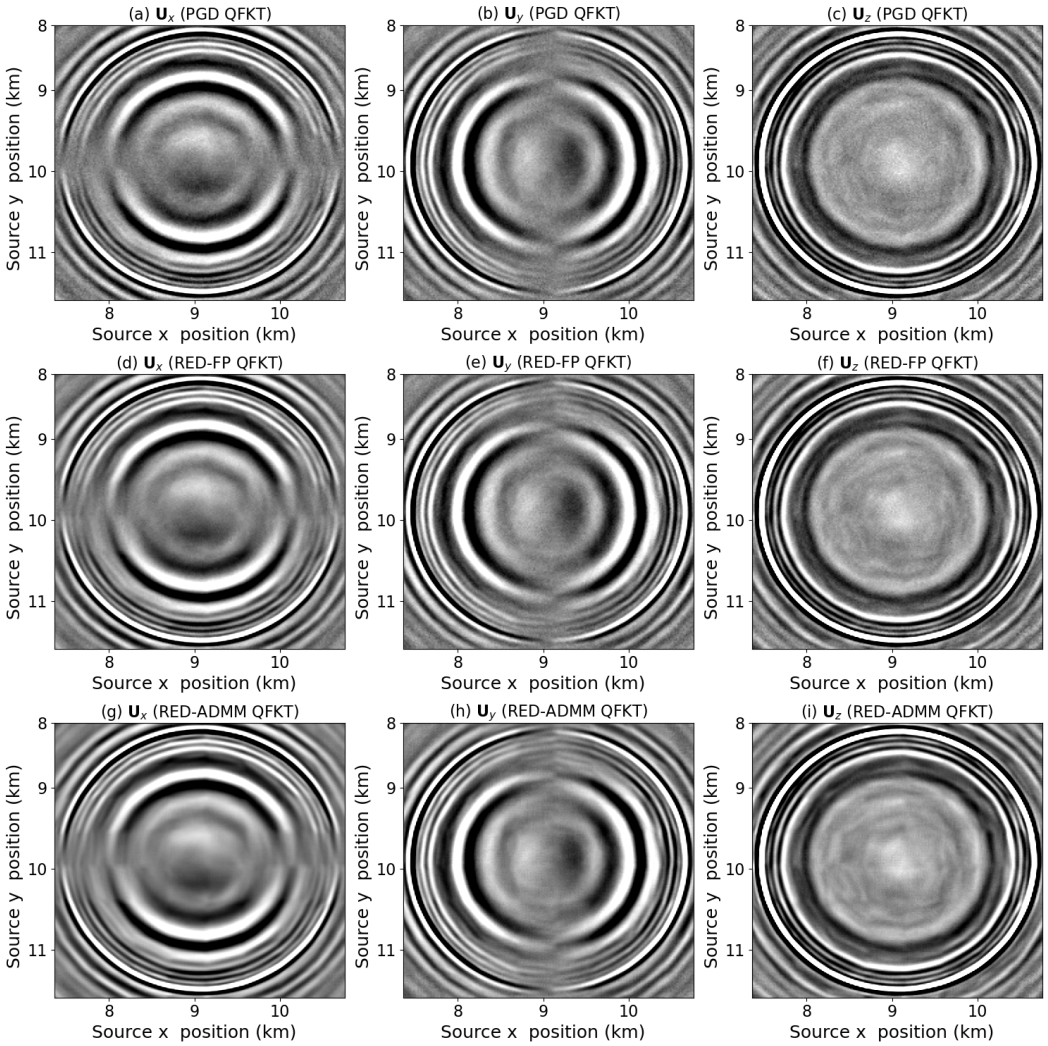

Figure 3.17: Time slice at 1.5s of reconstruction results for (a)-(c) SSA, (d)-(f) QSSA and (g)-(i) AQSSA.

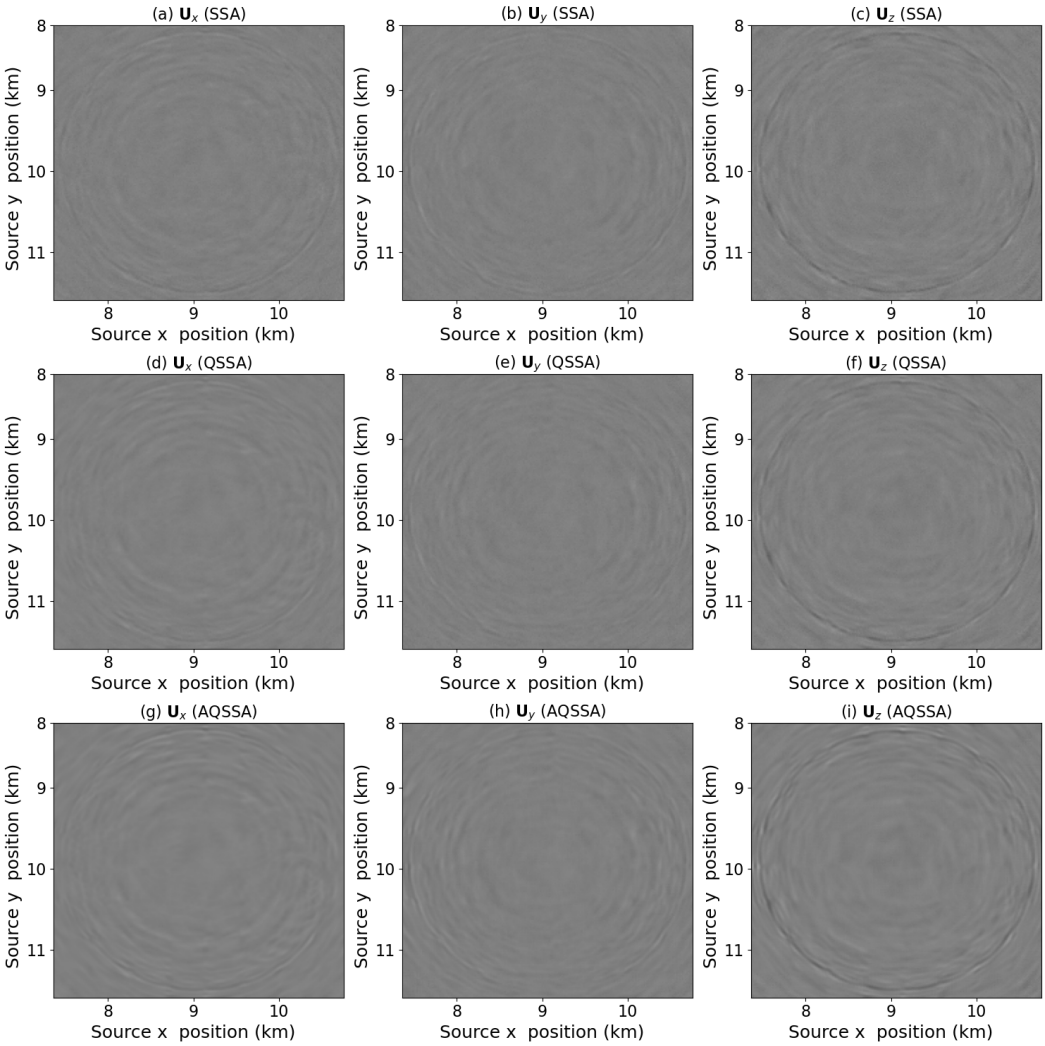

Figure 3.18: Time slice at 1.5 seconds of difference between reconstruction results (figure 3.17) and ideal data (figure 3.16(a)-(c)) for (a)-(c) SSA, (d)-(f) QSSA and (g)-(i) AQSSA.

3.4 Final remarks

3.4.1 Discussion

The results obtained with the quaternionic rank-reduction methods were similar to those obtained in the componentwise approach. Stanton and Sacchi (2013) discusses that the gains provided by quaternions can be associated with the spectral overlap of the components in the frequency domain. In practice, it was observed that if one component has very little to no signal, it compromises the quality of the reconstruction of the remaining components. However, if otherwise happens, the vector approach seems to be able to take advantage of the redundant information present in the signal components, introducing a slight improvement in its applications. Also, it is still unclear how to prove the increased rank of the augmented Hankel matrix in AQSSA. Including the quaternion involutions suggests that widely linear AR models (Ginzberg and Walden, 2013) should be used.

The number of Lanczos iterations is set following Gao et al. (2013), where a small subspace can be retrieved by allowing a small number of iterations, say $4-10$, in the case of simplistic datasets. Of course, this is data-dependent, and more complex datasets, such as those containing amplitude variations with source-receiver offset or curved events, might require a higher number of iterations. Nevertheless, adopting a windowing strategy reduces the effects of amplitude variations and curvature, which still allows one to set a small number of iterations per window. Also, notice that the increased signal subspace in the AQSSA requires more Lanczos iterations than SSA and QSSA. However, this parameter cannot be overestimated to avoid noise leakage into the processed sections.

The approach adopted in this thesis intends to be physically meaningful. Therefore, the seismic signal is represented by quaternions in the time domain. Note, however, that the different representations of quaternions, such as its Cayley-Dickson form (equation 2.23), can also be employed as a tool in the treatment of vector signals. For instance, it would be possible to process 2C post-stack datasets using quaternions in the Cayley-Dickson form following the approach given by Stanton and Sacchi (2013). However, there is no guarantee that post-stack datasets have preserved the vector-fidelity of the wavefield after processing and might not be suitable for the quaternion techniques. Further, another opportunity lies in the employment of biquaternions[3] for the representation of vector signals (e.g., Le Bihan et al. (2007)). This approach is attractive because each component of the vector-valued seismic data could be transformed to $f-x$ using the classic FT, which preserves conjugate symmetry in this domain and then be processed via biquaternion algebra.

[3]Biquaternions are quaternion numbers which have complex components.

3.4.2 Conclusions

This chapter has discussed the application of rank-reduction methods for multicomponent seismic data processing based on the natural ability of quaternions to represent vector-valued data. The methods entail the extensions of the SSA algorithm to the case of quaternion variables, and its main goal is to preserve and exploit the vector relationships in the data. Using the quaternionic representation, quaternion-based signal processing techniques discussed in this thesis, such as QFT and QSVD, have to be employed in the data processing steps, which illustrates a shift from scalar towards vector signal processing.

This chapter also extends the efficient formulations of the SSA to quaternion-valued seismic signals. The matrix decomposition methods adopted here are the quaternionic versions of the Lanczos bidiagonalization and the randomized QR decomposition. Both methods are valuable alternatives to the prohibitive QSVD. The efficient formulations of the QSSA made it possible to employ the method in the simultaneous denoising and reconstruction of multidimensional vector-valued datasets. The results presented in this chapter indicate that both QSSA and AQSSA offer reasonable alternatives to treating vector signals. In particular, the AQSSA can exploit second-order information from the input components, improving the final reconstruction results significantly.

One fundamental conclusion of this study is that the time-domain representation of vector-field seismic data via quaternion arrays does not affect the rank of the standard Hankel matrix built from its monochromatic frequency slices, which should be of low rank in the ideal case. On the other hand, the augmented Hankel matrix has increased rank due to the inclusion of the involutions in the process. However, it is still of low rank, and both approaches are useful for vector-data processing. The low-rankness character of the trajectory matrix in QSSA can be demonstrated through Q-AR models (Ginzberg and Walden, 2013) meaning that the quaternionic frequency slices of the data follow a predictive recursion given by a Q-AR model. Such is the basis for classic prediction filters in the case of scalar data. These filters, known as the $f - x$ deconvolution (FXDECON) or prediction filters, are widely used in the scalar seismic data processing. The next chapter introduces the technique under the quaternionic perspective to offer their hypercomplex extensions.

CHAPTER 4

Quaternion $f - x$ Widely-Linear Prediction Filters[1]

4.1 Introduction

Multicomponent seismology brought up the challenge of processing vector-valued seismic signals. Correspondingly, as discussed throughout this thesis, random noise suppression also gained a vectorial perspective and, although each of the acquired components could be processed separately, several publications address the topic following a vectorial approach. The shift in paradigm aims at exploiting the additional information available in the correlation between the components of vector-valued seismic data. For instance, one approach for random noise reduction of scalar seismic data is the frequency-space ($f - x$) deconvolution published by Canales (1984), known as FXDECON (Gülünay, 1986).

The FXDECON is a complex Wiener filtering technique that estimates the predictable portions of a given signal embedded in random noise via autoregressive (AR) models. Despite being already well-known, the FXDECON theory still attracts attention among researchers and practitioners. Recently, Gülünay (2017) examined the signal preservation ability of different formulations of the FXDECON on signal and noise sections. Moreover, Naghizadeh and Sacchi (2012) and Kamil et al. (2015) show how one can extend the FXDECON to the case of multicomponent seismic data through vector-autoregressive (VAR) models. Both publications indicate better random noise suppression when compared with its scalar counterpart due to the additional information exploited by VAR models.

[1] A version of this chapter is published in Bahia, B., and Sacchi, M. D. (2020). Widely linear denoising of multicomponent seismic data. Geophysical Prospecting, 68(2), 431-445.

The VAR-based methods assume that long-vectors can represent the components of the vectorial signal in the frequency domain. Notice that each component is Fourier transformed independently. This representation, however, can be regarded as restrictive to the processing of vector-valued signals since it does not preserve vectorial relationships between data components. As an alternative, vector signals can be numerically represented through hypercomplex numbers, such as the quaternions. Thanks to the recent developments in quaternion-based signal processing, one can now process vector-valued data following an approach that is not only physically consistent but also offers the possibility of making the most of the additional useful information acquired by multicomponent sensors.

In this chapter, following the advances in quaternion autoregressive (Q-AR) models introduced in Section 2.3.2, a hypercomplex version of the FXDECON is proposed, referred to as QFXDECON. This perhaps straightforward extension is based on the standard second-order statistics of quaternion signals, which is optimal for the case of proper signals only (Took and Mandic, 2009, 2011). In the case of improper signals, such as those sampled from processes with different variances in data components, it is possible to exploit the complete second-order statistics through widely-linear (WL) estimation of the underlying signal (Section 2.3.1). For such, one has to consider the $i-, j-, k-$ involutions of a quaternion signal in the prediction context, giving rise to what is referred to as the WL-QFXDECON.

4.2 Theory

The FXDECON (Canales, 1984) is an established technique for random noise attenuation from scalar seismic data. Its theory was best presented by Gülünay (1986) by combining the works by Treitel (1974) and Canales (1984). FXDECON corresponds to a linear prediction filter, which is based on autoregressive (AR) models. This section explains the theory of prediction filters via AR models, and how to obtain optimum filter coefficients following the least-squares theory (Ulrych and Sacchi, 2005).

4.2.1 $f - x$ prediction via AR Models

Consider a noise-free scalar seismic section $s(t, x)$ consisting of a single linear event of dip p

$$s(t, x) = u(t - px), \tag{4.1}$$

where $u(t)$ represents a source wavelet recorded in the data $s(t, x)$ as a function of time t and space x. The dip $p = \frac{1}{v} = \frac{\Delta t}{\Delta x}$ represents the horizontal slowness (or ray parameter) of the propagating wavefield. The Fourier representation of the data $s(t, x) \rightleftharpoons S(\omega, x)$ following

equation 4.1 is given by the phase shift property

$$S(\omega, n) = U(\omega)e^{-i\omega px} \tag{4.2}$$

where $U(\omega)$ represents the wavelet spectrum, and ω represents the temporal frequency. For regularly sampled data, the spatial dimension x can be discretized as $x = (n-1)\Delta x$, where $n = 1, \ldots, N$ represents the trace index in space with sampling interval Δx, then

$$S(\omega, x) = S_n = U(\omega)e^{-i\omega p(n-1)\Delta x}, \tag{4.3}$$

where the notation S_n was introduced for convenience, and the dependency on ω is omitted.

Notice that the change of variable $n = n - 1$ leads to

$$\begin{aligned} S_{n-1} = &\quad Ue^{-i\omega p((n-1)-1)\Delta x} && (4.4)\\ = &\quad Ue^{-i\omega p(n-1)\Delta x}e^{i\omega p\Delta x} && (4.5)\\ = &\quad S_n e^{i\omega p\Delta x}. && (4.6) \end{aligned}$$

Equivalently, one can write a forward linear recursion relating neighboring traces as

$$S_n = S_{n-1}f_1 \tag{4.7}$$

where $f_1 = e^{-i\omega p\Delta x}$. Therefore, linear events in the time-space $(t-x)$ domain are predictable in $f - x$ by means of recursive models such as that in equation 4.7, which represents an autoregressive model of order 1 with zero-mean and no innovation.

In the more general case of the superposition of K dipping linear events in $t - x$, it is well understood that these events correspond to the superposition of K plane waves in $f - x$. Following from the discussion above, these waves are represented by a superposition of complex exponentials written as

$$S_n(\omega) = \sum_{k=1}^{K} U_k(\omega)e^{-i\omega p_k(n-1)\Delta x}, \tag{4.8}$$

where p_k is the horizontal slowness of the k-th wave, and $U_k(\omega)$ is the Fourier transform of the source wavelet of the k-th wave. For each frequency in a range $[\omega_{min}, \omega_{max}]$, the FXDECON represents the signal composed by these complex exponentials by a zero-mean AR model of order K as

$$S_n = \sum_{i=1}^{K} S_{n-i}f_i + e_n, \quad n = K+1, \ldots, N. \tag{4.9}$$

Equation 4.9 can be used to recursively obtain the predictable portions of the signal S_n via the convolution of the data sequence, S_{n-i}, and the filter coefficients, f_i, where e_n is the innovation term. The above equation characterizes a K-order forward AR model since it predicts the current sample n based on its previous K samples. Similarly, the expression

$$S_n = \sum_{i=1}^{K} S_{n+i} b_i + e_n, \quad n = 1, \ldots, N-K \tag{4.10}$$

defines a backward recursion to predict the sample n based on its future K samples.

Least-squares Estimation of Filter Coefficients

Equations 4.9 and 4.10 can be written in matrix-vector form as (Ulrych and Sacchi, 2005)

$$\mathbf{d} = \mathbf{X}\mathbf{f} + \epsilon, \tag{4.11}$$

where $\mathbf{X}_{M\times K}$ is a transient-free convolution matrix built from the data samples, $\mathbf{d}_{M\times 1}$ is the data vector, $\mathbf{f}_{K\times 1}$ is the vector of unknown filter coefficients (forward or backward), $\epsilon_{M\times 1}$ is the vector of innovation terms, and $M = N - K + 1$.

The optimum filter $\tilde{\mathbf{f}}$ can be estimated in a least-squares sense by minimizing

$$J(\mathbf{f}) = \|\epsilon\|_2^2 = \|\mathbf{d} - \mathbf{X}\mathbf{f}\|_2^2, \tag{4.12}$$

which represents the innovation energy. The solution to this problem is known to be

$$\tilde{\mathbf{f}} = (\mathbf{X}^H\mathbf{X})^{-1}(\mathbf{X}^H\mathbf{d}), \tag{4.13}$$

where the superscript "^{H}" denotes the transpose-conjugate of a matrix. For a more algorithmic oriented description, the reader is referred to Gülünay (2017).

The enhanced complex exponentials can then be obtained by convolving the optimum forward and backward filters with the data, and finally averaging its overlapping results (Naghizadeh and Sacchi, 2012; Gülünay, 2017). The combination of forward and backward prediction is of great interest as it reduces possible errors produced by the individual application of forward (or backward) prediction, such as incomplete prediction of traces along a given direction. Besides, the filter length plays an important role in predicting the signal to avoid losses of useful information. For instance, for a noise-free single dipping event, *i.e.*, a single complex exponential in $f - x$, an AR model of order 1 ($K = 1$) should be enough to predict this event properly. For sections with more linear events, say K, the filter length should be no less than K if a proper prediction is to be achieved. However, due to distortion

in amplitudes caused by random noise K linear events cannot be accurately predicted by AR models of order K. Therefore, one should favor larger prediction filters since they are known to perform better in terms of signal preservation (Gülünay, 2017).

4.2.2 $f - x$ prediction via Q-AR Models

The above illustrates why FXDECON has its place in the processing stage of scalar seismic data. It is simple, computationally efficient, and effective in random noise suppression. In the case of non-linear events in $t - x$, the assumption of dipping linear events can be approximated by applying suitable windowing strategies to the data. This same technique can also be implemented in 3D using 2D convolutions under the assumption of planar events in time (Chase, 1992). Furthermore, one can attenuate random noise in the case of multicomponent measurements via VAR models (Naghizadeh and Sacchi, 2012; Kamil et al., 2015). The development of autoregressive models for quaternion signals (Q-AR) (Ginzberg and Walden, 2013) introduces another opportunity in the treatment of vector-valued signals.

Review on quaternion-based signal processing

The 4D nature of quaternions facilitates the numerical representation of three- and four-component signals, such as multicomponent seismic data, allowing real physical phenomena to be modelled in the space they actually happen. For instance, recall that full quaternion arrays can represent four-component signals as

$$\mathbf{q}[n] = \mathbf{a}[n] + \mathbf{b}[n]i + \mathbf{c}[n]j + \mathbf{d}[n]k, \tag{4.14}$$

and pure quaternion arrays can represent three-component signals

$$\mathbf{q}[n] = \mathbf{b}[n]i + \mathbf{c}[n]j + \mathbf{d}[n]k, \tag{4.15}$$

where $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ are the components of the quaternion signal $\mathbf{q}$, and $n = [1, N]$ represents temporal or spatial samples. Such models have been increasingly adopted for processing vector-field seismic datasets as it allows one to naturally preserve and exploit the vector relationships of the data (Le Bihan and Mars, 2004). The practicioner has to notice, however, that quaternion-based signal processing tools have to be employed instead of their classic versions if one has decided to define the data as a quaternion array. For formulating the QFXDECON, this chapter uses the QFT and Q-AR models, which are discussed next.

Quaternion Fourier Transform Notice, first, that the quaternionic extension of the FXDECON will require the data to be transformed to $f - x$, as usually done for scalar data

by means of the classic Fourier transform. Similarly to the QSSA, therefore, the quaternion-valued seismic data is to be transformed to $f-x$ by means of the QFT, which allows signal processing techniques performed in the frequency domain, such as the FXDECON itself, to be generalized to quaternion-defined datasets (Ell and Sangwine (2007); Ortolani et al. (2017); Bahia and Sacchi (2019b)). One key difference, as previously discussed, is the lack of conjugate symmetries in the frequency representation of quaternion-valued signals. In addition, quaternion algebra is non-commutative, and the QFT is not uniquely defined. It requires the definition of a *side* and of an arbitrary pure unit quaternion ($\mu^2 = -1$), called the *eigenaxis*, for the transformation (Section 2.3.4). The QFT is analogous to the classical Fourier transform, and most of its properties are kept unchanged in both cases. For example, following equations 2.62 or 2.63, application of the QFT to the quaternion-defined seismic data composed of K dipping events gives rise to plane waves defined by hypercomplex exponentials and amplitudes in $f-x$, similarly to equation 4.8.

$f-x$ Predictability and Second-order Statistics In the quaternion domain, one can show $f-x$ predictability of linear events following the same steps as for equations 4.1-4.7, now using equations 2.62 or 2.63 via the phase shifts property. In that sense, notice how the most straightforward derivation starts with a right-sided QFT since it is conformant with the matrix-vector notation adopted in equation 4.11. Therefore, in this chapter, the QFT is always applied from the right using an eigenaxis $\mu = \frac{1}{\sqrt{3}}(i+j+k)$. Furthermore, the predictability of linear events in the quaternion domain allows modeling vector-valued seismic data composed of linear events synthesized by hypercomplex phase shifts. More importantly, with the formalization of Q-AR models by Ginzberg and Walden (2013), the $f-x$ predictability in the quaternion domain can be further exploited in two different formulations of the FXDECON to the case of quaternion-valued seismic data.

The first, in the case of proper quaternion signals (Section 2.3.1), corresponds to a strictly linear prediction scheme that is understood as the trivial extension of the procedure described above. The method is named QFXDECON, and it requires the $t-x$ quaternion-valued seismic data to be transformed to $f-x$ via the QFT. For each frequency slice in both sides of the spectrum, the same process of filter estimation and data filtering takes place. Since the Weiner solution for quaternionic predictive filters follows the same form as in equation 4.13 (Took and Mandic, 2009), the main difference between FXDECON and QFXDECON is that the data-dependent variables ($\mathbf{X}$ and $\mathbf{d}$) and filter coefficients ($\mathbf{f}$) are now quaternion valued. However, such extension could lead to suboptimal performance as useful additional second-order complementary information is being ignored.

In the context of linear estimation, as in the $f-x$ prediction discussed earlier, taking full advantage of the second-order information in quaternion signals is also possible. Hence, to

exploit the second-order statistics of quaternion valued signals, one has to consider a quaternion WL model. This is done by allowing additional linear operations on the quaternion involutions in equation 4.11 as

$$\mathbf{d} = \mathbf{X}\mathbf{f} + \mathbf{X}^i\mathbf{f}_i + \mathbf{X}^j\mathbf{f}_j + \mathbf{X}^k\mathbf{f}_k + \epsilon, \tag{4.16}$$

which can be recast as

$$\mathbf{d} = \mathbf{X}^a\mathbf{f}^a + \epsilon \tag{4.17}$$

with $\mathbf{X}^a = \begin{pmatrix}\mathbf{X} & \mathbf{X}^i & \mathbf{X}^j & \mathbf{X}^k\end{pmatrix}$ and $\mathbf{f}^a = \begin{pmatrix}\mathbf{f} & \mathbf{f}^i & \mathbf{f}^j & \mathbf{f}^k\end{pmatrix}^T$. The Wiener solution for the widely-linear model also has the general form of the standard solution given in equation 4.13, but it is now based on the augmented data matrix $\mathbf{X}^a$. Note that $\mathbf{X}, \mathbf{X}^i, \mathbf{X}^j$, and $\mathbf{X}^k$ are transient-free convolution matrices; then the product $\mathbf{X}^{aH}\mathbf{X}^a$ provides an estimator of the augmented covariance matrix of the type given in equation 2.41. In addition, the augmented filter to be estimated is four times longer than before, but its action represents four different filtering procedures each given by $(\mathbf{f}, \mathbf{f}^i, \mathbf{f}^j, \mathbf{f}^k)$. This is of great interest, as already mentioned, since it allows for better signal preservation as it will be shown in the Examples section below.

4.3 Examples

This section describes the tests of the effectiveness of the techniques presented in this chapter. Through 2D-3C synthetic examples and one shot gather of a 3C ocean bottom cable survey (Naghizadeh and Sacchi, 2012), it is possible to show that widely-linear Q-AR models provide enhanced signal preservation when compared to its linear counterparts. The datasets are composed of one vertical and two horizontal components, but the methods could easily incorporate the pressure component in 4C datasets as well.

4.3.1 3C Synthetic Data Examples

The synthetic example consists of a 2D-3C dataset with 4 distinct dips in the time domain (figure 4.1(a)-(c)), having 1 second of record length with temporal sampling $\Delta t = 2$ ms, and 100 traces spaced 10 meters apart. Notice how the three components show the same reflection travel-time signatures, but have different polarities and amplitudes among them. Since the data is composed of linear events only, it is not necessary to use any windowing strategy. Also, notice that the figures display only one every four traces in the sections. One can corrupt the data following the assumption of band-limited Gaussian noise such that each component has different signal-to-noise ratios (SNR). Figure 4.1(d)-(f) shows the dataset with added noise sequences having distinct SNRs as $[SNR_x, SNR_y, SNR_z] = [0.4, 0.3, 0.5]$.

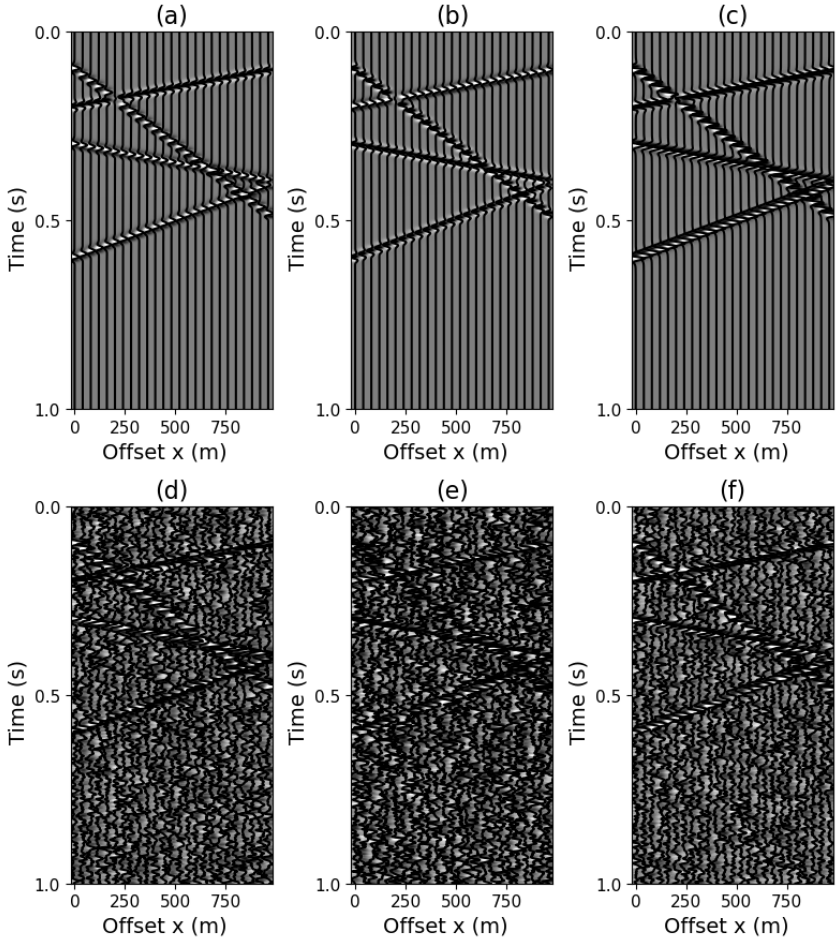

Figure 4.1: (a) $\mathbf{U}_x$ (b) $\mathbf{U}_y$ and (c) $\mathbf{U}_z$ ideal synthetic multicomponent data, and corrupted with band-limited random noise with $[SNR_x, SNR_y, SNR_z] = [0.4, 0.3, 0.5]$ in (d)-(f).

Using a 4-th order AR and Q-AR models ($K = 4$), one can attenuate random noise from the input sections shown in figure 4.1(d)-(f). The overall length of the augmented filter is $K^a = 16$ in this case. All methods are implemented using transient-free forward and backward prediction filters, and a prewhitening value of 0.01% was added to the main diagonal of all techniques and tests presented here. Figures 4.2 and 4.3 show, respectively, the filtering results and the estimated noise section (the difference between the input section and the filtering result) for each method applied in the frequency range of 0 to 100 Hz.

As already mentioned, FXDECON is an efficient technique in random noise suppression, as shown in figures 4.2(a)-(c). It is clear how the weaker events, especially in the $\mathbf{U}_y$ component, could be recovered from its noisy background. From the differences shown in figure 4.3(a)-(c), however, it is possible to notice portions of the predictable signal leaking into the noise sections. As already previously discussed, this is a consequence of the random noise present in the data, which introduces amplitude variations in the traces. Although longer filters can achieve better results, the goal of this section is to compare the signal preservation capability of each technique using the ideal optimum filter length.

Figure 4.2(d)-(f) show the results of the QFXDECON filtering. Qualitatively, one can point to improvements in random noise suppression when compared with the FXDECON results (figure 4.2(a)-(c)). However, the technique also has the drawback of signal leakage into the noise space (figure 4.3(d)-(f)). On the other hand, the WL-QFXDECON (figures 4.2(g)-(i) and 4.3(g)-(i)) could achieve enhanced signal preservation as a result of the additional information exploited from the correlation between data components, and due to the longer filter estimated in its formulation. However, also notice that the WL-QFXDECON provides higher levels of random noise in the filtered sections. This leakage happens because of the redundancy created in the noise samples during the augmentation step in equation 4.16, which induces the prediction of some noisy samples. Nevertheless, the WL-QFXDECON provides the best signal preservation scheme among its linear and componentwise counterparts.

Due to the similarity of the results and for completeness in the comparison, it is rather useful to assess the effectiveness of the denoising methods by means of the reconstruction quality R_i (in dB) given by

$$R_i = 10 \log_{10} \left(\frac{\|\mathbf{U}_i^o\|_F^2}{\|\mathbf{U}_i - \mathbf{U}_i^o\|_F^2} \right), \tag{4.18}$$

where $\mathbf{U}_i^o$ and $\mathbf{U}_i$ are the true (noise-free) and denoised data, respectively, with $i = (x, y, z)$ representing the component of the data. The subscript $_F$ represents the Frobenius norm of a matrix. As defined, higher prediction gains R_i indicates better performance by a given denoising methods.

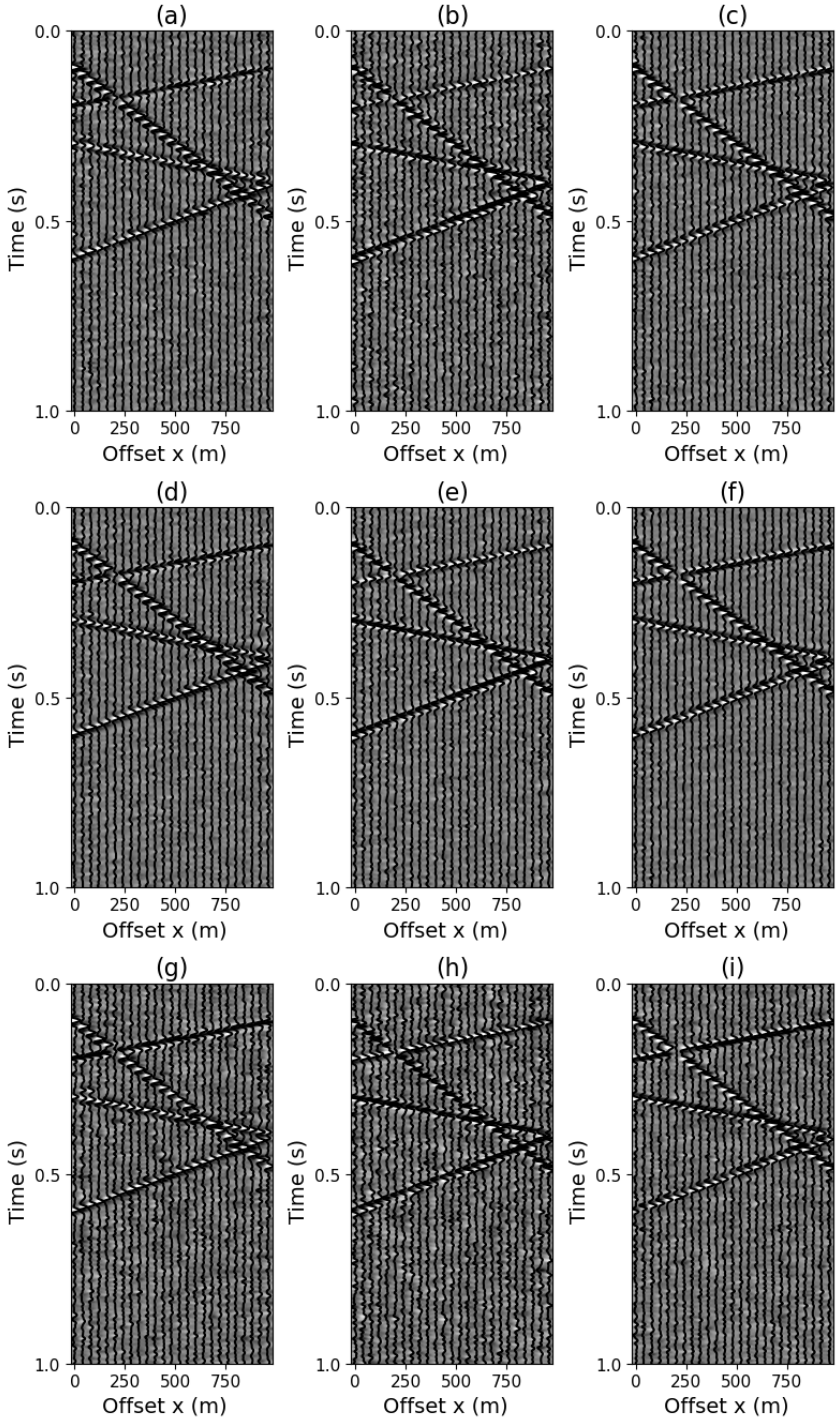

Figure 4.2: Denoising results with $K = 4$ for each component ($\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$) in the (a)-(c) FXDECON, (d)-(f) QFXDECON, and (g)-(i)WL-QFXDECON.

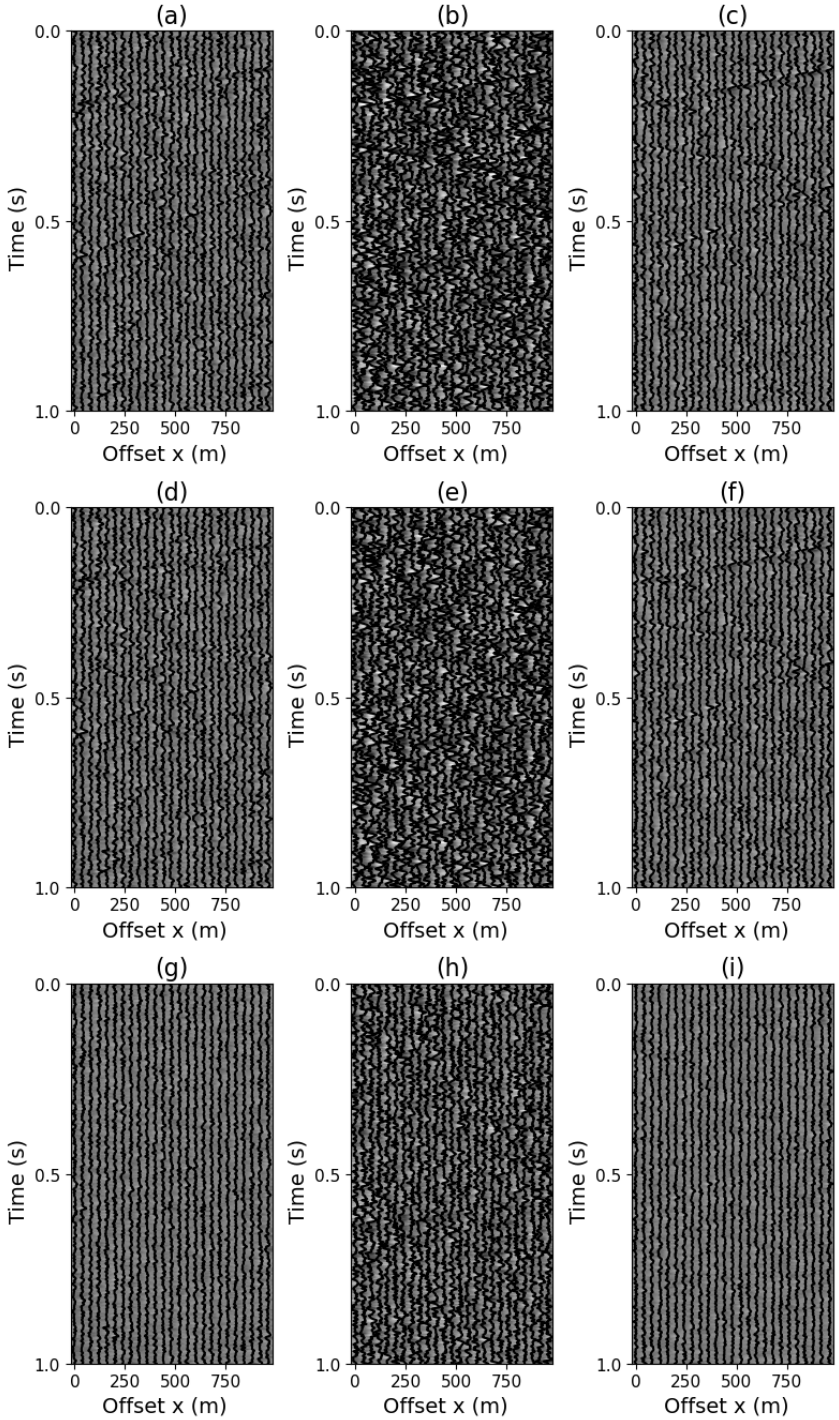

Figure 4.3: Estimated noise sections with $K = 4$ for each component ($\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$) in the (a)-(c) FXDECON, (d)-(f) QFXDECON, and (g)-(i)WL-QFXDECON.

The same example discussed above was run with varying levels of random noise and increasing filter length from 1 to 10. The initial SNRs ($[SNR_x, SNR_y, SNR_z] = [0.4, 0.3, 0.5]$) were successively increased by unity also from 1 to 10. Figure 4.4 portrays the prediction gains obtained by each method for each component following equation 4.18. This result indicates that the widely-linear approach outperforms its linear counterparts in the case of lower SNR and shorter prediction filters. Notice, however, that the performance of the FXDECON and QFXDECON do approach that of WL-QFXDECON in the case of high SNR and longer prediction filters. Not only this is due to the longer prediction filters being employed by FXDECON and QFXDECON, but also because the augmentation step of the widely-linear approach introduces noise in the final section which reduces its reconstruction quality despite the higher levels of the signal being recovered. Besides, the QFXDECON provided slightly better prediction gains when compared with the componentwise FXDECON.

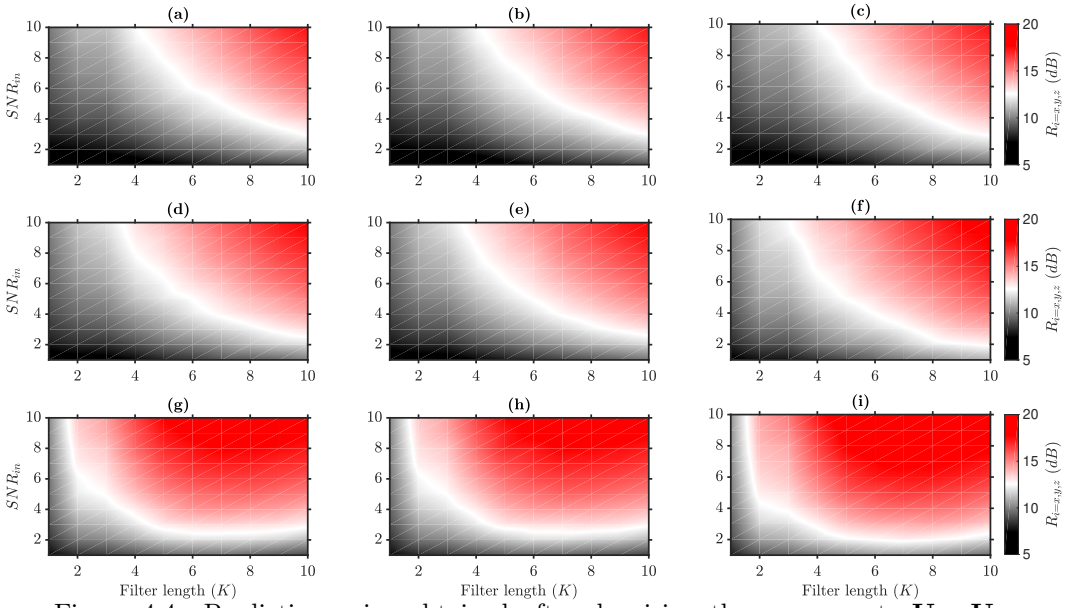

Figure 4.4: Prediction gains obtained after denoising the components $\mathbf{U}_x$, $\mathbf{U}_y$, and $\mathbf{U}_z$ using (a-c) the FXDECON, (d-f) for QFXDECON, and (g-i) for WL-QFXDECON.

4.3.2 3C OBC Field Data Example

Figure 4.5 shows the two horizontal and the vertical components of an OBC dataset. Surprisingly, the vertical component of this dataset has the most detectable levels of random noise. Due to the presence of curved events, the whole dataset is subdivided into overlapping patches as to approximate the assumption of linear dips in the time domain. All techniques (FXDECON, QFXDECON, and WL-QFXDECON) are applied to each window separately, which are then added back together to recover the processed data. The patches have 100 time samples and 50 traces each, with overlapping of 20% in both dimensions. The filter length is $K = 3$ for all methods, which run in the frequency range $[0, 100]$ Hz.

Figures 4.6, 4.7, and 4.8 display the results of the FXDECON, QFXDECON, and WL-QFXDECON, respectively. Despite the higher levels of noise in the WL-QFXDECON results being noticeable, it is arguable that all methods have good performance in eliminating random noise from the vertical component. Thus, a more interesting analysis would be devoted to their signal preservation capability. The estimated noise sections were computed for all components, and are displayed in the bottom rows of its respective filtering results. Once more, these results indicate that both FXDECON and QFXDECON have higher levels of coherent energy leakage in the estimated noise sections than WL-QFXDECON.

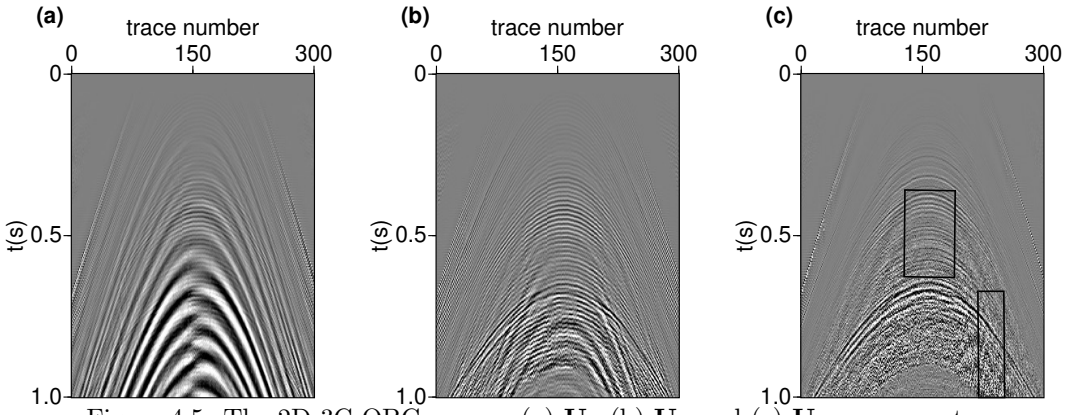

Figure 4.5: The 2D-3C OBC survey. (a) $\mathbf{U}_x$ (b) $\mathbf{U}_y$ and (c) $\mathbf{U}_z$ components.

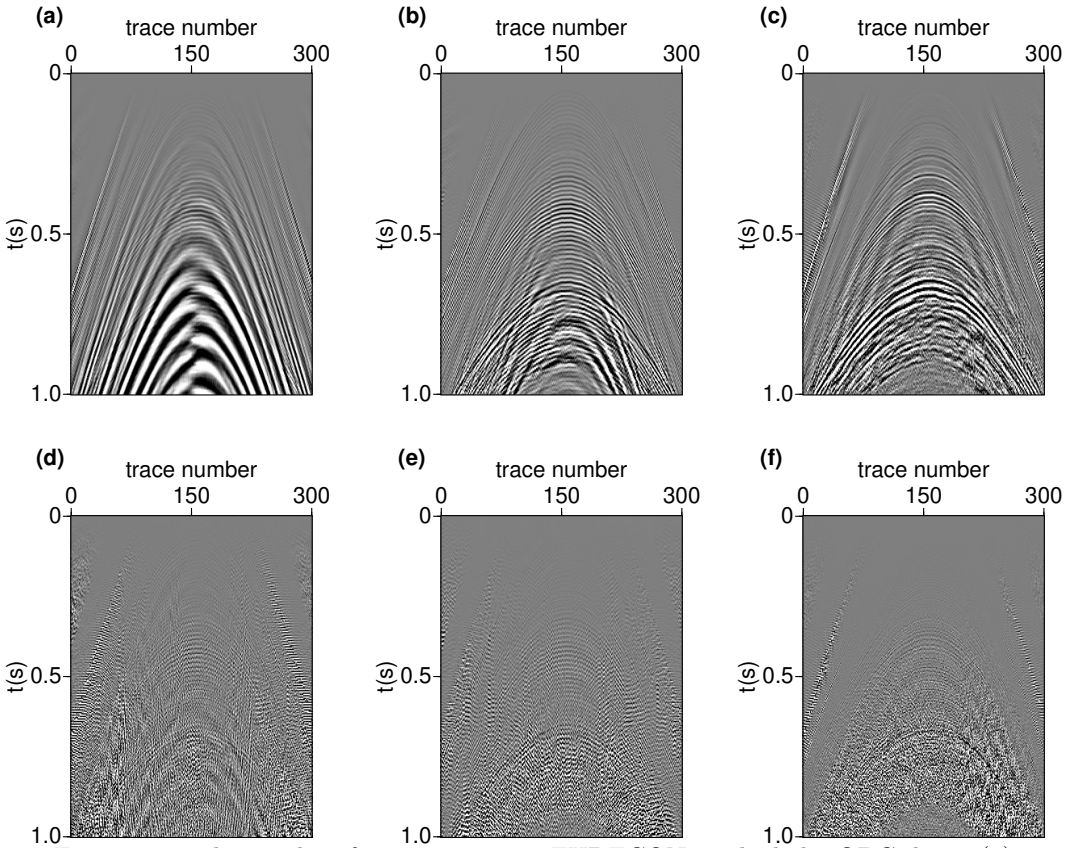

Figure 4.6: The results of componentwise FXDECON applied the OBC data. (a) $\mathbf{U}_x$ (b) $\mathbf{U}_y$ and (c) $\mathbf{U}_z$ components. (d)-(f) Respective estimated noise sections.

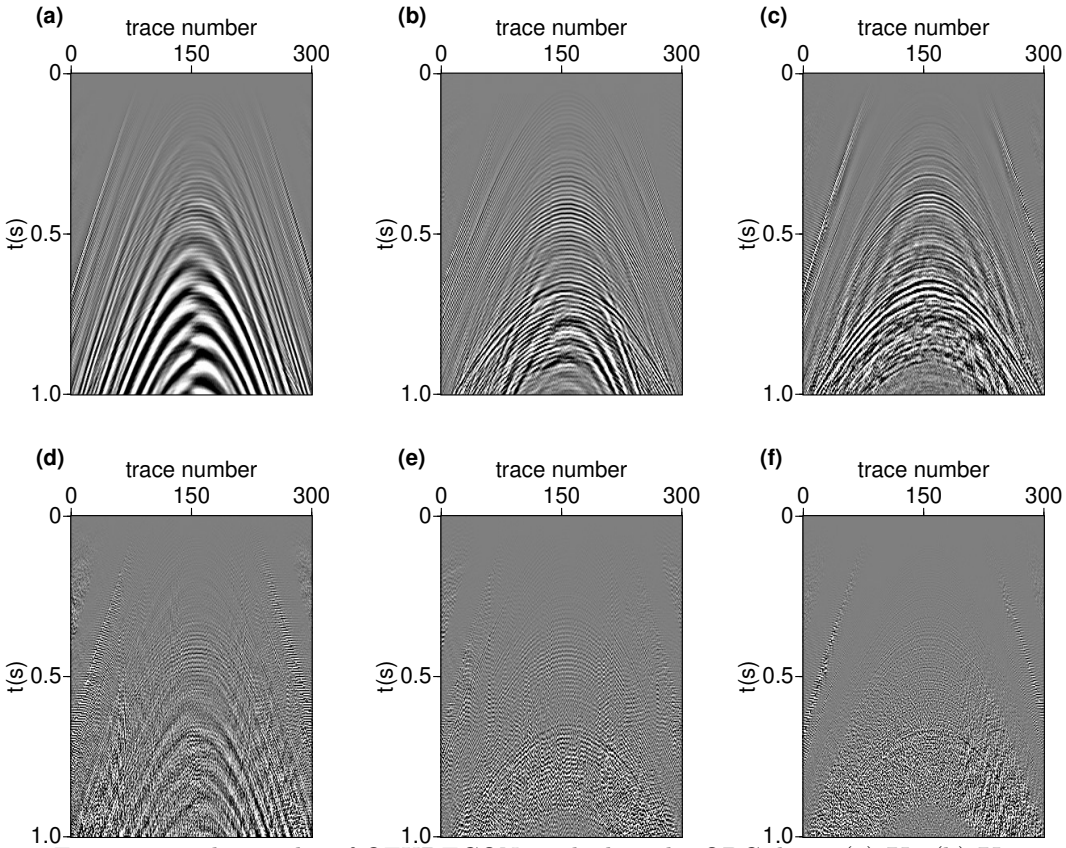

Figure 4.7: The results of QFXDECON applied to the OBC data. (a) $\mathbf{U}_x$ (b) $\mathbf{U}_y$ and (c) $\mathbf{U}_z$ components. (d)-(f) Respective estimated noise sections.

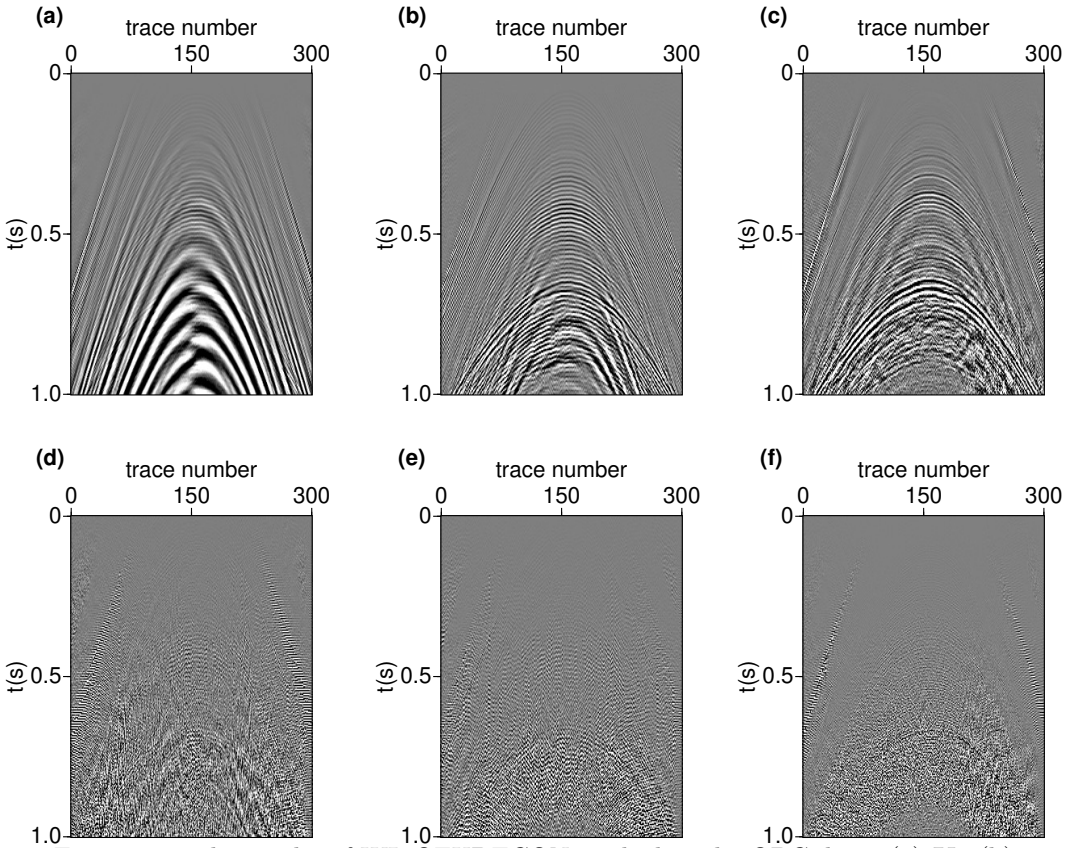

Figure 4.8: The results of WL-QFXDECON applied to the OBC data. (a) $\mathbf{U}_x$ (b) $\mathbf{U}_y$ and (c) $\mathbf{U}_z$ components. (d)-(f) Respective estimated noise sections.

For a more detailed view, figure 4.9 shows a zoom into the upper and lower windows highlighted in the raw vertical component (figure 4.5**(c)**). Notice that the figures display one of every other trace in the horizontal axes for each window. Figures 4.10 and 4.11 show the estimated signal and noise sections for the upper and lower windows, respectively, in each method. The signal sections (top rows) for all methods are fairly clean, confirming the efficiency of the FXDECON and its hypercomplex extensions. The noise sections (bottom rows) show that signal leakage is still present in the linear prediction schemes, as indicated by the red arrows. The WL-QFXDECON filtered sections (figures 4.10**(c)** and 4.11**(c)**) have higher levels of random noise, but its noise sections (figures 4.10**(f)** and 4.11**(f)**) show very little signal leakage. Nevertheless, these results give further substance to the claims of better signal preservation by the WL-QFXDECON. Though one can argue that a few events are more coherent in the FXDECON than in its hypercomplex versions, one should bear in mind that FXDECON has not been able to recover as many events as quaternion-based algorithms. Figure 4.11 shows that signals have not been properly recovered by FXDECON; these signal are present in the in the QFXDECON and WL-QFXDECON results.

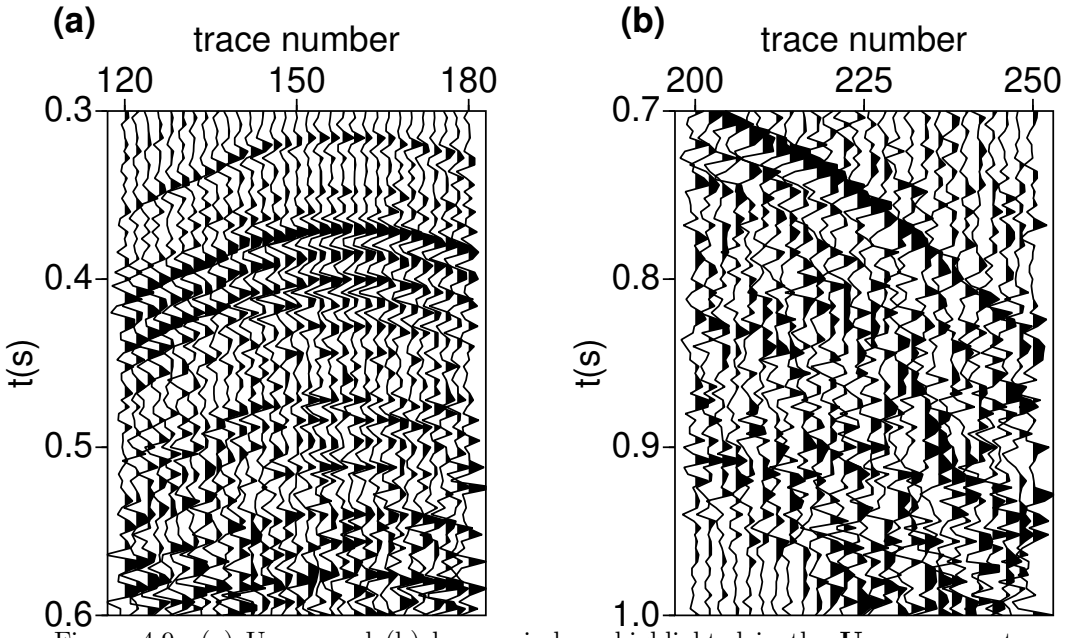

Figure 4.9: (a) Upper and (b) lower windows highlighted in the $\mathbf{U}_z$ component (figure 4.5).

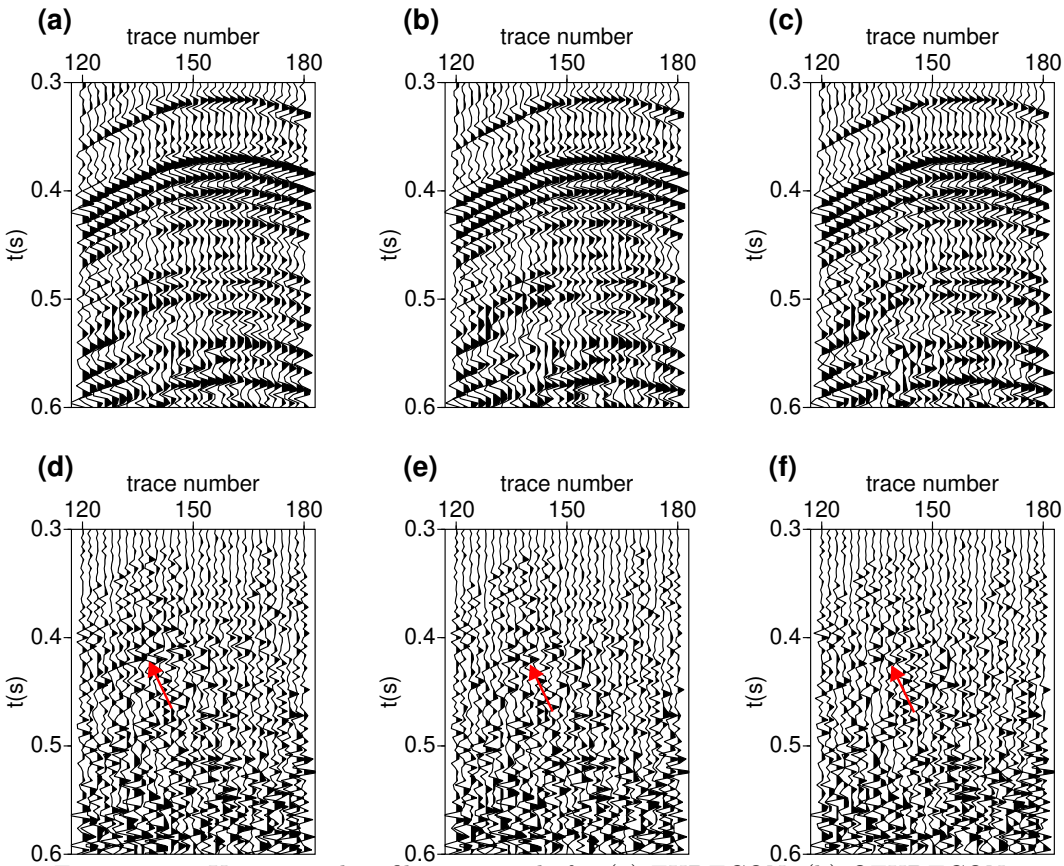

Figure 4.10: Upper window filtering result for (a) FXDECON, (b) QFXDECON, and (c) WL-QFXDECON. (d)-(f) Respective estimated noise sections.

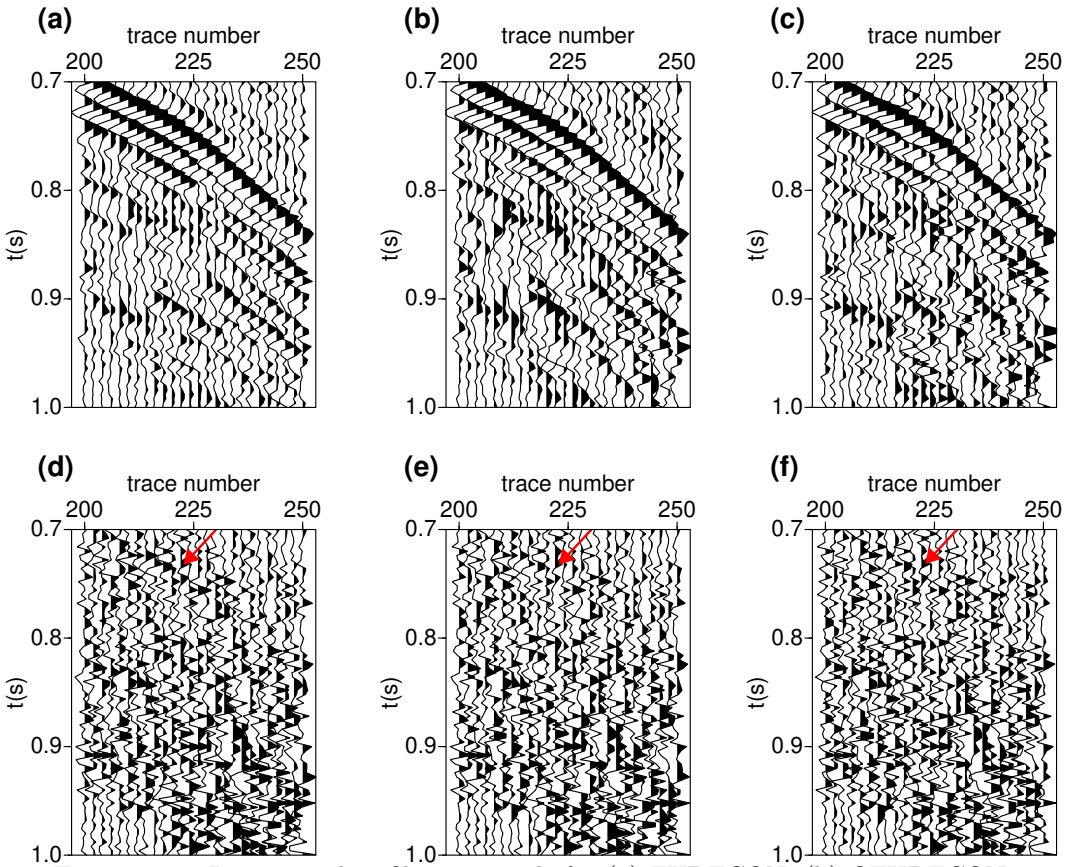

Figure 4.11: Lower window filtering result for (a) FXDECON, (b) QFXDECON, and (c) WL-QFXDECON. (d)-(f) Respective estimated noise sections.

4.4 Final Remarks

4.4.1 Discussion

It is essential to recall that the approaches presented in this chapter are consistent with the physical space of the data acquisition. Thus, this chapter focused on using quaternions to represent the multicomponent seismic signals in the pre-stack shot-receiver domain. Of course, when working with quaternion-defined seismic data, it is necessary to employ processing techniques based on the non-commutative quaternion algebra. Even though the main steps in the processing stage remain the same, this particularity can introduce a few differences.

Regarding processing times, the FXDECON technique is known to be computationally efficient. The same is true for its quaternion formulations, but these techniques have greater computational complexity than its componentwise counterpart given the quaternion multiplication. Furthermore, quaternion-based processing of 3C or 4C datasets in the frequency domain does contain all known and explored properties of the Fourier transform, such as conjugate symmetries. Thus, the techniques have to be applied on both sides of the spectrum, increasing processing times. For instance, while applying the FXDECON in the synthetic example using $K = 4$ takes on average 0.1 seconds for a single component, while the QFXDECON takes 1.1 seconds, and the WL-QFXDECON takes 3.4 seconds for all three. Finally, the additional parameters required by quaternion-based methods refer to the application of the QFT, which requires the choice of a side and an eigenaxis for the transformation.

For random noise reduction applications, it is possible to extend the FXDECON filter to the case of multicomponent seismic data using Q-AR models. Likewise the FXDECON, the techniques are relatively simple, easy to implement, and very useful in random noise suppression. Widely-linear Q-AR models are introduced as a novelty in the treatment of vector-valued seismic data, which provided encouraging results in signal estimation. The advantages offered by widely-linear models are related to the augmentation step in equation 4.16, which increases signal redundancy and filter length. The main outcome of this formulation is that the predictable portions of the signal could be better preserved in the filtered sections. However, the WL-QFXDECON results also show higher levels of random noise than the QFXDECON and FXDECON. This is also related to the augmentation step, which increases redundancy in the noise samples as well.

Application of further filtering to the components, following an iterative filtering approach (e.g., Canales (1984)), could lead to better results but this was not the focus of this chapter. Also, one should bear in mind that all of the components are assumed to contain signal

since noise-only components can degrade the final results of the quaternion components (Bahia and Sacchi, 2019b). In such situations, one is likely to be dealing with uncorrelated components which will yield vanishing off-diagonal terms in the augmented covariance matrix (equation 2.41). In that matter, componentwise techniques might better suited for data processing. Finally, the literature on geophysical applications of AR models entails not only denoising applications, but also data interpolation (Spitz, 1991; Porsani, 1999). By extension, these applications can also be performed using the techniques presented in this chapter.

4.4.2 Conclusion

We presented the extension of $f-x$ prediction filters to the case of quaternion-defined seismic data. The methods consist of the extension of a classical technique, the FXDECON, to the case of quaternion-defined seismic data. The concepts of proper and improper complex/hypercomplex signals, based on the definition of complementary covariance matrices, were revised. It was discussed how the trivial extension of the FXDECON to its hypercomplex version, named QFXDECON, might not take full advantage of the signals if those are improper. In case of improper signals, such as those acquired in multicomponent seismology, additional useful information can be exploited by employing widely-linear models in the prediction context (WL-QFXDECON). Solving for the filter coefficients follows the classical Weiner filtering theory, which maintains the simplicity of the FXDECON.

The main differences are the additional parameters required by the QFT to be applied (sideness and eigenaxis), the fact that one would be dealing with quaternion (instead of complex-valued) data and filter coefficients, and the additional augmentation step in the widely-linear formulation. Such an augmentation increases sample redundancy of both signal and noise allowing for a better prediction of the coherent portions of the data at the cost of noisier results. We discuss that for an increasing filter length, the performances of FXDECON and QFXDECON approach to that of WL-QFXDECON in terms of prediction gains, and that longer filters could be detrimental to the final section of the widely-linear approach due to the augmentation step. We apply the techniques to 3C datasets, but applications to 4C datasets are also possible employing full quaternions. Extension to 3D datasets is also straightforward. Through synthetic and field data examples, one can argue that the WL-QFXDECON has better performance than QFXDECON and FXDECON when concerning signal preservation, but it allows for higher levels of noise in the filtered section.

CHAPTER 5

Regularization by Denoising for Simultaneous Sources Separation

5.1 Introduction

Chapters 3 and 4 presented two classic and highly used techniques in seismic data processing and their extension to vector-valued seismic data through quaternionic representations. In applied seismology, the related denoising engines FXDECON and SSA have their roots going back to the 1980s and 1990s, respectively, but they persist up to today. Since then, however, the number of denoisers being developed has been overwhelming. Especially with the advent of machine learning and its undeniable success through deep models, new denoisers are expected to be developed by the day. Potentially, however, such denoisers become obsolete at the same rate they are developed. That is not to say , however, that research on more efficient and effective denoising engines, as presented in this thesis, should be halted. A better idea is to adapt these denoising engines to other problems, as discussed in this chapter through the Regularization by Denoising (RED) framework (Romano et al., 2017).

Amongst others options to be discussed, RED offers one way to exploit denoising algorithms to solve inverse problems beyond denoising itself. However, it is essential to clarify that the idea of combining inversion and denoising is not necessarily new. Other similar approaches to RED exist, such as the pioneering work of Venkatakrishnan et al. (2013) in the Plug and Play Prior (P^3), as well as other recent approaches that use denoisers within the projected gradient descent (PGD) algorithm (Mahdad et al., 2012; Cheng and Sacchi, 2015). These approaches use the priors imposed by a given denoising engine, such as predictability, low-rankness, or sparsity, in the implicit or explicit regularization of a given inverse problem.

5.1.1 Inverse Problems

The Bayesian approach to inverse problems combines a prior distribution for a model $\mathbf{x}$, with its measured version $\mathbf{y}$, to obtain a posterior conditional probability, $P(\mathbf{x}|\mathbf{y})$, in order to infer $\mathbf{x}$. Assuming the Maximum aposteriori Probability (MAP) model, Bayes' rule implies an optimization problem of the form

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) = \arg\max_{\mathbf{x}} \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})}, \tag{5.1}$$

which can be rewritten through a log transformation as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} -\log\{P(\mathbf{y}|\mathbf{x})\} - \log\{P(\mathbf{x})\}. \tag{5.2}$$

since $P(\mathbf{y})$ is independent on $\mathbf{x}$, and the function $-\log\{\cdot\}$ is monotonically decreasing.

The term $-\log\{P(\mathbf{y}|\mathbf{x})\}$ is known as the log-likelihood term, and it describes the relationship between the desired model parameter $\mathbf{x}$ and its measurements $\mathbf{y}$, under the assumption that $\mathbf{x}$ is known. This term describes the data fidelity, and it is usually written in the form of a loss or cost function $\ell(\mathbf{y}, \mathbf{x}) = -\log\{P(\mathbf{y}|\mathbf{x})\}$. Under the assumption of Gaussian noise $\mathbf{e}$ of unitary variance, a common expression used for the loss function is given as the ℓ_2 norm

$$\ell(\mathbf{y}, \mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2, \tag{5.3}$$

referring to the case where $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{A}$ is a linear operator mapping the solution $\mathbf{x}$ to the observation space $\mathbf{y}$. The case of other noise distributions can be handled through alternative cost functions other than the classic ℓ_2 (Maronna et al., 2019).

The second term in equation 5.2, $-\log\{P(\mathbf{x})\}$, is called the *prior*. It mathematically describes an statistical belief over the unknown $\mathbf{x}$. Such a term is also usually referred to as the regularization term of an inverse problem, and it is highly used to condition the overall inversion to obtain unique and useful solutions. It is also usual to express it as

$$\frac{\mu}{2}\mathcal{R}(\mathbf{x}) = -\log\{P(\mathbf{x})\} \tag{5.4}$$

with the scalar $\mu > 0$, the trade-off parameter, controlling the relative confidence of the regularization term. One can now rewrite the MAP estimation problem in equation 5.1 as the optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ell(\mathbf{y}, \mathbf{x}) + \frac{\mu}{2}\mathcal{R}(\mathbf{x}) \tag{5.5}$$

thus defining a much broader class of inverse problems that will be exploited in this chapter under the applications of simultaneous sources separation (section 5.2). Choosing an appro-

priate prior $\mathcal{R}(\mathbf{x})$ is paramount, as this is the term resposible for driving the most recent breakthroughs in image and signal processing.

5.1.2 Classical Denoising

Noise attenuation (denoising) plays an important role in image and signal processing; it is the process of removing noise from a signal while minimizing the loss of useful content in the signal. From the applied seismology context, two denoising methods have been discussed, the SSA and FXDECON, which recover the underlying harmonics within the observed frequency slices of the seismic data. For completeness, it is helpful to introduce the classical image denoising problem and discuss their solution through approaches termed "variational denoising methods." These methods, building upon the Bayesian approach discussed above, use a suitable prior and seek to minimize an appropriate cost function, for a given observed (noisy) image $\mathbf{y}$. In this case, the linear operator corresponds to an identity matrix, $\mathbf{A} = \mathbf{I}$, in the ℓ_2 norm (equation 5.3) that turns the cost function in equation 5.5 to

$$\hat{\mathbf{x}} = f(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\mathcal{R}(\mathbf{x}). \tag{5.6}$$

Above, $f(\cdot)$ represents the denoising engine associated with the prior $\mathcal{R}(\cdot)$. The remaining task is to select a proper prior for obtaining the best possible solution $\hat{\mathbf{x}}$ under such criteria. Remarkably useful priors include sparsity, low-rankness, and, more recently, deep learning methods.

5.1.3 Plug and Play (PnP)

The PnP prior (or P^3) is the one possible framework that allows the combination of generic inversion routines with state-of-the-art denoising methods. It provides an elegant and straightforward way to introduce denoising engines to solve more general inverse problems. For these reasons, the P^3 method has drawn much attention in many applications.

The PnP method is known for being closely tied to the Alternating Direction Method of Multipliers (ADMM) to solve an inverse problem as equation 5.5

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\mathcal{R}(\mathbf{x}). \tag{5.7}$$

ADMM uses variable splitting and the Lagrange multipliers method to solve smaller problems in an alternating fashion. The method is known to converge for convex misfit and regularization terms; more on its theoretical developments and guarantees can be found in Boyd et al. (2011). Following Venkatakrishnan et al. (2013), one can decouple the data

fidelity and prior terms by splitting $\mathbf{x}$ into two variables $(\mathbf{x}, \mathbf{v})$, and reformulating equation 5.7 as a constrained optimization problem

$$\hat{\mathbf{x}}, \hat{\mathbf{v}} = \arg\min_{\mathbf{x},\mathbf{v}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\mathcal{R}(\mathbf{v}) \quad \text{subject to} \quad \mathbf{x} = \mathbf{v}. \tag{5.8}$$

The solution to this problem can be found through the scaled ADMM, by introducing the Lagrange multiplier $\mathbf{u}$, and completing squares to formulate the cost function

$$\hat{\mathbf{x}} = \arg\min_{(\mathbf{x},\mathbf{v},\mathbf{u})} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\mathcal{R}(\mathbf{v}) + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{v} + \mathbf{u}\|_2^2 - \frac{\gamma}{2}\|\mathbf{u}\|_2^2, \tag{5.9}$$

where γ is a user-provided ADMM parameter. The algorithm consists of repeatedly solving the following sub-problems in an alternating fashion

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\gamma}{2}\|\mathbf{x} - \hat{\mathbf{v}} + \hat{\mathbf{u}}\|_2^2 \tag{5.10}$$

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \frac{\mu}{2}\mathcal{R}(\mathbf{v}) + \frac{\gamma}{2}\|\hat{\mathbf{x}} - \mathbf{v} + \hat{\mathbf{u}}\|_2^2 \tag{5.11}$$

$$\hat{\mathbf{u}} = \mathbf{u} + \hat{\mathbf{x}} - \hat{\mathbf{v}}. \tag{5.12}$$

First, one can solve for $\hat{\mathbf{x}}$ via methods such as conjugate gradients (CG) as this is a damped least-squares problem. The update of the $\mathbf{v}$ variable, however, depends only on the choice of the prior $\mathcal{R}(\cdot)$, similarly to equation 5.6. It can, hence, be interpreted as a denoising step over the update $\hat{\mathbf{x}} + \hat{\mathbf{u}}$, and Venkatakrishnan et al. (2013) suggest replacing the optimization step for obtaining $\hat{\mathbf{v}}$ by the action of a denoiser of choice. Such a step implicitly imposes the previously defined by the denoising engine, as shown below in the PnP-ADMM algorithm.

Algorithm 1 PnP-ADMM

Inputs:$\mathbf{A}$, $\mathbf{A}^T$, $f(\cdot)$, $\mathbf{y}$, μ, γ, $\mathtt{K}$
Initialize $\mathbf{x}_0, \mathbf{v}_0, \mathbf{u}_0$
for $k = 1 : \mathtt{K}$ **do**
  $\mathbf{x}_k = \underset{\mathbf{x}}{\mathtt{arg\ min}} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{v}_{k-1} + \mathbf{u}_{k-1}\|_2^2 \right\}$
  $\mathbf{v}_k = f(\mathbf{x}_k + \mathbf{u}_{k-1})$
  $\mathbf{u}_k = \mathbf{u}_{k-1} + \mathbf{x}_k - \mathbf{v}_k$
end for
Output: $\mathbf{x}_{\mathtt{K}}$

While the PnP scheme is a solid and attractive approach, some comments are in order. First, the method is not accompanied by a clear definition of a cost function since the regularization term used is only implicit, given by the activation of the denoising engine. Second, using arbitrary denoisers within ADMM might induce a non-provable convergence regime for the

method. The parameter tuning of the ADMM algorithm thus becomes nontrivial. Since the PnP is intimately coupled with the ADMM, it does not offer much flexibility in replacing the iterative procedure. These points are discussed in more depth by Romano et al. (2017) when introducing the Regularization by Denoising (RED) framework, which is discussed in the next subsection as a plausible alternative to the PnP approach.

5.1.4 Regularization by Denoising

The RED method (Romano et al., 2017) offers a framework that systematically uses denoising engines in the definition of the regularization term of any inverse problem (e.g., equation 5.24). No other approach suggests a regularization term explicitly employing denoisers in its cost function as RED does. Under the so-called "RED conditions," the RED cost function is convex and optimal solutions can be obtained via first-order optimization methods.

In RED, the regularization term is defined as

$$\mathcal{R}(\mathbf{x}) = \mathbf{x}^T(\mathbf{x} - f(\mathbf{x})), \tag{5.13}$$

where $f(\cdot)$ is a denoising engine of choice. Notice that the denoiser is applied to the candidate solution (in this case, $\mathbf{x}$), and the regularization is proportional to the inner product between this candidate and its denoising error. The overall cost function is therefore written as

$$J(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\mathbf{x}^T(\mathbf{x} - f(\mathbf{x})). \tag{5.14}$$

Minimizing the above cost function has an important interpretation. The first term, unchanged from usual inversion approaches such as equation 5.24, still ensures data fidelity. The second term, however, can be understood as the unnormalized cross-correlation between the solution ($\mathbf{x}$) and its residual ($\mathbf{x} - f(\mathbf{x})$). Hence, having $\mathbf{x}^T(\mathbf{x} - f(\mathbf{x})) \approx 0$ enforces orthogonality between the data and its residuals, a concept highly exploited in the normal equations of least-squares problems. More specifically, small values of $\mathcal{R}(\mathbf{x})$ can either mean that the residual does not contain features from the solution, which means these are uncorrelated, or that $\mathbf{x} \approx f(\mathbf{x})$ and the candidate solution is a fixed-point of the denoiser, as in $\mathbf{x}_{k+1} = f(\mathbf{x}_k)$ (Gezerlis, 2020), and there is nothing else to be filtered out. Either way and also owing to the uncorrelated character of the blending noise in given domains, it is highly desirable to minimize such term.

RED Conditions

For those who deal with denoising and inverse problems daily, the framework proposed by RED seems to offer the possibility of achieving the goal of obtaining denoised solutions to

any inverse problem. In the face of grand promises, RED has been shown to deliver state-of-the-art results in different applications, including image deblurring and super-resolution (Hong et al., 2019; Sun et al., 2020; Mataev et al., 2019), tomography reconstruction (Perelli and Andersen, 2019), and phase-retrieval (Wu et al., 2019). More specifically, in geophysics, its potential has already been recognized in non-linear inverse problems such as traveltime tomography (Vargas, 2020) and full-waveform inversion (Anagaw and Sacchi, 2019).

Although its practical success has been demonstrated in various applications, the formulation for RED as initially proposed by Romano et al. (2017) currently lacks strong theoretical guarantees of convergence. Not surprisingly, more attention has been paid to explaining the undeniable success of such a framework, and other publications have clarified, interpreted, and extended RED even further (Reehorst and Schniter, 2018). The need for such clarification arises from the so-called RED conditions, which consist of specific properties that a denoiser of interest has to fulfill. These properties rely on the differentiability of the denoiser $f(\cdot)$ and are necessary to facilitate RED derivations. Romano et al. (2017) state that the choice of the denoiser has to follow two primary conditions: local homogeneity and strong passivity. Shortly after, Reehorst and Schniter (2018) further clarifies RED by showing that possible denoisers to be used within it must fulfill an additional condition: Jacobian symmetry. Appendix B discusses numerical tests provided by Romano et al. (2017) and Reehorst and Schniter (2018) that ensure a given denoiser satisfies the conditions introduced below. The conditions can be summarized as follows:

- **Local homogeneity** states that the denoiser applied to a scaled version of the input should result in a scaled version of the denoised input. Specifically, $f((1+\epsilon)\mathbf{x}) = (1+\epsilon)f(\mathbf{x})$ for small ϵ.
- **Strong passivity** restricts the denoiser to not magnify the norm of a given input. This means that the Jacobian of the denoising engine is stable, i.e., $\eta(\nabla f(\mathbf{x})) \leq 1$ (where $\eta(\mathbf{M})$ is the spectral radius of a matrix $\mathbf{M}$ and ∇ represents the differentiation operator). A weaker form of this condition means $\|f(\mathbf{x})\| \leq \|\mathbf{x}\|$.
- **Jacobian symmetry** states that the denoiser has a symmetric Jacobian, i.e., $\nabla f(\mathbf{x}) = \nabla f(\mathbf{x})^T$.

One can promptly argue that differentiability is yet a fourth condition to be satisfied by a prospective denoiser. However, as differentiability is the basis of the strong passivity and Jacobian symmetry conditions, one may opt to leave this condition as implicitly defined but definitely not as satisfied. The reason is that, under the assumption that all of RED conditions are satisfied by a given denoiser, the regularization term given in equation 5.13

is differentiable and convex. Thus, one can obtain the gradient of equation 5.14 following

$$\begin{aligned}\nabla J &= \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) + \frac{\mu}{2}\nabla(\mathbf{x}^T(\mathbf{x}-f(\mathbf{x}))) \\ &= \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) + \frac{\mu}{2}\nabla\mathbf{x}^T\mathbf{x} - \frac{\mu}{2}\nabla\mathbf{x}^T f(\mathbf{x}) \\ &= \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) + \mu\mathbf{x} - \frac{\mu}{2}(f(\mathbf{x}) + \nabla f(\mathbf{x})\mathbf{x}).\end{aligned} \tag{5.15}$$

Because the term $\nabla f(\mathbf{x})\mathbf{x}$ can be replaced by $f(\mathbf{x})$ based on the local homogeneity property (Appendix B), one arrives to the following gradient

$$\nabla J = \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) + \mu(\mathbf{x}-f(\mathbf{x})). \tag{5.16}$$

Finding the minimum of equation 5.14 entails setting $\nabla J = 0$ and imposing it as a fixed-point condition on the solution $\mathbf{x}$. Such a condition is the basis for all algorithms that can be used as solvers for RED, but it is highly dependent on the gradient of the regularization term being equal to the denoising residual, i.e., $\nabla\mathcal{R}(\mathbf{x}) = \mathbf{x} - f(\mathbf{x})$.

The need for clarification in RED was first pointed out by Reehorst and Schniter (2018), who thoroughly discussed the condition of Jacobian symmetry and how several classical denoisers, such as median filters and non-local means, actually lack it. Interestingly, even though the above gradient rule is only valid under the assumptions discussed earlier, denoisers that fail to satisfy the Jacobian symmetry condition have yielded excellent results in different applications of RED. It is promptly noticeable that even though RED's theoretical justification is an ongoing scientific development, its practical application is a concrete reality. In that light, Reehorst and Schniter (2018) also introduced the framework of *score-matching by denoising* that explains RED algorithms for generic denoisers that may not satisfy the local homogeneity and Jacobian symmetry properties.

RED Algorithms

Romano et al. (2017) provide options for using RED when solving general inverse problems: steepest descent (RED-SD), fixed-point (RED-FP) iterations, and alternating direction method of multipliers (RED-ADMM). Given its increased attention, several publications have already proposed improvements to the existing solvers and even new solvers for RED (e.g., Reehorst and Schniter (2018); Hong et al. (2019, 2020)). In all cases, the solutions are found iteratively, and together with other simple steps, the denoising engine is applied once or more per iteration, as discussed in this section.

RED-SD

The simplest option, usually the first to be employed, would be the steepest descent. The algorithm follows the usual gradient descent recipe. It iteratively updates the solution using the gradient in equation 5.16 and a given step-size, which might be obtained via backtracking. It is crucial to notice that RED-SD has the same drawbacks as any steepest descent algorithm, and it might take longer to converge than RED-FP and RED-ADMM.

RED-ADMM

As already discussed, ADMM uses variable splitting and the Lagrange multipliers method to solve smaller problems in an alternating fashion, and it is known to converge for convex misfit and regularization terms. For brevity, ADMM is given in Algorithm 2 for arbitrary regularization $\mathcal{R}(\cdot)$ and a user-provided ADMM parameter γ.

Algorithm 2 Alternating Direction Method of Multipliers: minimizes equation 5.24.

Inputs:$\mathbf{A}$, $\mathbf{A}^T$, $\mathcal{R}$, $\mathbf{y}$, μ, γ, $\mathtt{K}$
Initialize $\mathbf{x}_0, \mathbf{v}_0, \mathbf{u}_0$
for $k = 1 : \mathtt{K}$ **do**
$\mathbf{x}_k = \underset{\mathbf{x}}{\mathtt{arg\,min}} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{v}_{k-1} + \mathbf{u}_{k-1}\|_2^2 \right\}$
$\mathbf{v}_k = \underset{\mathbf{v}}{\mathtt{arg\,min}} \left\{ \frac{\mu}{2}\mathcal{R}(\mathbf{v}) + \frac{\gamma}{2}\|\mathbf{v} - \mathbf{x}_k - \mathbf{u}_{k-1}\|_2^2 \right\}$
$\mathbf{u}_k = \mathbf{u}_{k-1} + \mathbf{x}_k - \mathbf{v}_k$
end for
Output: $\mathbf{x}_{\mathtt{K}}$

In Algorithm 2, there are two optimization problems. Again, one can solve for $\mathbf{x}_k$ via many methods as this is a damped least-squares problem. Now, focus on the update of the $\mathbf{v}$ variable and assume the RED framework (equation 5.13). Setting its gradient to zero leads to a fixed-point condition on $\mathbf{v}$ as

$$\mu(\mathbf{v}_k - f(\mathbf{v}_k)) + \gamma(\mathbf{v}_k - \mathbf{x}_k - \mathbf{u}_{k-1}) = 0, \tag{5.17}$$

and the following refinement rule can be used to obtain an approximation to $\mathbf{v}_k$

$$\mathbf{z}_i = \frac{\mu}{\gamma + \mu} f\left(\mathbf{z}_{i-1}\right) + \frac{\gamma}{\gamma + \mu}\left(\mathbf{x}_k + \mathbf{u}_{k-1}\right), \tag{5.18}$$

where $\mathbf{z}_0 = \mathbf{v}_{k-1}$, and $i = [1, I]$ an inner iteration variable. It becomes visible that RED-ADMM allows one to estimate $\mathbf{v}_k$ by activating the denoising engine I times, which stands as a particularity of the RED-ADMM solver. However, since denoisers tend to be computationally expensive, iterating many times is impractical and perhaps even redundant

(Reehorst and Schniter, 2018). Finally, if one uses the update rule given in equation 5.18 to estimate $\mathbf{v}_k$, Algorithm 2 corresponds to what was referred as RED-ADMM.

RED-FP

Similar to the argument above, another possibility exists for solving RED using a fixed-point strategy on the gradient given in equation 5.16 to yield

$$\mathbf{A}^T(\mathbf{A}\mathbf{x}_k - \mathbf{y}) + \mu\left(\mathbf{x}_k - f(\mathbf{x}_{k-1})\right) = 0, \tag{5.19}$$

which can be handled by linear solvers if written as

$$\left(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I}\right)\mathbf{x}_k = \left(\mathbf{A}^T\mathbf{y} + \mu f(\mathbf{x}_{k-1})\right), \tag{5.20}$$

with $\mathbf{I}$ an identity matrix of appropriate dimensions. Unlike RED-ADMM, equations 5.19 and 5.20 above only need one activation of the denoiser, which is followed by a step also closely related to the solution of a damped least-squares problem (Algorithm 3).

Algorithm 3 Fixed-point iteration strategy for RED: minimizes equation 5.14.

Inputs:$\mathbf{A}$, $\mathbf{A}^T$, f, $\mathbf{y}$, μ, $\mathtt{K}$
Initialize: $\mathbf{x}_0$, $\mathbf{G} = \left(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I}\right)$
for $k = 1 : \mathtt{K}$ **do**
  $\tilde{\mathbf{x}}_k = f\left(\mathbf{x}_{k-1}\right)$
  `Define` $\mathbf{b} = \left(\mathbf{A}^T\mathbf{y} + \mu\tilde{\mathbf{x}}_k\right)$
  `Solve` $\mathbf{G}\mathbf{x}_k = \mathbf{b}$
end for
Output: $\mathbf{x}_{\mathtt{K}}$

The current section is an introduction to RED, based on both Romano et al. (2017) and its clarification proposed by Reehorst and Schniter (2018). Both publications still carry significant levels of detail that do not fit the purpose of the present work, which now follows with RED-ADMM and RED-FP analysis for seismic data processing applications such as simultaneous source separation and reconstruction. Although the PnP prior has already seen recent applications in seismic data processing (Xu et al., 2022), the benchmarking presented in the next section uses more established methods within the seismic processing community, such as the Projected Gradient Descent (PGD) algorithm. The PGD represents yet another way to introduce classic denoisers in iterative inversion procedures, and it has been used for deblending (Cheng and Sacchi, 2015) and data reconstruction (Carozzi and Sacchi, 2021).

5.2 Simultaneous Source Separation

Seismic data acquisition has seen fundamental changes to maximize its cost-effectiveness and efficiency. To date, significant improvements have been published in the light of blended acquisition strategies in which several sources are fired at short random time intervals (Beasley et al., 1998; Berkhout, 2008). In addition to its economic benefits, simultaneous-source (blended) surveys allow for denser source sampling, which leads to higher quality subsurface images. That, however, assumes one can suppress interference caused by overlapping shots. Indeed, the advent of high-quality source separation (or deblending) techniques has been a critical factor in implementing blended acquisition practices in the industry.

Given recorded blended data, the main goal of deblending algorithms is to obtain shot records as if they were acquired via conventional acquisition. Blended data are equivalent to time-shifting the data generated from separated sources, followed by a summation over the source index. Let $d(t, s)$ represent a common-receiver/channel gather (CRG) of a conventional seismic experiment, with $t = [1, N_t]$ and $s = [1, N_s]$ representing time and source numbers, respectively. Assuming a blended towed-streamer marine acquisition, the response recorded by a receiver can be written as

$$d_b(t) = \sum_{j \in S} d(t - \tau_j, s_j), \tag{5.21}$$

where $d_b(t)$ is the blended data, τ_j represents the random time delay for the $j-$th source, and S represents all sources fired in a blended gather. Equation 5.21 can be compactly represented as

$$\mathbf{d}_b = \mathbf{B}\mathbf{d}, \tag{5.22}$$

with $\mathbf{B}$ a matrix of size $N_{tb} \times N_t \cdot N_s$ representing the blending operator, $\mathbf{d}$ is a vector of length $N_t \cdot N_s \times 1$ representing the desired CRG, and $\mathbf{d}_b$ is a vector of length $N_{tb} \times 1$ that represents the blended CRG. For future reference, the ratio $\beta = \frac{N_t \cdot N_s}{N_{tb}}$ defines the blending factor. It is understood that these represent the forward operator previously presented as $\mathbf{A}$, the observed data $\mathbf{y}$ is given by $\mathbf{d}_b$, and the solution of interest $\mathbf{x}$ is represented by $\mathbf{d}$.

The simplest deblending process entails extracting the contributions from individual sources in a blended record. It is usually named pseudo-deblending (Berkhout, 2008), or combing (Abma and Foster, 2020), since there is no attempt to reduce interference. One can write

$$\tilde{\mathbf{d}} = \mathbf{B}^T \mathbf{d}_b, \tag{5.23}$$

where $\tilde{\mathbf{d}}$ is the combed data having the same dimensions as $\mathbf{d}$ (i.e., $N_t \cdot N_s \times 1$), $\mathbf{B}^T$ represents the pseudo-deblending operator, and the superscript "T" denotes matrix transposition. In

practice, forward and adjoint blending operators are implemented in implicit form without creating matrices. In other words, $\mathbf{B}$ and $\mathbf{B}^T$ are linear operators designed to pass the dot product test (Claerbout, 1992) so that they behave as matrix-vector multiplication. In addition, one should notice that the pseudo-deblending operation does not remove the interference resulting from the simultaneous sources, i.e., the combed data still contains the so-called blending noise. Different approaches can be taken to better achieve the deblending goal by tackling this interference. These can be divided into two categories: deblending by denoising and deblending by inversion.

Deblending via denoising approaches operates directly on $\tilde{\mathbf{d}}$, usually relying on the spatial incoherency of the blending noise in different domains such as common-receiver or common-offset gathers. In this case, one will likely run into variant formulations of filtering techniques, such as the classical frequency-space prediction error filters (Spitz et al., 2008; Peng et al., 2013) and low-rank filtering (Maraschini et al., 2012, 2016; To et al., 2018; Jeong et al., 2020). Variants of median filters have also shown remarkable success in removing blending interference (Huo et al., 2012). Finally, different formulations of the sparse Radon transform also have notable efficacy in simultaneous source separation (Akerberg et al., 2008; Ibrahim and Sacchi, 2013). These examples represent a limited set close to the options available in the literature, and this fuels the argument that powerful denoisers are being developed each day, having great performances in their applications and conceivably also applicable for simultaneous-source data processing.

Although delivering satisfactory results, deblending by denoising has limitations related to signal distortions in the pre-stack data volume (Abma et al., 2015). Thus, aiming for high quality pre- and post-stack results, deblending by inversion approaches are introduced. Basically, one can pose deblending as an inverse problem similar to equation 5.5 or 5.7

$$\texttt{minimize } J = \frac{1}{2}\|\mathbf{B}\mathbf{d} - \mathbf{d}_b\|_2^2 + \frac{\mu}{2}\mathcal{R}(\mathbf{d}), \tag{5.24}$$

where $\mathcal{R}(\mathbf{d})$ is a regularization term and μ is a trade-off parameter. Minimizing equation 5.24 means solving for the deblended data ($\mathbf{d}$) that honours the blended records ($\mathbf{d}_b$) while having features imposed by the regularizer $\mathcal{R}(\mathbf{d})$. As in any inverse problem, the choice of an appropriate regularizer depends on the features of $\mathbf{d}$ that one wants to attenuate. Usual choices for the regularization term are those that induce sparse solutions (Lin and Herrmann, 2009; Chen et al., 2014; Abma et al., 2015; Li et al., 2019; Ibrahim and Trad, 2020) or associated to low-rank matrices (Cheng and Sacchi, 2015; Kumar et al., 2015).

Advances in deblending, by both denoising and inversion, are critical to the motivation of the current work. In the following sections, the central idea is to bring these efforts together through RED. In contrast with the existing methods for deblending that already combine inversion and denoising, such as Mahdad et al. (2011) and Doulgeris et al. (2012), the novelty

of such a framework derives from in its cost function through which it is possible to define a deblending by inversion approach that explicitly employs denoising engines in the definition of the regularization term; one can refer to such a method as REDeblending.

State-of-the-art: Sparse Inversion

The relation between deblending and compressive sensing can be quickly drawn since a blended record is much more compressed than its corresponding unblended or deblended record. The state-of-the-art approach for separating sources is that of sparse inversion. Having the above in mind, a short introduction to deblending via sparse inversion is in order. The main publication to follow is Abma et al. (2015), which discusses simultaneous sourcing in depth. Song et al. (2019) also provide a comprehensive example of sparse inversion in deblending. The idea is to induce sparsity in an auxiliary domain by means of the ℓ_0 pseudo-norm ($\|\cdot\|_0$). This amounts to defining $\mathcal{R}(\mathbf{d}) = \|\mathcal{F}\mathbf{d}\|_0$ in equation 5.24 if one chooses to induce sparse solutions in the frequency-wavenumber domain, and minimize it via an iterative hard-thresholding algorithm (Blumensath and Davies, 2008)

$$\mathbf{d}_k = \mathcal{F}^H H_\mu \left\{ \mathcal{F} \left(\mathbf{d}_{k-1} - \lambda \left[\mathbf{B}^T \left(\mathbf{B}\mathbf{d}_{k-1} - \mathbf{d}_b \right) \right] \right) \right\}, \tag{5.25}$$

where λ is a step-size to downscale the gradient, $\mathcal{F}$ and $\mathcal{F}^H$ represent the multidimensional forward and inverse Fourier transform pair, and H_μ is the element-wise hard thresholding operator (the proximity operator of the ℓ_0 pseudo-norm)

$$H_\mu\{y_i\} = \begin{cases} 0, & \text{if } |y_i| \leq \mu^{0.5} \\ y_i, & \text{if } |y_i| > \mu^{0.5} \end{cases}. \tag{5.26}$$

The level of desired sparseness is controlled by μ, and it is usually set as a decaying function of the iteration number and maximum input amplitude (Song et al., 2019). Moreover, the Fourier transform is one possible option to induce sparsity in an auxiliary domain, and several others are applicable, such as curvelet (Wason et al., 2011), seislet (Chen et al., 2014), or Radon (Lin and Sacchi, 2020) transforms.

From a more general and preferred perspective, one can define the frequency-wavenumber thresholding (FKT) operator $P(\cdot) = \mathcal{F}^H H_\mu\{\mathcal{F}(\cdot)\}$ as a spatial coherency constraint being applied to the data. Such is the approach presented by Mahdad et al. (2011) and Doulgeris et al. (2012), which allows for more flexibility in how one imposes the spatial coherency constraint. Several denoisers can be considered projection operators to enforce coherency, such as the transform-domain thresholding denoisers mentioned above or reduced-rank filtering with the singular spectrum analysis (SSA) filter (Cheng and Sacchi, 2015). It is also usual

for projection operators to be patch-based denoisers, meaning that they act on overlapping patches of data and reconstruct the output by averaging the denoised patches. Algorithm 4 describes a projected gradient descent (PGD) algorithm with a projection operator $P(\cdot)$ to be a denoiser such as the frequency-wavenumber thresholding operator of Abma et al. (2015), used to benchmark the results obtained with RED.

Algorithm 4 Projected Gradient Descent

Inputs:$\mathbf{B}$, $\mathbf{B}^T$, P, $\mathbf{d}_b$, λ, $\mathtt{K}$
Initialize $\mathbf{d}_0$
for $k = 1 : \mathtt{K}$ **do**
  $\tilde{\mathbf{d}}_k = \mathbf{d}_{k-1} - \lambda \left[\mathbf{B}^T \left(\mathbf{B}\mathbf{d}_{k-1} - \mathbf{d}_b\right)\right]$
  $\mathbf{d}_k = P\left(\tilde{\mathbf{d}}_k\right)$
end for
Output: $\mathbf{d}_{\mathtt{K}}$

On the Denoisers

One should remember that both frameworks studied in this chapter, RED and PGD, require the definition of either a denoiser or a projection operator. Since both are used for noise removal, the term "*denoiser*" will be used to refer to either operation, i.e. $f(\cdot) \equiv P(\cdot)$. A critical point for RED is the denoising engine of choice. There exist several options that one might choose from, and this is a crucial argument favoring RED. However, it might be redundant to deploy several denoising engines at once here, so two denoisers, the frequency-wavenumber thresholding (FKT) and singular spectrum analysis (SSA), should suffice for supporting the presented claims. These denoisers are well-known and commonly used in seismic data processing applications. For further reference, specifically on the topic of simultaneous sources separation, the interested readers are referred to Abma et al. (2015), and Cheng and Sacchi (2015) for detailed descriptions of the FKT and SSA, respectively.

Turning to RED conditions, Romano et al. (2017) provides numerical tests to ensure a given denoiser satisfies the local homogeneity and strong passivity conditions, while Reehorst and Schniter (2018) discuss how to investigate its Jacobian symmetry. In these tests, discussed in Appendix B, the FKT operator shows insufficient local homogeneity only for large threshold values (i.e., more restrictive thresholding), and it passes the numerical tests for the strong passivity and Jacobian symmetry conditions. When testing RED conditions, the SSA satisfies the local homogeneity and strong passivity tests but lacks Jacobian symmetry. However, as one should keep in mind, even if denoisers have insufficient local homogeneity and Jacobian symmetry, RED solvers still deliver high-quality results arguably explained under the score-matching by denoising framework (Reehorst and Schniter, 2018).

The denoisers have to be appropriately set to be applied in each iteration of PGD or RED because their optimality is expected to rely upon how the denoiser has been designed. However, although mentioned in the following sections, this work is not discussing the best setup for any denoiser since it mainly focuses on exploring RED as a suitable framework for solving geophysical inverse problems.

5.2.1 Synthetic 3D VSP Example

In this example, the 3D-3C synthetic VSP dataset (O'Brien, 2010)[1], was used to model simultaneous sources for single and multicomponent receivers numerically. The data was originally modeled as a 9C dataset, but only the vertical sources are considered here. In this case, 31 receivers are placed down a well from 1350.0 to 1850.0 meters in depth. At the surface, 42025 sources are split into 205 source lines with 205 sources each. A Ricker wavelet of central frequency 15.4 Hz was used to model about 2.5 seconds of data with time sampling $\Delta t = 0.012$ s, whereas both source and receivers spacing are set to 16.67 m. The final data is obtained as a regular tensor $(Nt \times Ns_x \times Ns_y \times Nr) = (218, 205, 205, 31)$.

The sources were blended following equation 5.21, where two different source vessels were assumed to perform a blended acquisition with blending factor $\beta = 2$. In this example, the second vessel shoots randomly with respect to the first as to cover its half of the acquisition grid simultaneously. Figure 5.1a shows a time slice at 1.5 seconds of a CRG at depth 1550 m, and 5.1b its corresponding pseudo-deblended (equation 5.23) time slice. Figure 5.2a shows the recordings for a conventional source, and 5.2b its pseudo-deblended counterpart.

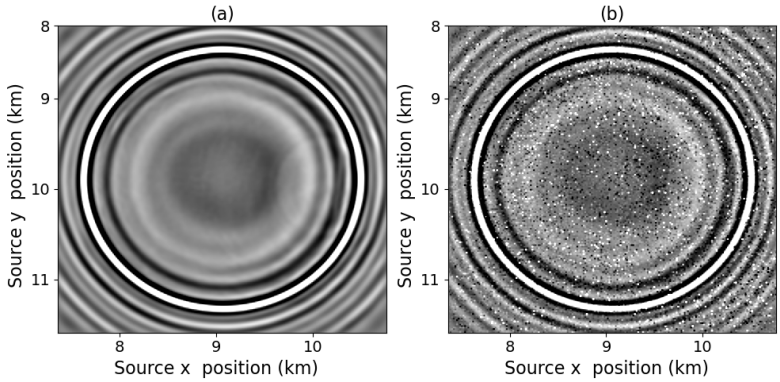

Figure 5.1: Time slice at 1.5 seconds for the (a) ideal and (b) pseudo-deblended ($\beta = 2$) vertical component of a CRG at 1550.0 m.

[1] Avaiable at https://wiki.seg.org/wiki/3D_9C_synthetic_VSP_data.

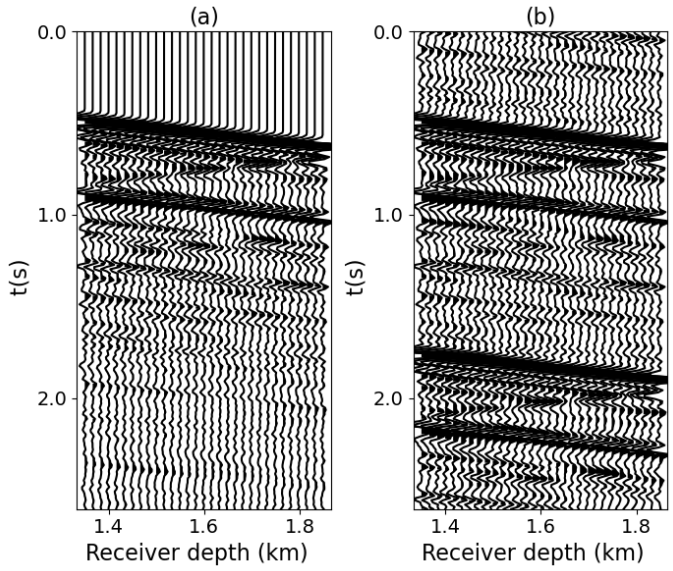

Figure 5.2: Common-shot-gather for the (a) ideal and (b) pseudo-deblended ($\beta = 2$) vertical component of the data.

Different first-order algorithms can be designed to minimize RED's cost function. However, using RED-SD as a solver for deblending is detrimental since the gradient in equation 5.16 has a noise-add-back term corresponding to the regularization gradient. Specifically, the noise term refers to blending noise that would be mapped back to the solution space, which leads to cross-talk since this interference fits the blended data (To et al., 2018). Therefore, although a noise-add-back term might be useful in other applications, RED-SD is likely to not perform well in deblending, thus making RED-ADMM and RED-FP more attractive solvers. Each algorithm will require different parameter settings for each application, such as selecting the trade-off parameter μ for RED or the step-size λ for the PGD. Moreover, although these parameters are important in any inversion, their estimation might be cumbersome. The tests presented here favored a heuristic approach for its selection, such as constraining step sizes and trade-off parameters according to the blending factor. Both the FKT and SSA are coupled with the PGD and RED, and each algorithm runs for $K = 100$ iterations. A convergence criteria of $\left|\frac{J(\mathbf{d}_{k-1})-J(\mathbf{d}_k)}{J(\mathbf{d}_k)}\right| < 10^{-6}$ was not met in any case. The prediction gain is computed at each iteration k as

$$R(k) = 10\log_{10}\left(\frac{\|\mathbf{d}^o\|_F^2}{\|\hat{\mathbf{d}}_k - \mathbf{d}^o\|_F^2}\right), \tag{5.27}$$

where $\mathbf{d}^o$ and $\hat{\mathbf{d}}$ are the ideal (noise-free) and denoised data, respectively.

Scalar Case: FKT Denoiser

The FKT was applied to overlapping patches of size $128 \times 32 \times 32$ samples, having overlaps of $(10, 20, 20)\%$ along each dimension. In addition, an exponential schedule is adopted to reduce, at each iteration, the amplitude thresholding from 99.9% to 0.01% of the maximum Fourier coefficient across the patched data. The FKT setup was held constant across all tested methods, and it allows recovering finer details of interest in the final section. Figure 5.3a shows the results obtained with the PGD and a step-size $\lambda = \frac{1}{\beta} = 0.5$, and 5.3b and 5.3c the results with RED-FP and RED-ADMM, respectively. The regularization and ADMM parameters were set as $\mu = \lambda = 0.2$. While all results are of good quality, RED-ADMM delivered better source separation as indicated by the difference with respect to the ideal data in Figure 5.1a. The RED-FP residuals (Figure 5.3e) show signal leakage not present in the results obtained with the PGD and RED-ADMM (Figures 5.3d and 5.3f). All of these differences have been scaled by 5.0 for visualization.

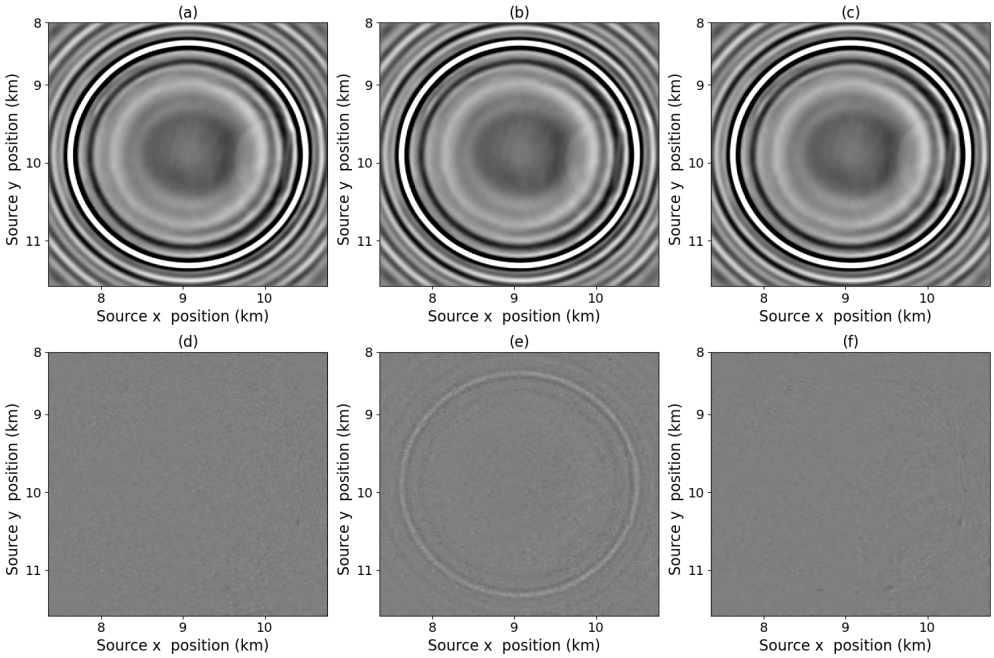

Figure 5.3: Deblended ($\beta = 2$) CRG using the FKT denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their scaled difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively. The difference scale is 5.

Figure 5.4a-5.4c compares one CSG after separating all receivers. Again, RED-FP shows small levels of signal residuals (Figure 5.4e) which are not present in the results obtained with PGD and RED-ADMM (Figures 5.4d-5.4f). The final reconstruction gains obtained with the PGD, RED-FP, and RED-ADMM were, respectively, 84.5, 82.59, and 90.19 dB.

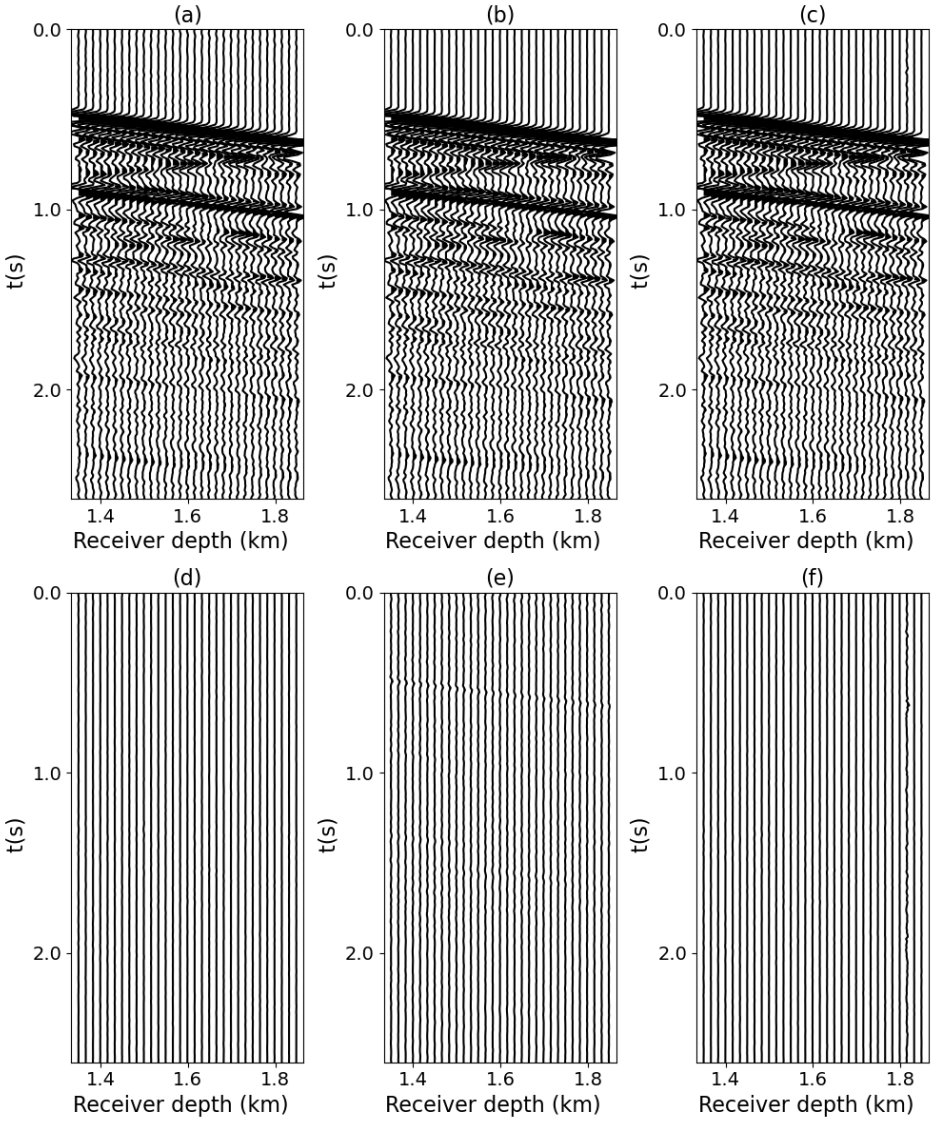

Figure 5.4: Deblended ($\beta = 2$) CSG using the FKT denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively. The difference scale is 5.

Scalar Case: SSA Denoiser

The SSA filter was employed in the same examples as the FKT above, where the patching and inversion parameters were kept the same across the tests. A Lanczos-based SSA filter runs over the frequency range of 0 to 50 Hz in all cases, and the number of Lanczos iterations was set to 5. Notice that one might employ any more sophisticated rank-selection strategy. Examples entail adopting an increasing rank with iteration to recover details beyond the linear moveout assumption or a more restrictive rank for noisier conditions.

The analysis follows with the SSA filter as a denoiser within PGD and RED. The results in Figure 5.5a show good attenuation of blending interference, but it is also noticeable from its difference sections in Figure 5.5d that the method did not recover finer details of interest such as diffractions. Such might be the action of adopting a constant rank for the inversion, especially for later iterations when blending noise is expected to be attenuated (Cheng and Sacchi, 2015). When used as the denoising engine within RED, the SSA also attenuates blending interference with RED-FP and RED-ADMM, obtaining slightly superior results than the PGD algorithm. The final reconstruction gains obtained with the PGD, RED-FP, and RED-ADMM were 47.67, 53.17, and 54.27 dB.

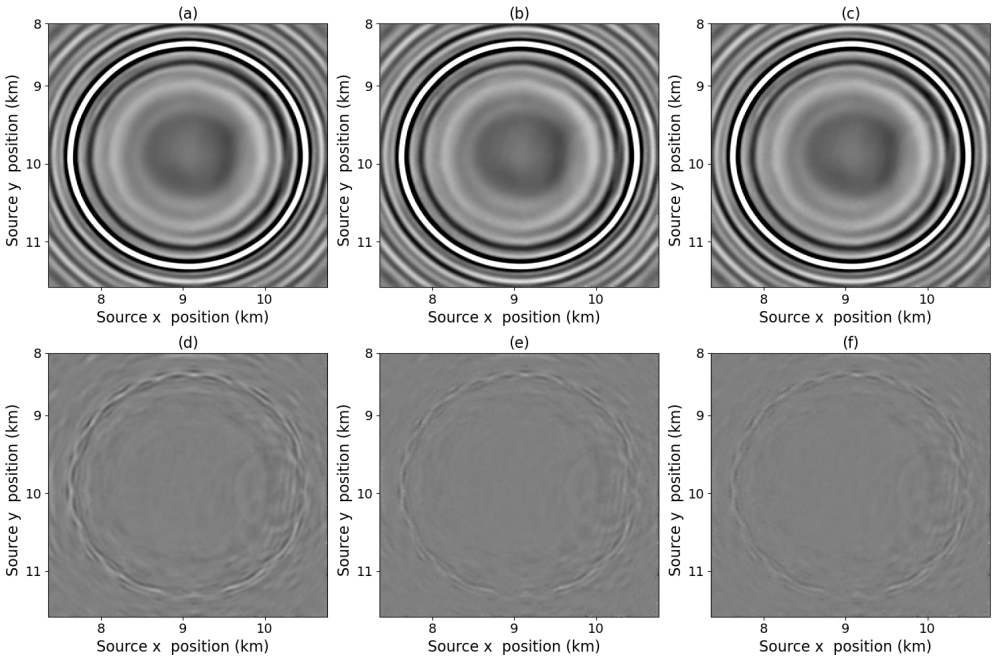

Figure 5.5: Deblended ($\beta = 2$) CRG using the SSA denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively.

Using the SSA filter, the process was repeated for the same 31 blended common-receiver gathers as before. Figures 5.6a-5.6c show the separated common-shot gather obtained by each method. The results show small levels of detail still being retrieved in each separated shot despite having higher levels of signal leakage in comparison with the FKT denoiser, as shown in the difference sections (Figures 5.6d-5.6f).

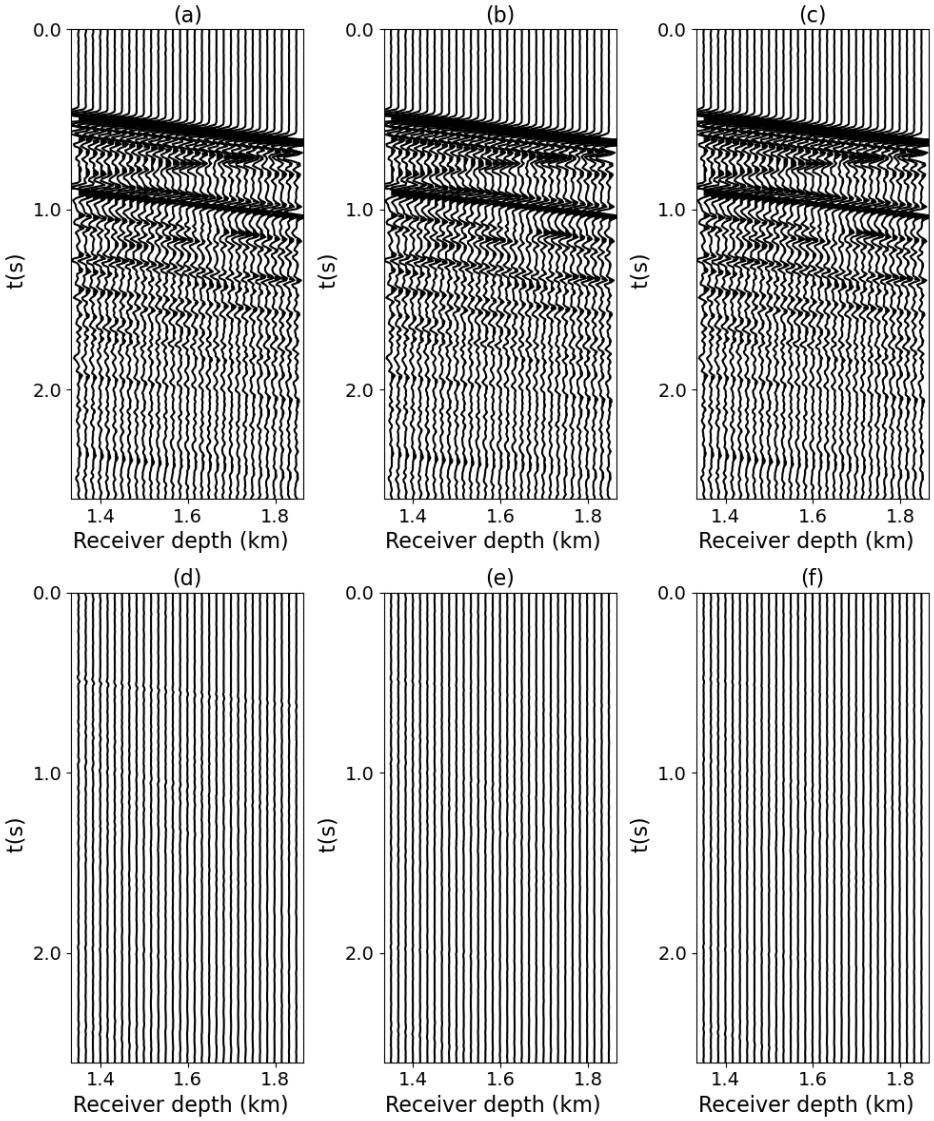

Figure 5.6: Deblended ($\beta = 2$) CSG using the SSA denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively.

It is noticeable, nevertheless, that RED could improve the final reconstruction gains for all cases, as summarized in Table 5.1. To avoid redundancy, the denoiser was activated only once with RED-ADMM ($I = 1$). Further contrasting the denoisers employed in the inversion, the FKT denoiser could better solve for the separated sources than the SSA. It is arguable, in this case, that the SSA denoising setup can be tunned to improve the inverted results further, as the difference section in Figure 5.5 suggests room for improvement.

Scalar case		
$f(\cdot)$	FKT	SSA
PGD	84.50	47.67
RED-FP	82.59	53.17
RED-ADMM ($I = 1$)	89.95	54.27

Table 5.1: Final prediction gain obtained with the FKT and SSA denoisers within the PGD and RDB.

The examples presented so far focused on a componentwise deblending of the vertical component obtained from the 3D-3C synthetic VSP data. The next section of this chapter provides an example of multicomponent simultaneous sources separation using the whole 3C dataset. Arguably, the denoising step in the PGD and RED can exploit the additional information in the correlation between data components through quaternion-based techniques, as presented in chapters 3 and 4. Although a rigorous theoretical attempt to derive a quaternionic RED framework goes beyond the purpose of this thesis, the practical example presented here shows value in such an approach. More important for the context of this thesis is the extension of high-quality deblending techniques to multicomponent datasets, which is of great value because continuous records, for instance, facilitate recording later converted-wave reflections. Since the length of a separated shot record becomes a processing parameter in the deblending routines rather than being a constraint coming from acquisition, obtaining the slower converted and S-wave modes becomes convenient. What motivates the following section on vector-valued deblending through quaternion-based inversion.

Vector Case

Figure 5.7 shows the time slice in 5.1a-5.1c for all displacement components (u_x, u_y, u_z), and its pseudo-deblended counterparts in 5.1d-5.1f. Notice that obtaining a quaternionic pseudo-deblended operator follows from equation 5.21, as each data component can be pseudo-deblended separately. Assuming pure quaternion arrays and with some abuse of

notation, its time domain formulation can be written as

$$\begin{aligned} d_b(t) = & \quad \textstyle\sum_{j\in S} d(t-\tau_j, s_j) \\ = & \quad \textstyle\sum_{j\in S} (u_x i + u_y j + u_z k)(t-\tau_j, s_j), \\ = & \quad \textstyle\sum_{j\in S} u_x(t-\tau_j, s_j)i + u_y(t-\tau_j, s_j)j + u_z(t-\tau_j, s_j)k \end{aligned} \tag{5.28}$$

where $d_b(t)$ now represents the longer multicomponent record, and $d(t,s)$ represent a multicomponent CRG of a conventional (not blended) seismic experiment. Both are now quaternion-valued functions, i.e., $(d_b(t), d(t,s)) \in \mathbb{H}$ with components related to the displacements fields $(u_x(t,s), u_y(t,s), u_z(t,s))$. In the frequency domain, using a left-sided forward quaternion Fourier transform with eigenaxis μ, equation 5.28 is written as

$$D_b(\omega) = \sum_{j\in S} e^{-\mu\omega\tau_j} D^L_{-\mu}(\omega, s_j) \tag{5.29}$$

based on the time-shift property discussed in equation 2.63. As mentioned, obtaining its time-domain counterpart requires consistency in terms of sideness and eigenaxis of the inverse QFT. Both equations 5.28 and 5.29 provide a quaternionic linear forward model for multicomponent blended data that can be compactly represented in matrix-vector form as

$$\mathbf{d}_b = \mathbf{B}\mathbf{d}, \tag{5.30}$$

with $\mathbf{B} \in \mathbb{H}^{N_{tb}\times N_t\cdot N_s}$ representing the quaternionic blending operator, $\mathbf{d}$ is a $\mathbb{H}^{N_t\cdot N_s\times 1}$ vector representing the desired CRG, and $\mathbf{d}_b$ is a $\mathbb{H}^{N_{tb}\times 1}$ vector that represents the blended CRG. Unlike the adjoint operator given in equation 5.23, its quaternionic version is represented by its conjugate transpose

$$\tilde{\mathbf{d}} = \mathbf{B}^H \mathbf{d}_b, \tag{5.31}$$

and their adjointness, for instance, is still given by the dot product test provided the scalar and vector parts of the products are equal within a small tolerance.

Given the hypercomplex nature of the problem, the quaternionic RED cost function would be written as

$$J(\mathbf{d}) = \frac{1}{2}\|\mathbf{B}\mathbf{d} - \mathbf{d}_b\|_2^2 + \frac{\mu}{2}\mathbf{d}^H(\mathbf{d} - f(\mathbf{d})), \tag{5.32}$$

where $f(\cdot)$ represents a quaternion-based denoiser. The data-fitting term is a real function of quaternion varabiales, but RED's regularization is a quaternion function. In other words, as it is written, $J(\mathbf{d})$ represents a quaternion function of quaternion variables $J : \mathbb{H}^{N\times 1} \to \mathbb{H}$. One option is to follow the developments of Flamant et al. (2021), and use a real-valued inner product to define the regularization term to obtain a real function of quaternion variables $J : \mathbb{H}^{N\times 1} \to \mathbb{R}$. The framework of generalized $\mathbb{HR}$ calculus given by Xu et al. (2015)

extends complex calculus ($\mathbb{CR}$) to the case of quaternion variables and offers a sound way to develop and solve quaternionic optimization problems as done by Flamant et al. (2021). These publications bridge the theoretical gap of many quaternion-based applications arising in the literature, including the development of the quaternionic projection-onto-convex-sets already exploited in vector-valued seismic data reconstruction by Stanton and Sacchi (2013). Further developing a quaternion-based RED inherits all of its conceptual difficulties already from the real domain, and the approach followed can be described as finding a fixed-pont to the gradient in equation 5.16.

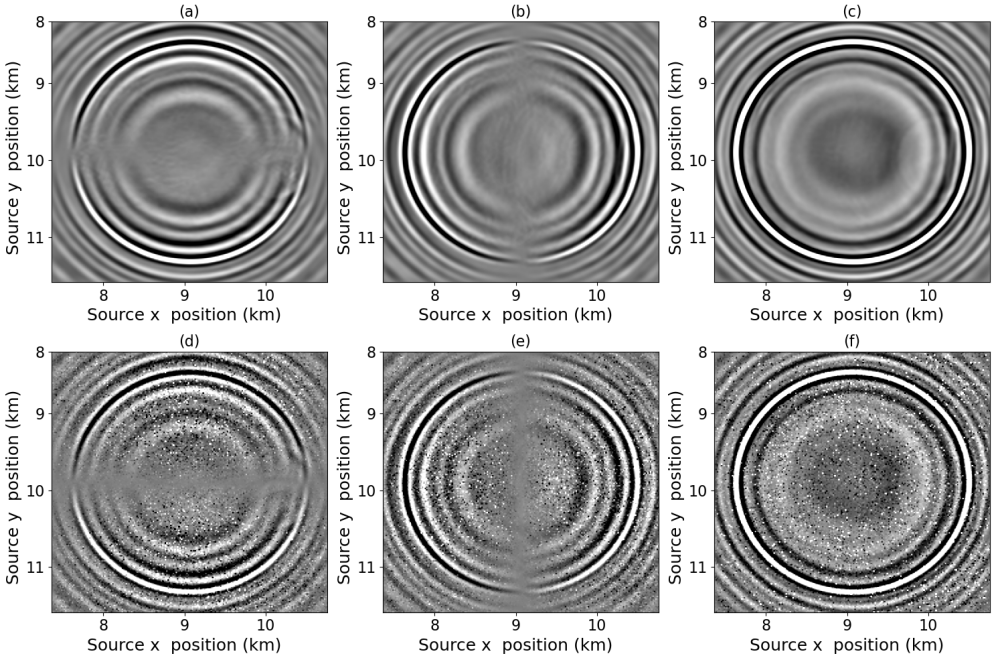

Figure 5.7: Time slice at 1.5 seconds of a multicomponent (a)-(c) ideal and (d)-(f) pseudo-deblended CRG at 1550.0 m.

Vector Case: QFKT Denoiser

The frequency-wavenumber thresholding denoiser has a straightforward extension to the quaternion domain (Stanton and Sacchi, 2013). Its threshold schedule is obtained based on the QFT of the data, here taken from the left with $\mu = i$, and its amplitudes are again set to decrease from 99.9% to 0.01% of the maximum (quaternion) Fourier coefficient. Figure 5.8a-5.8c shows the results obtained with the PGD, and 5.3d-5.8f and 5.8g-5.8i the results with RED-FP and RED-ADMM, respectively, for all data components.

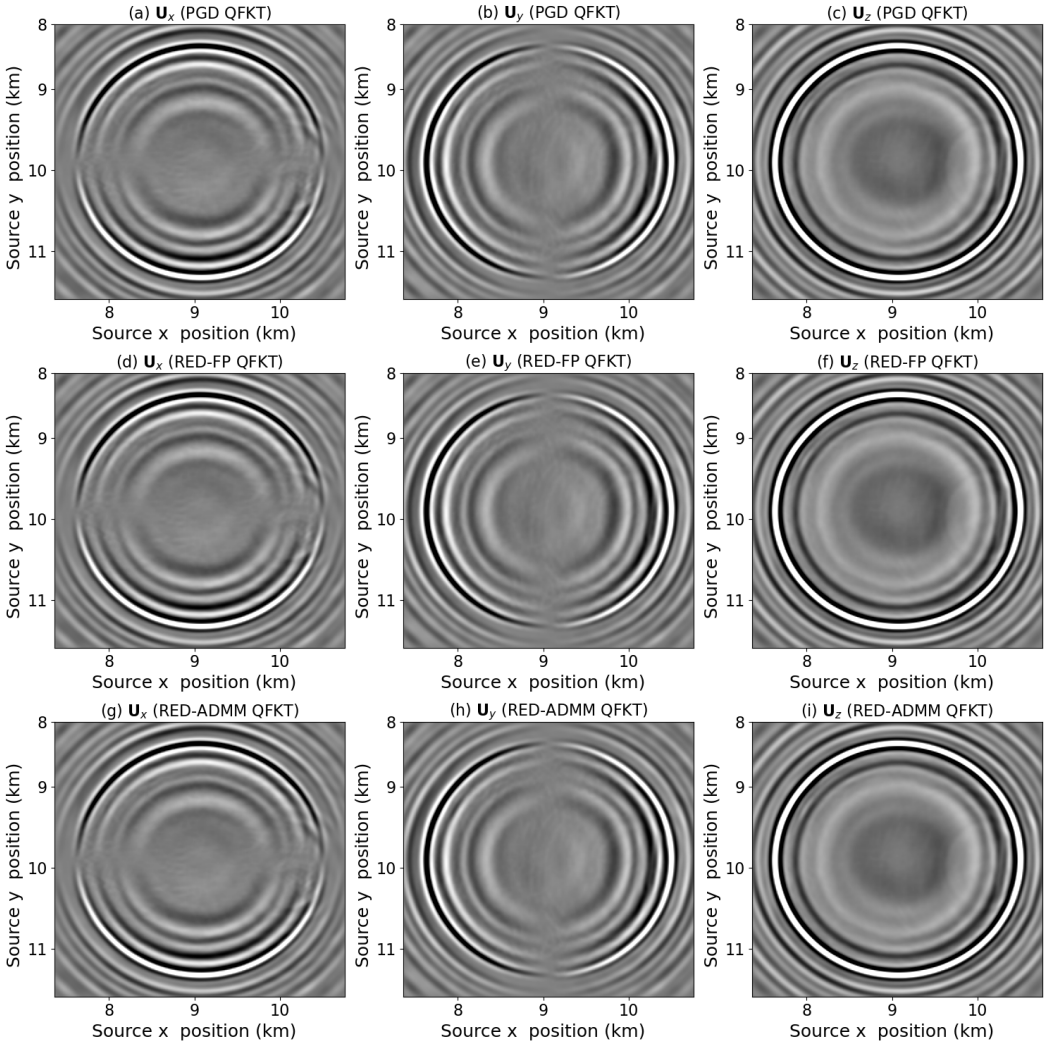

Figure 5.8: Time slice at 1.5 seconds of the deblended components ($\beta = 2$) with (a)-(c) PGD, (d)-(f) RED-FP, and (g)-(i) RED-ADMM using the QFKT denoiser.

While all results are of good quality, RED delivered excellent source separation as indicated by the difference with respect to the ideal data in Figure 5.9. For matters of brevity, Figure 5.10a-5.10c compares the vertical component CSG obtained after separating all receivers using the vector-valued approaches. Again, the PGD method shows small levels of signal residuals (Figure 5.10d) which are not present in the results obtained with RED (Figures 5.10e-5.10f). The final reconstruction gains obtained with the PGD, RED-FP, and RED-ADMM were, respectively, 87.71, 88.99, and 95.83 dB.

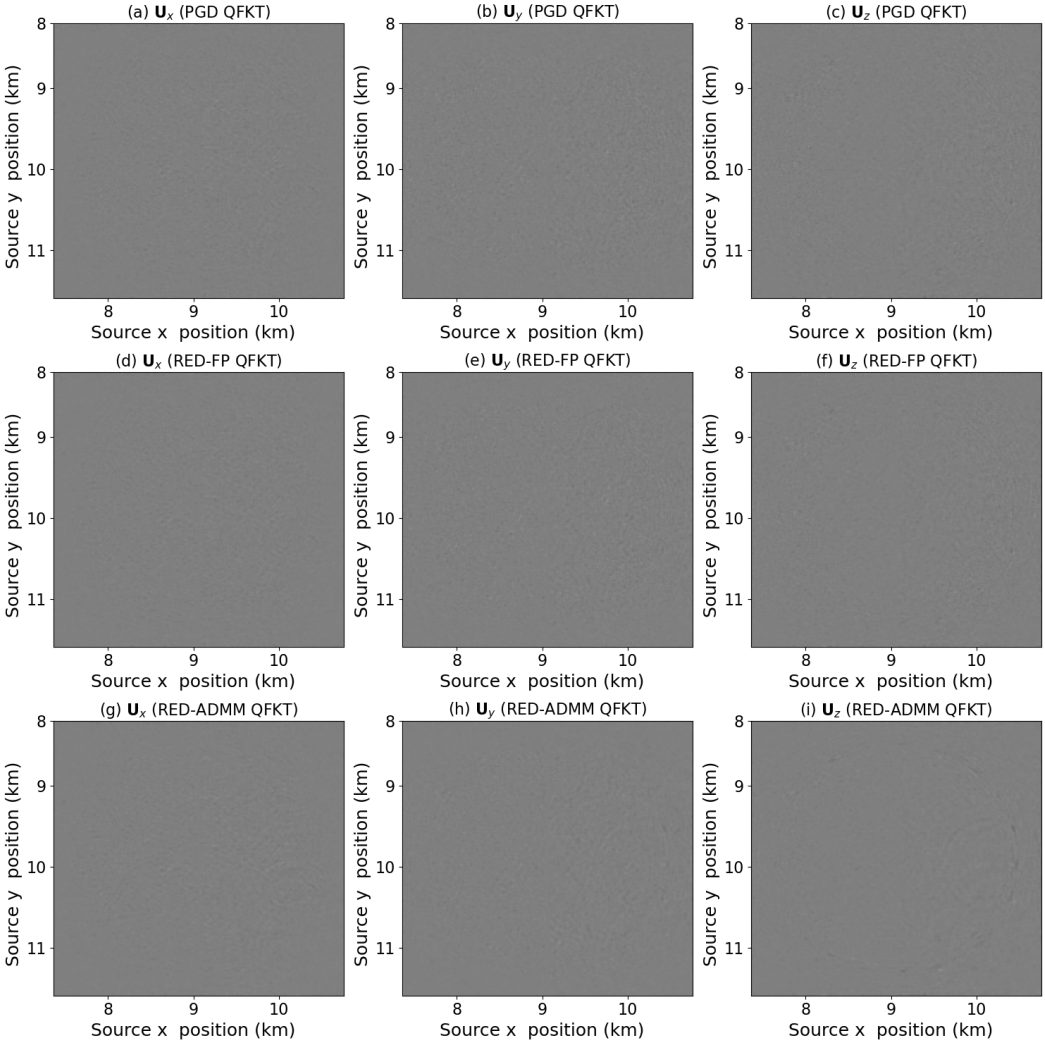

Figure 5.9: Difference time slice at 1.5 seconds of the deblended components with (a)-(c) PGD, (d)-(f) RED-FP, and (g)-(i) RED-ADMM, using the QFKT denoiser, with respect to the clean data in Figure 5.9a-5.9c, respectively.

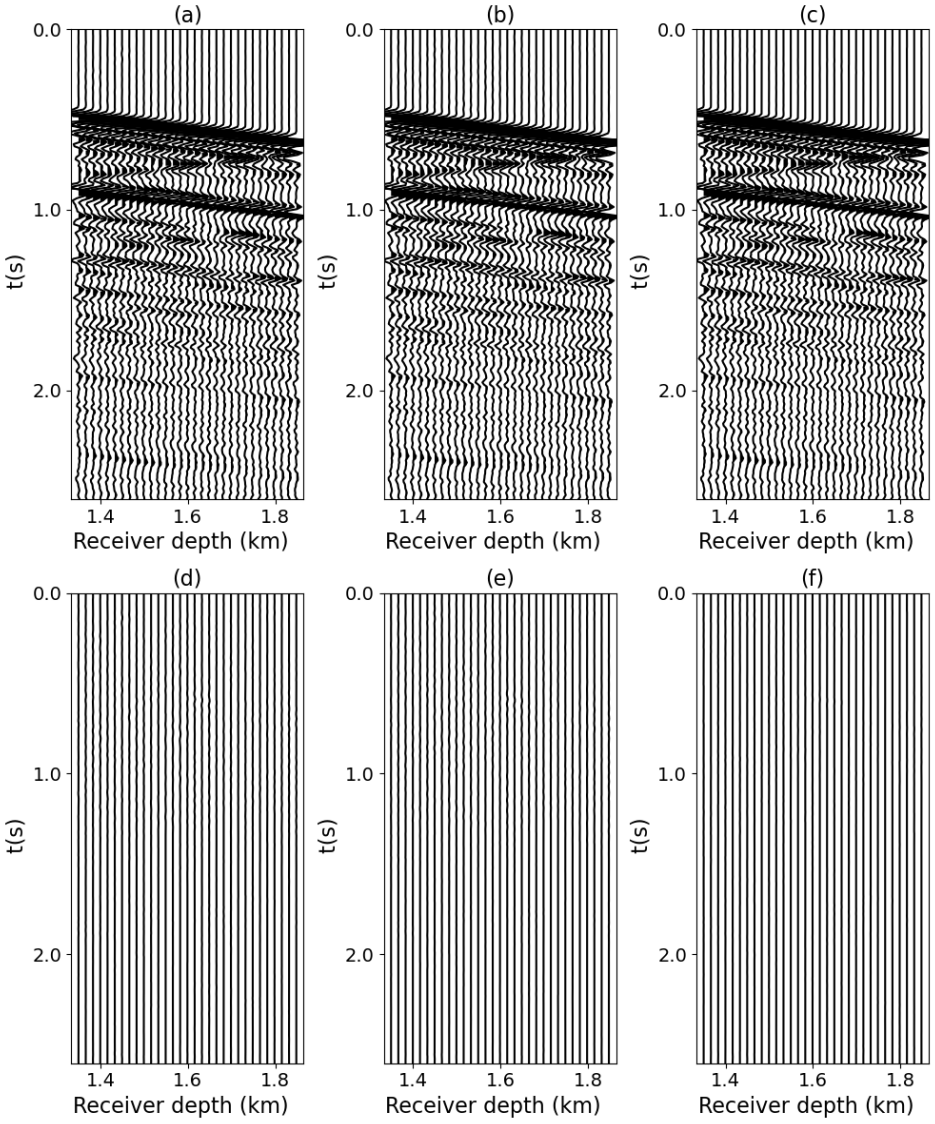

Figure 5.10: The vertical component of the deblended ($\beta = 2$) CSG using the QFKT denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively.

Vector Case: QSSA Denoiser

The tests with the QSSA as a denoiser in PGD and RED followed similarly to the SSA tests, and the same parameters were employed. The QSSA was also based on the Lanczos bidiagonalization, which uses spectral quaternion convolutions to obtain an efficient rank-reduction formulation, as discussed in Appendix A. The number of Lanczos iterations is set to $k = 5$ in the QSSA, and the separated results are shown in Figure 5.11 for all methods.

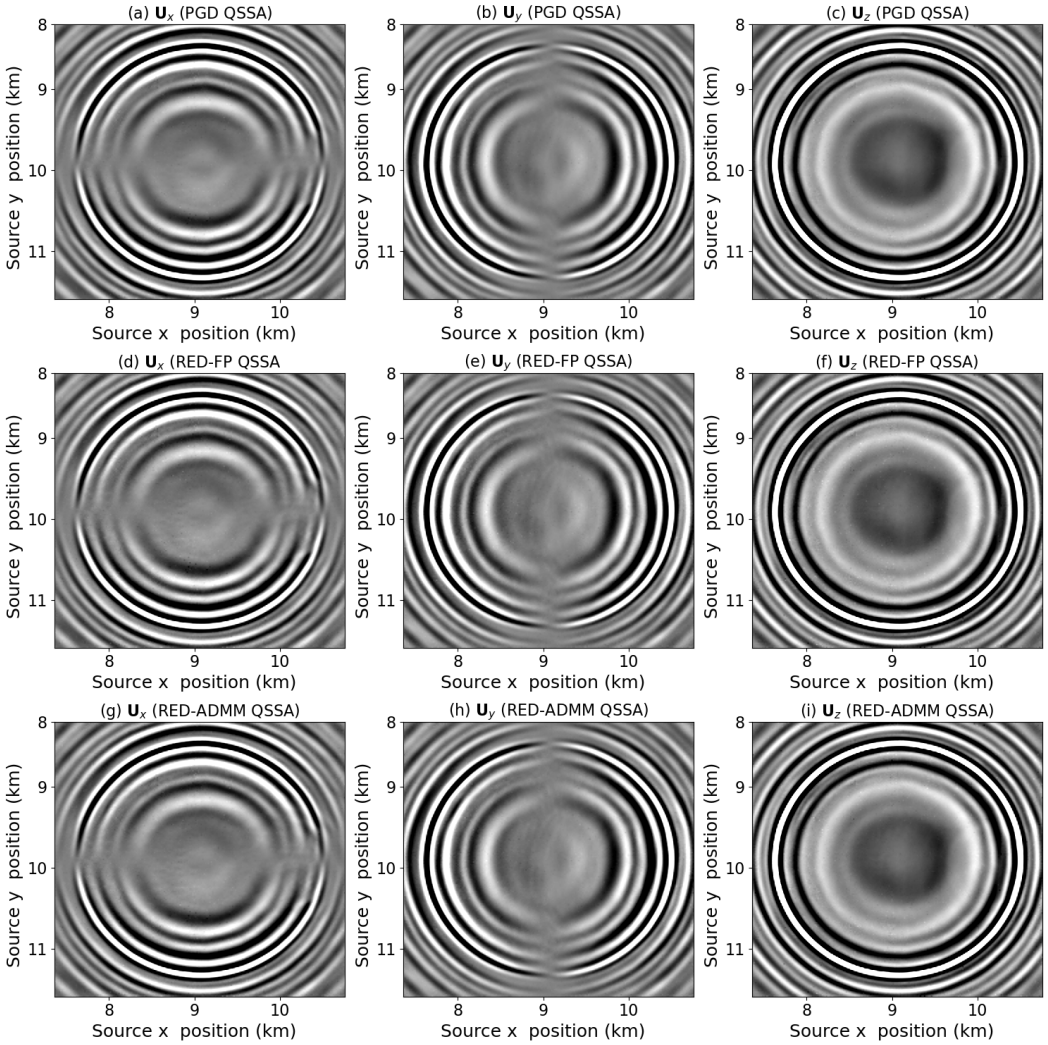

Figure 5.11: Time slice at 1.5 seconds of the deblended components ($\beta = 2$) with (a)-(c) PGD, (d)-(f) RED-FP, and (g)-(i) RED-ADMM using the QSSA denoiser.

Compared to the QFKT, similarly to the scalar case, the QSSA still has room for improvement. All methods, for all components, have higher levels of leaking signals into the residual panels, as shown in Figure 5.12. This is also observable in the common-shot-gathers shown in Figure 5.13, but only focusing on the vertical component for brevity. The final reconstruction quality gains for each method were 51.88, 52.26, and 52.69 dB. Therefore, in the vector case, RED could improve upon the results of the PGD marginally.

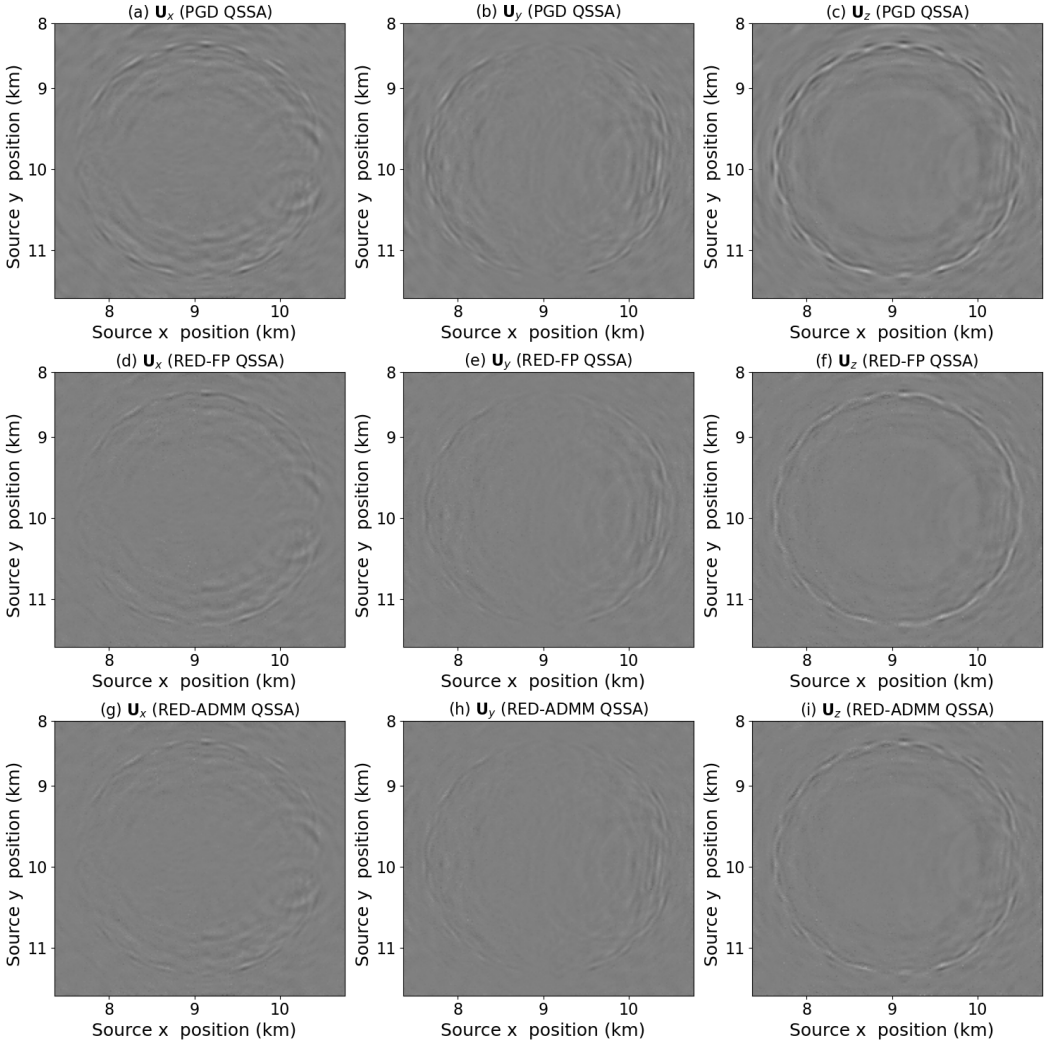

Figure 5.12: Difference time slice at 1.5 seconds of the deblended components with (a)-(c) PGD, (d)-(f) RED-FP, and (g)-(i) RED-ADMM, using the QSSA denoiser, with respect to the clean data in Figure 5.9a-5.9c, respectively.

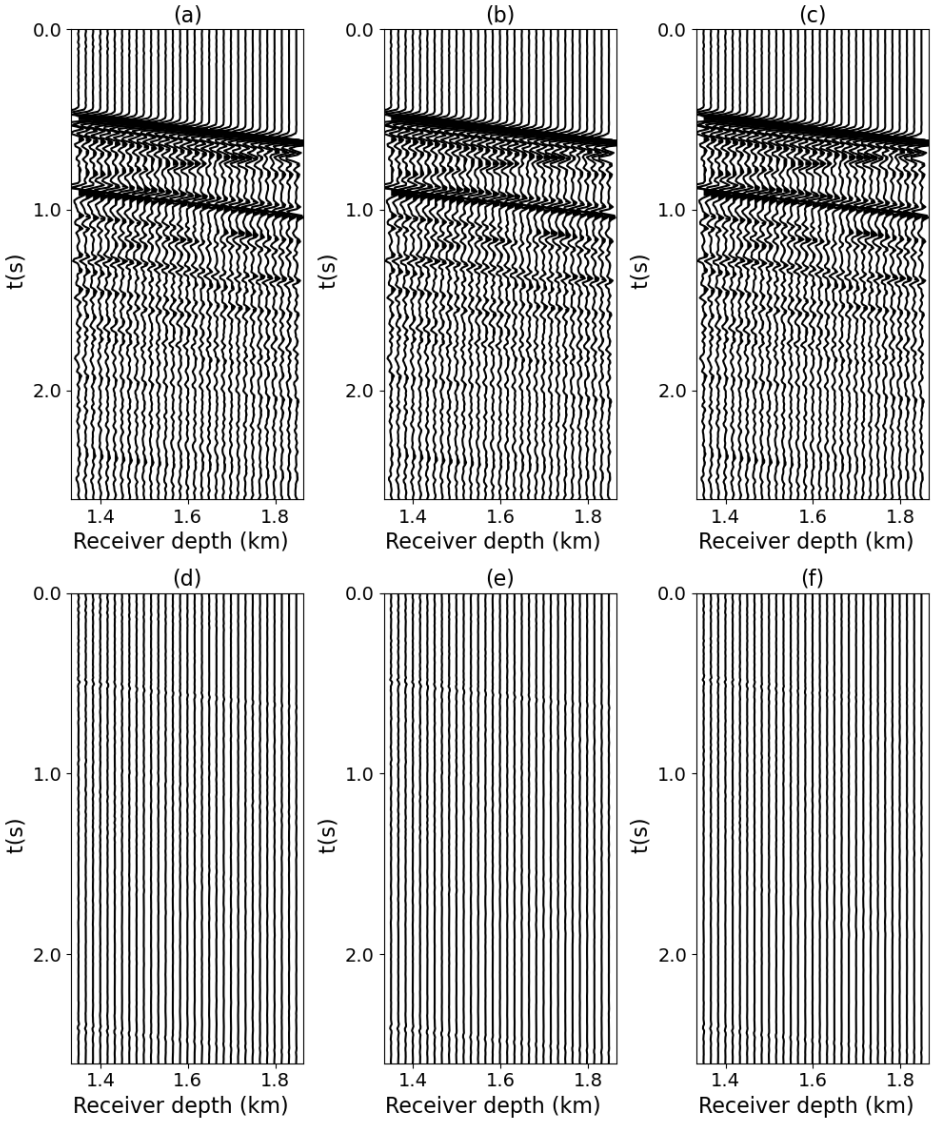

Figure 5.13: The vertical component of the deblended ($\beta = 2$) CSG using the QSSA denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 1$). Their difference with respect to the clean data in Figure 5.2a is shown in panels (d)-(f), respectively.

5.2.2 Numerically Blended Scalar Example

For this numerical experiment, 790 shots of the Mississippi Canyon data were numerically blended, having different blending factors. Each conventionally acquired shot has $N_t = 1751$ time samples with sampling interval $\Delta t = 0.004$ ms, and a total of 7 s of data. Figure 5.14a displays a patch of the desired clean common-receiver gather of the Mississippi Canyon data, while Figure 5.14b-5.14d show the combed data for blending factors 2, 4, and 8 where the increasing level of interference is obvious. These data are deblended using RED-FP and RED-ADMM, and the results are benchmarked against the PGD to determine its suitability.

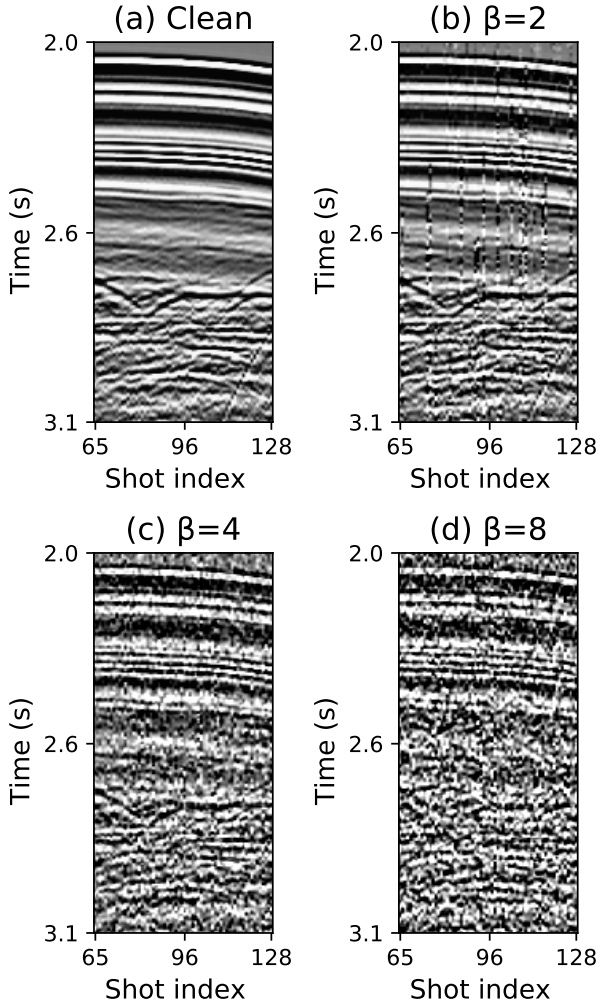

Figure 5.14: A patch of the desired CRG (a), and combed data for blending factors (β) 2 (b), 4 (c), and 8 (d). The increasing level of interference is evident.

Examples: FKT Denoiser

The best results with the FKT denoiser were obtained by applying it on overlapping patches of size 512×64 samples, having overlaps of 50% in both dimensions. Again, an exponential schedule is adopted to reduce, at each iteration, the amplitude thresholding from 99.9% to 0.01% of the maximum Fourier coefficient in the data. Figure 5.15 displays the deblended results of the same patch of data shown in Figure 5.14 for all methods and blending factors. Each row shows the results obtained with a different method, and columns are used to display the same blending factor. Figure 5.15a-5.15c show the results obtained from the PGD algorithm with the FKT denoiser for all blending factors. While it is evident that higher blending factors make it harder to recover the underlying signal of interest in Figure 5.14c, the results show better signal detail than its combed counterparts.

RED provides a valuable alternative to the Fourier sparse inversion approach. Using RED-FP (Figure 5.15d-5.15f), one obtains similar results as compared with those of the PGD method where the FKT denoiser obtained great levels of recovered detail. Particularly for RED-ADMM, the denoiser is allowed I internal activations at the expense of running time. Two sets of tests were performed with RED-ADMM, allowing $I = 1$ and $I = 2$ using the FKT denoiser. As it will become apparent, the second denoising iteration was helpful for the FKT operator for all blending factors. Therefore, for matters of room and brevity, only the results for $I = 2$ are shown in Figure 5.15g-5.15i. The pattern of signal recovery among different blending factors is also consistent with the previous results. Signals highly masked by blending interference, especially due to high blending factors, are still hard to recover, as shown in the difference panels (Figure 5.16). Note that the difference panels for blending factors 2, 4, and 8 have been scaled by 5.0, 3.0, and 1.0, respectively, for better visualization.

Table 5.2 displays the final prediction gain when the FKT denoiser is used in each of the methods above. Here one can easily notice how all methods follow each other closely. Such an observation is key to the claim that RED can achieve state-of-the-art deblending results, and therefore provides an attractive framework to solving other geophysical inverse problems.

FKT			
Blending factor	2	4	8
PGD	19.12	10.47	5.84
RED-FP	19.39	10.61	5.83
RED-ADMM ($I = 1$)	19.45	10.28	5.12
RED-ADMM ($I = 2$)	19.57	10.69	5.66

Table 5.2: Final prediction gain obtained with the FKT denoiser within the PGD and RDB. The table containst results for all blending factors.

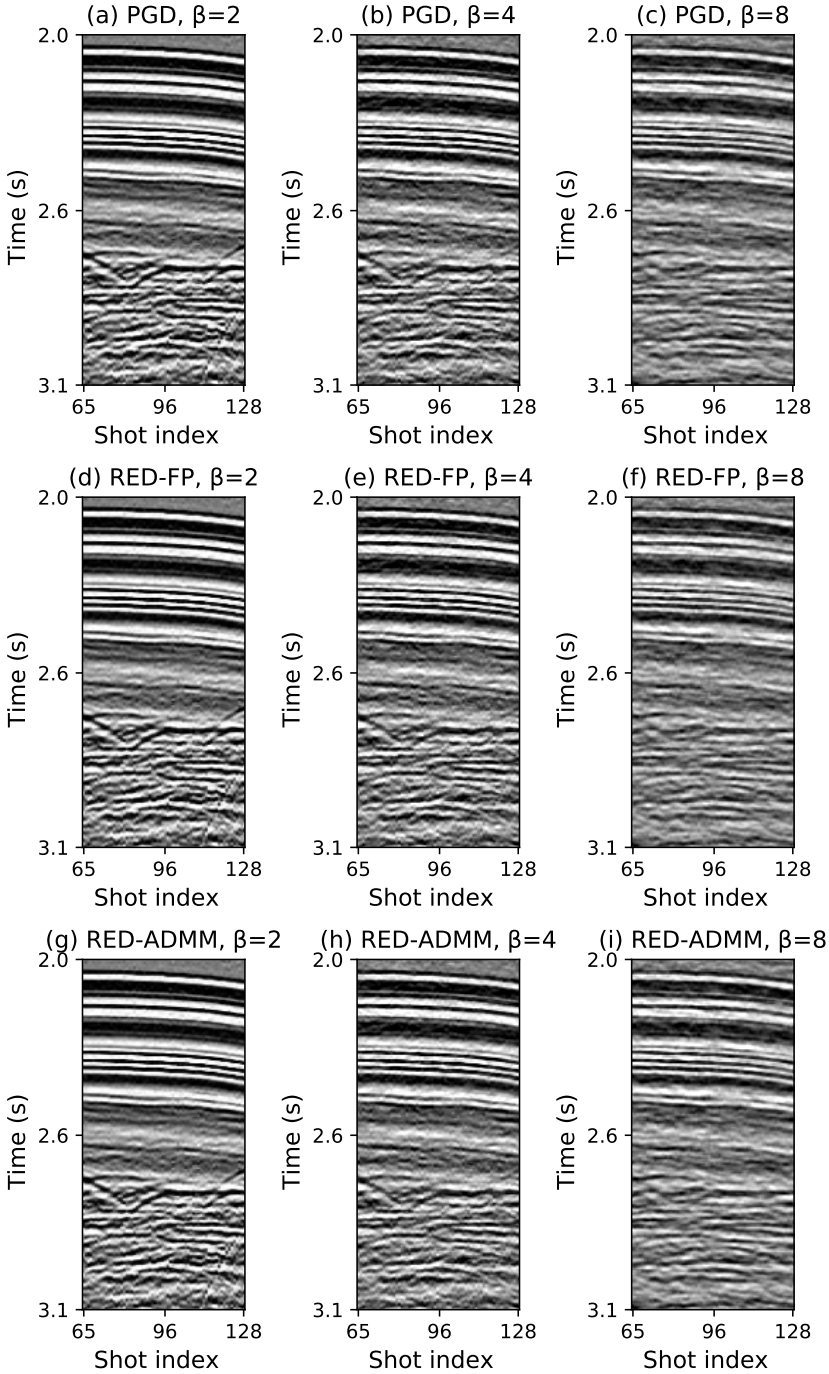

Figure 5.15: Deblending results with the FKT filter for blending factors (β) 2, 4 and 8 using the (a)-(c) PGD, (d)-(f) RED-FP and (g)-(i) RED-ADMM ($I = 2$).

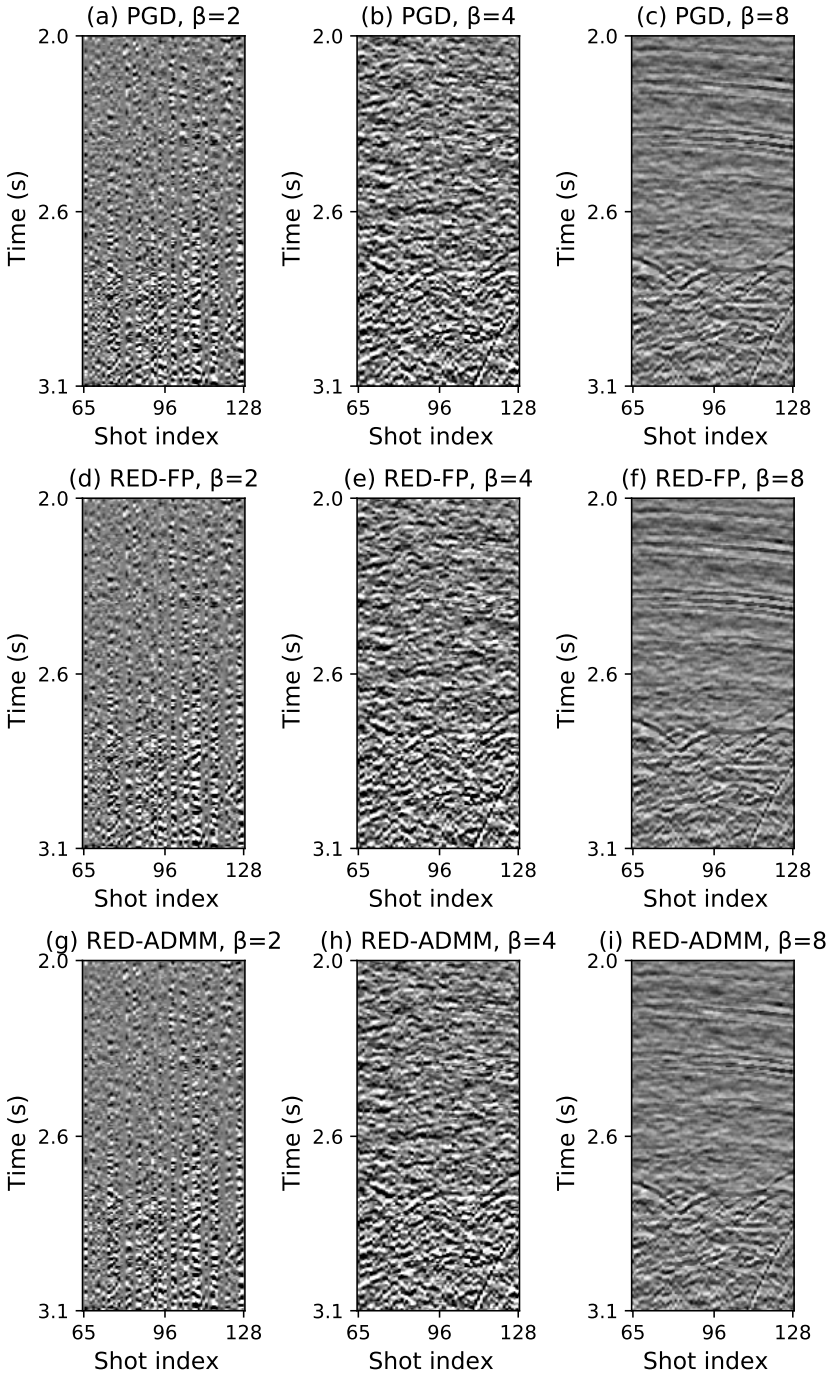

Figure 5.16: Difference between deblending results in Figure 5.15 and clean patch in Figure 5.14a. Blending factors (β) 2 and 4 and scaled by 5.0 and 3.0, respectively, for better visualization.

The above deblending process was repeated for 183 blended common-receiver gathers using the same solvers and denoisers. To avoid redundancy, the remaining analyses focus on the data with a blending factor of 2. Figure 5.17a and 5.17b show the 120-th common-shot gather in conventional and combed forms, respectively. Similarly, Figure 5.18a and 5.18b show the clean and combed common-receiver gathers, respectively. Figure 5.19a-5.19c show the separated shots, except for RED-ADMM with a single denoiser activation. The results resemble each other while also being in good agreement with the desired data in Figure 5.17a. The same is also visible from the difference panels in Figure 5.19d-5.19f, obtained with respect to the desired data and scaled 5.0 times for visualization. Further quantitative analysis also indicates similar results by the methods through their prediction gains. Figure 5.20 shows the result obtained with the FKT operator in the PGD, RED-FP, and RED-ADMM with $I = 1$ and $I = 2$ internal iterations. Figure 5.21 follows the same layout as before and shows the difference sections between the clean and deblended common-receiver gather after scaling by 5.0. The results are similar, although the prediction gains indicate improvement using RED.

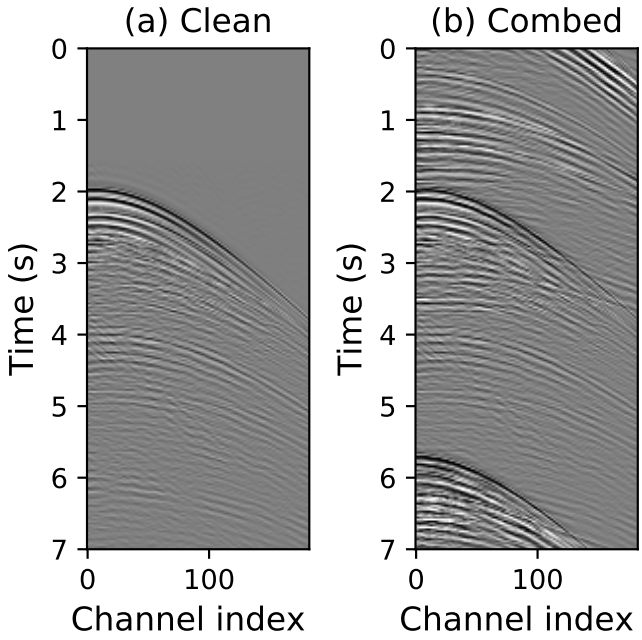

Figure 5.17: Common-shot gather view of the (a) conventional data, (b) pseudo-deblended data.

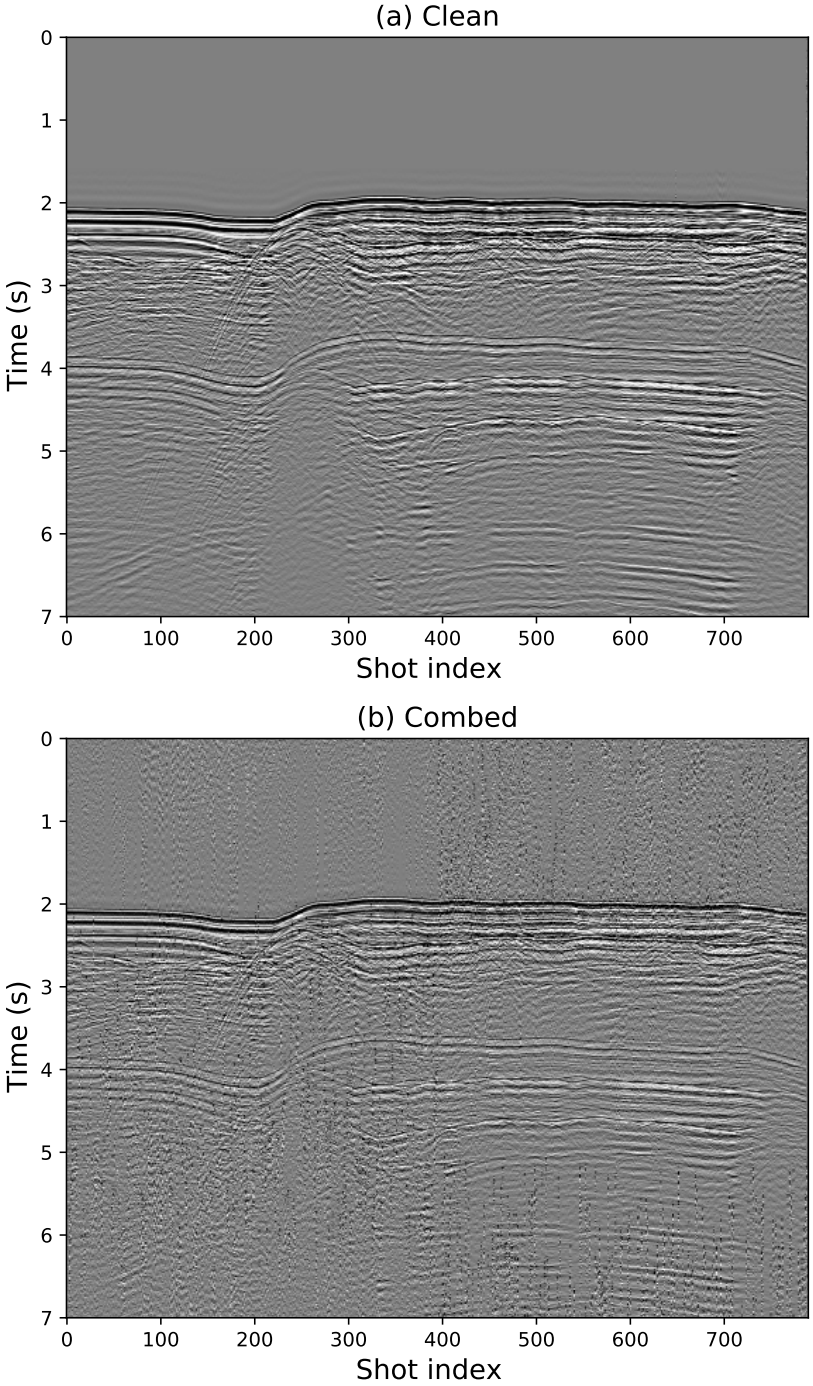

Figure 5.18: (a) Clean common-channel gather and (b) combed data for blending factor 2.

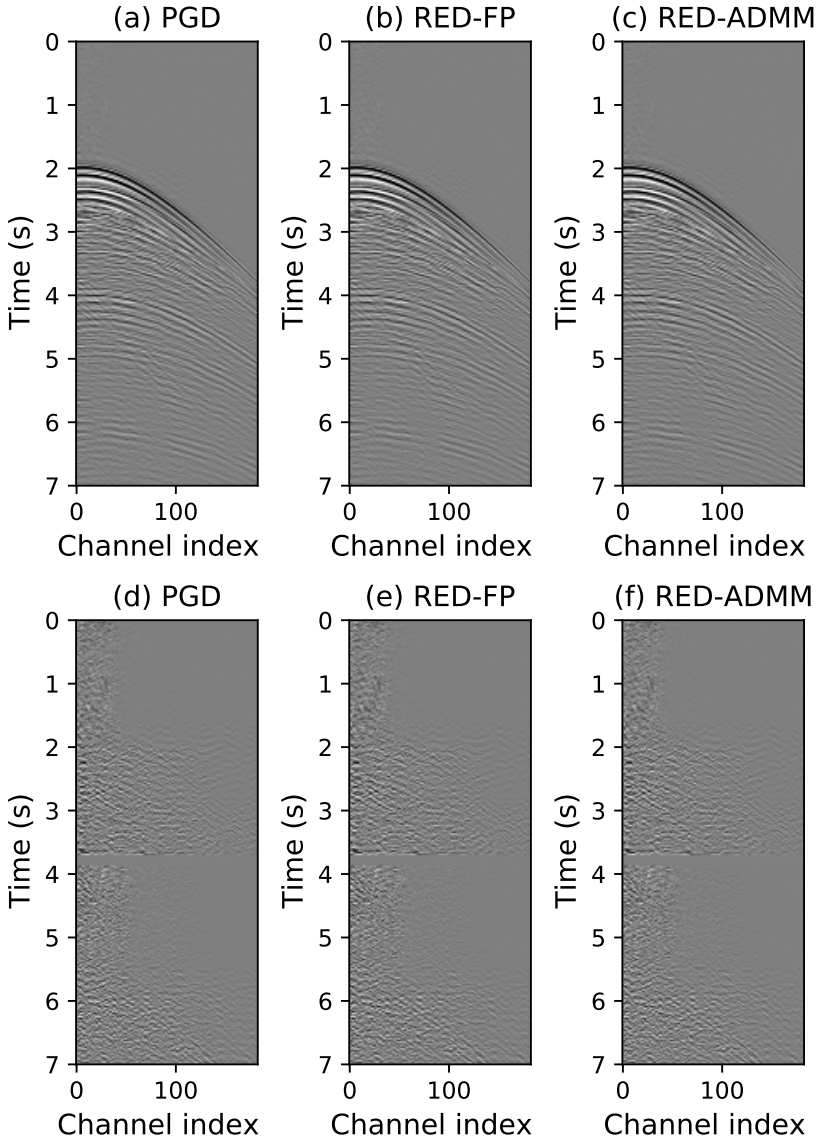

Figure 5.19: Common-shot gather view of the deblended results with the FKT denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 2$). The respective difference with the ideal data are scaled by 5.0 and given in panels (d)-(f).

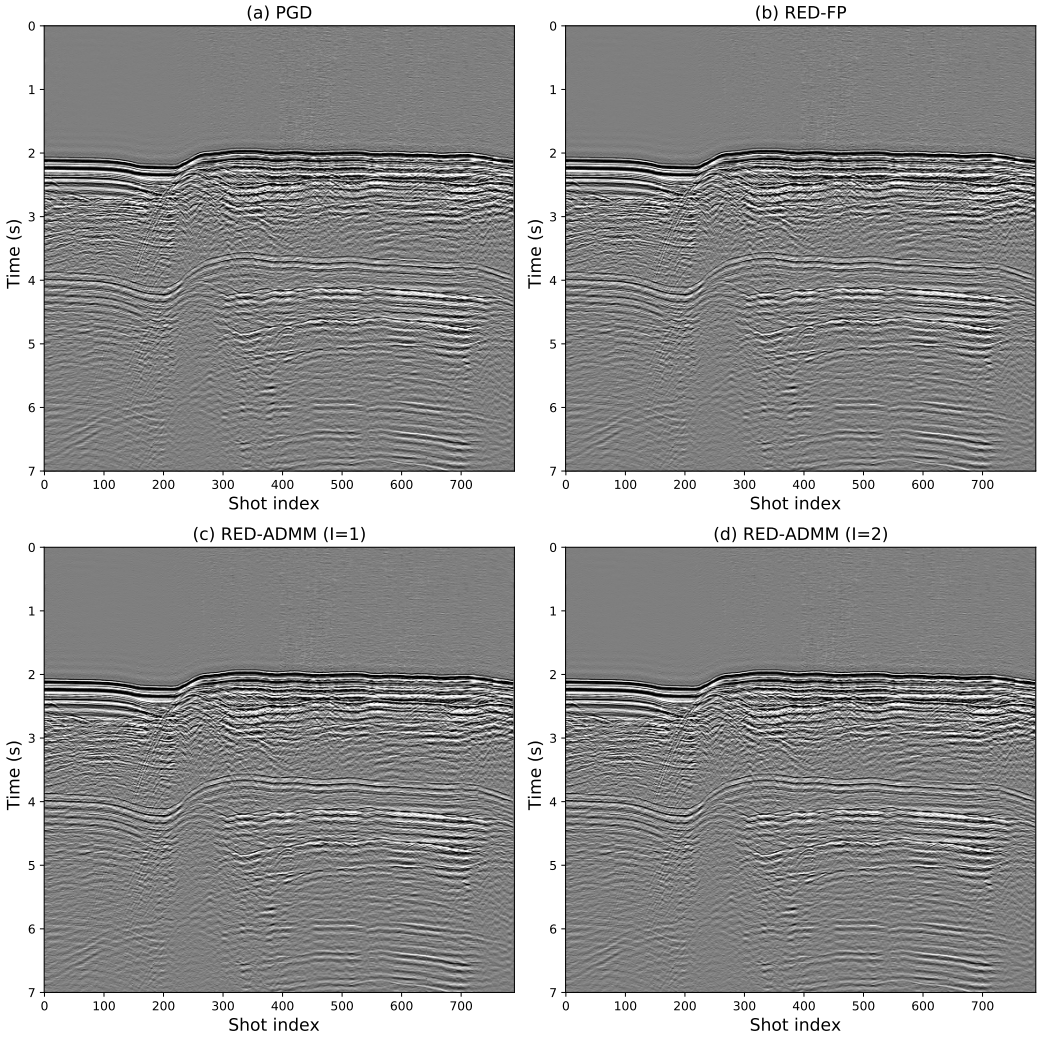

Figure 5.20: Deblended CRG results using the FKT operator in (a) PGD, (b) RED-FP, and RED-ADMM with (c) $I = 1$ and (d) $I = 2$ internal iterations.

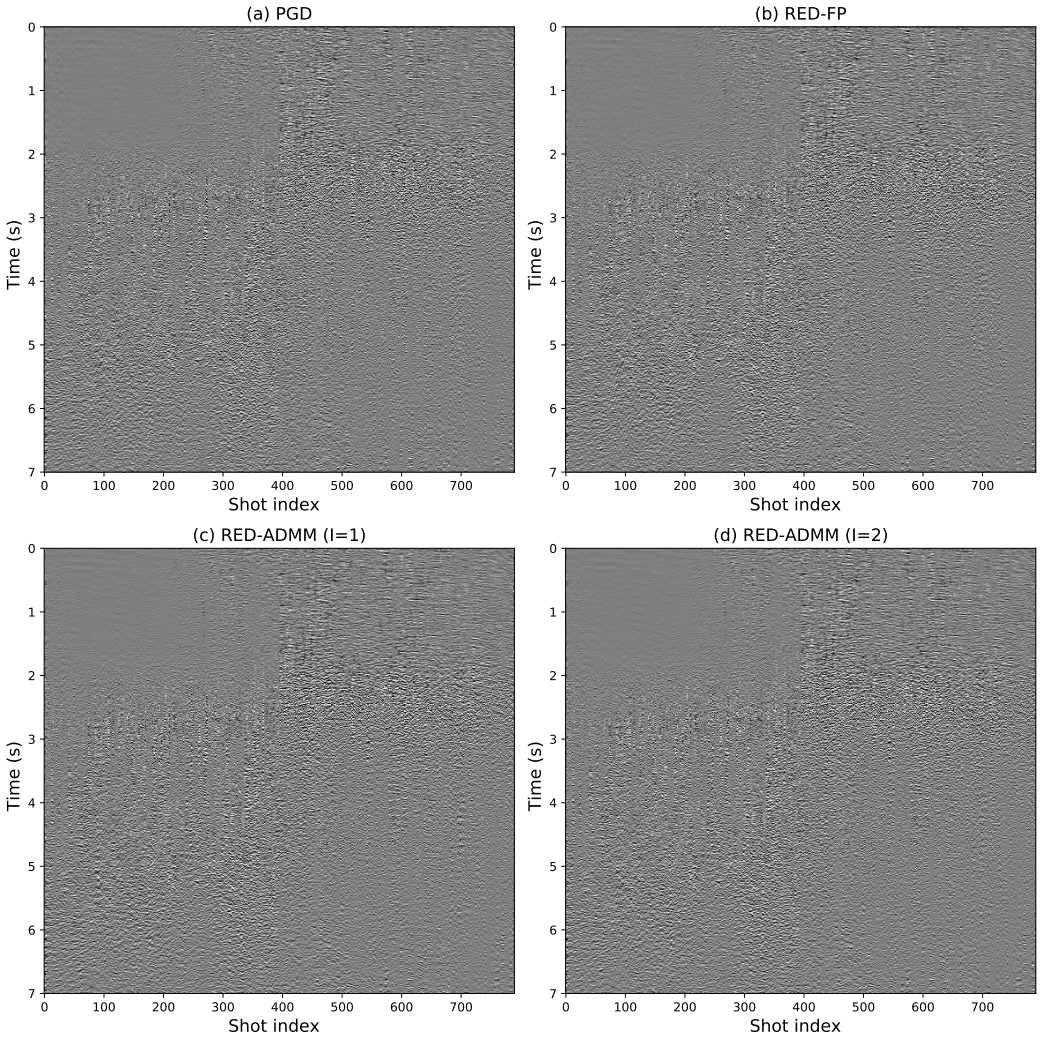

Figure 5.21: Difference sections between ideal data and deblending results using the FKT operator in (a) PGD, (b) RED-FP, and RED-ADMM with (c) $I = 1$ and (b) $I = 2$ internal iterations. All figures have been scaled 5.0 times for visualization.

Examples: SSA Denoiser

In its applications, the SSA had better performance with patches of size 256×32 in signal preservation, arguably because it better approximates the linear moveout assumption in the SSA theory. Finally, the SSA filter runs over the frequency range of 0 to 150 Hz in all cases, and it achieved the best results with decreasing ranks for increasing levels of interference; specifically, the rank was set to 7, 5 and 4 for blending factors 2, 4, and 8, respectively.

The results in Figure 5.22 show good attenuation of blending interference, but it is also noticeable from its scaled difference sections in Figure 5.23 that the method did not recover finer details of interest. RED could improve the final reconstruction gains shown in Table 5.3. Similar to the FKT operator, a second activation of the SSA denoiser in RED-ADMM was useful to obtain faster convergence towards the final solution. However, as it will become clear, its second application was redundant for the final results in all blending factors.

SSA			
Blending factor	2	4	8
PGD	14.11	8.75	5.04
RED-FP	15.54	9.30	5.27
RED-ADMM ($I = 1$)	16.56	9.53	5.36
RED-ADMM ($I = 2$)	16.56	9.53	5.36

Table 5.3: Final prediction gain obtained with the SSA denoiser within the PGD and RDB. The table containst results for all blending factors.

The process was repeated for the same 183 blended common-receiver gathers as before, and the remaining analyses are again focused on the data with a blending factor of 2. Figure 5.24 shows the common-receiver gathers obtained in each method. As indicated by the prediction gains, the results have higher levels of remaining noise than those obtained with the FKT denoiser (Figure 5.20), and lower levels of detail are being retrieved. That is also observed in the difference sections, shown in Figure 5.25, where one can observe fine signals leaking into the noise space after scaling by a factor of 5.0. Figure 5.26a-5.26c show common-shot gathers obtained with the SSA denoiser in PGD and RED. One can observe small levels of remaining blending interference in the results, and the comparison with the ideal data (Figure 5.17a) shows this interference more evidently in Figure 5.26d-5.26f.

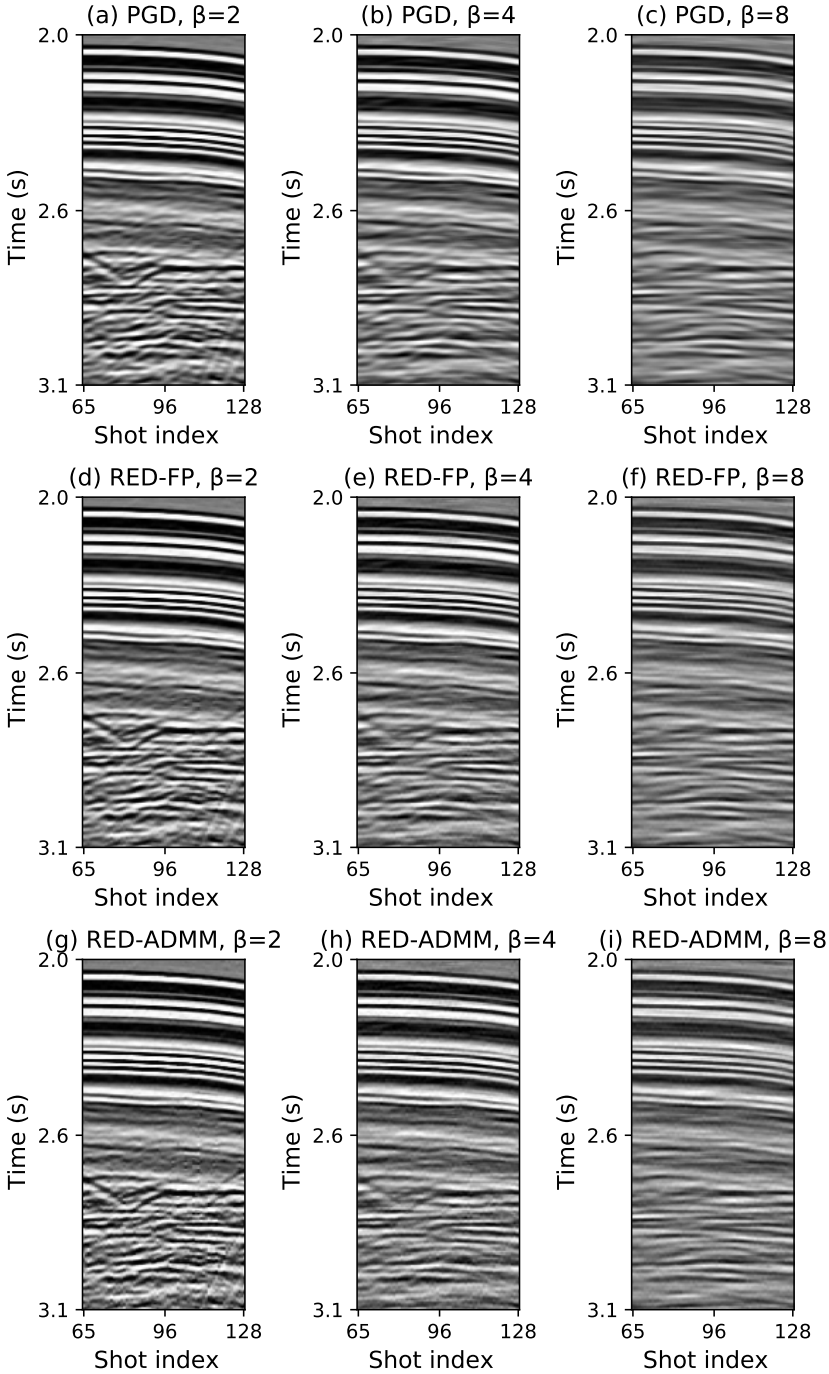

Figure 5.22: Deblending results with the SSA filter for blending factors (β) 2, 4 and 8 using the (a)-(c) PGD, (d)-(f) RED-FP and (g)-(i) RED-ADMM ($I = 2$).

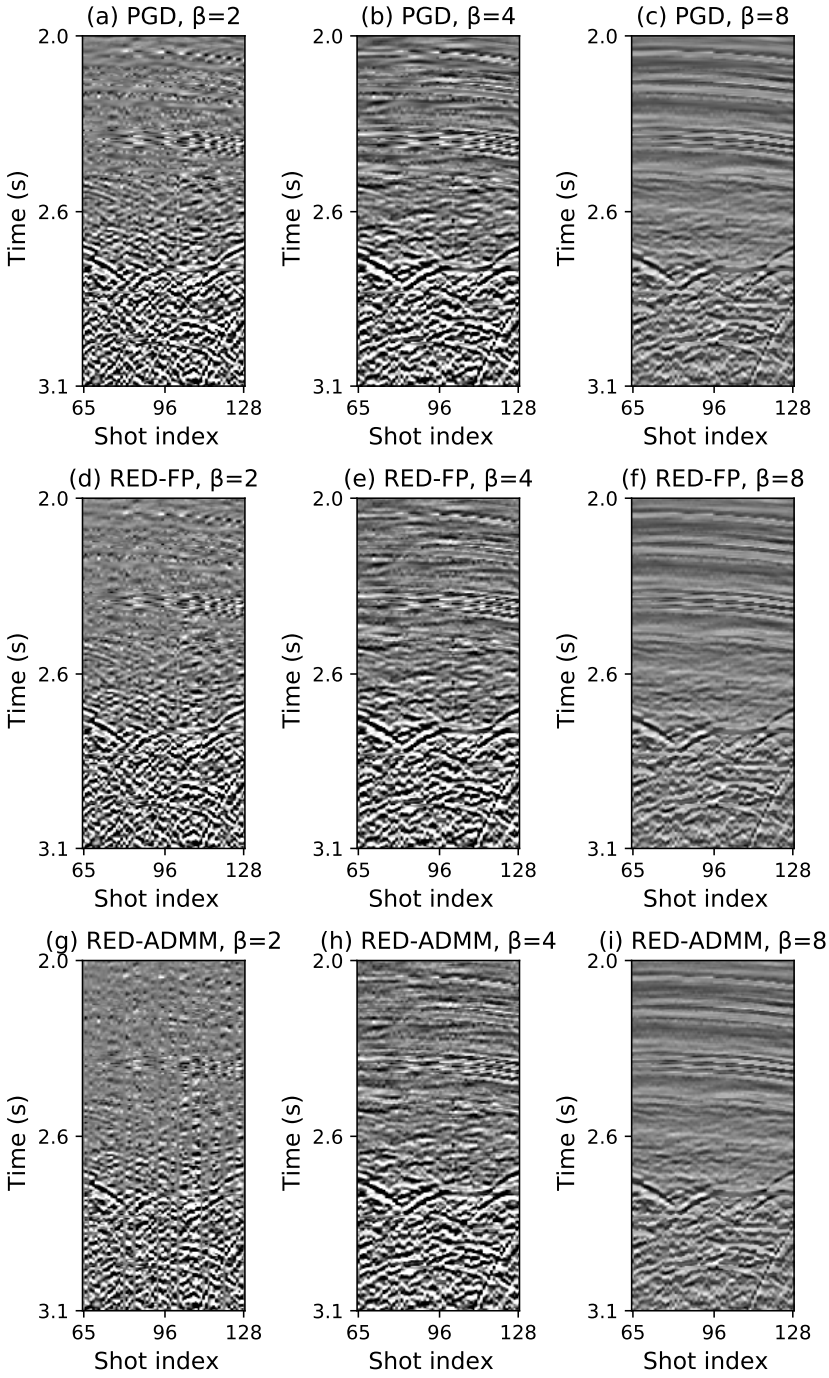

Figure 5.23: Difference between deblending results in Figure 5.22 and clean patch in Figure 5.14a. Blending factors (β) 2 and 4 and scaled by 5 and 3, respectively, for better visualization.

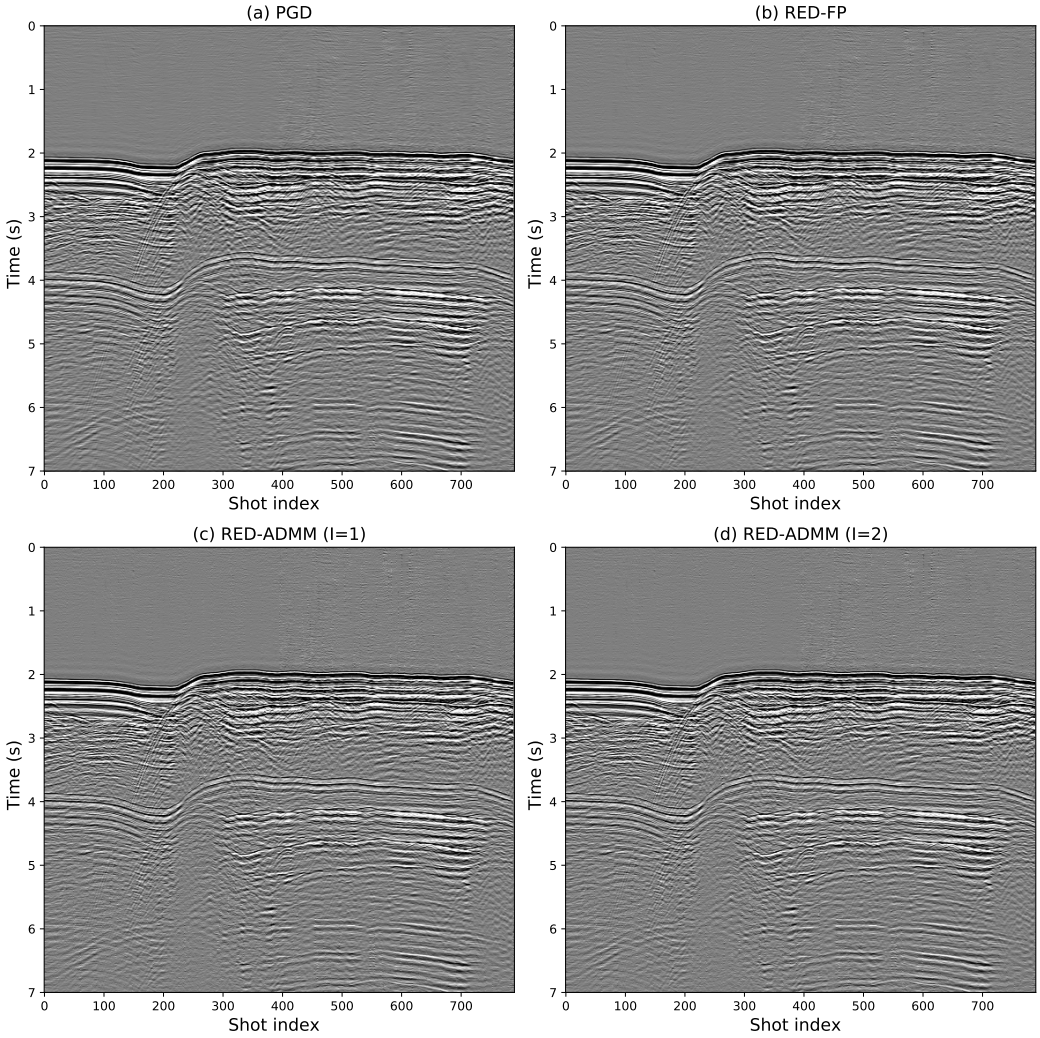

Figure 5.24: Deblended CRG results using the SSA operator in (a) PGD, (b) RED-FP, and RED-ADMM with (c) $I = 1$ and (b) $I = 2$ internal iterations.

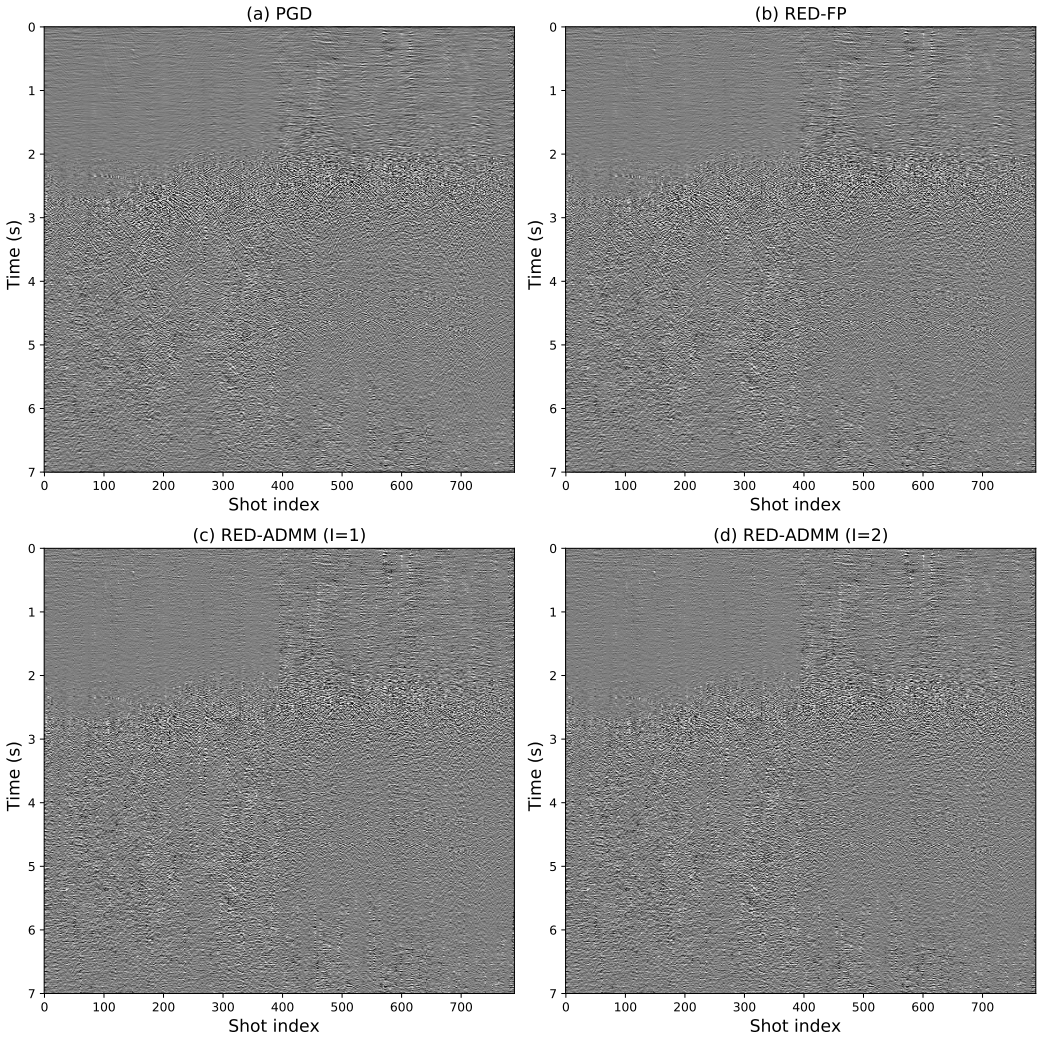

Figure 5.25: Difference sections between ideal data and deblending results using the SSA operator in (a) PGD, (b) RED-FP, and RED-ADMM with (c) $I = 1$ and (d) $I = 2$ internal iterations. All figures have been scaled 5.0 times for visualization.

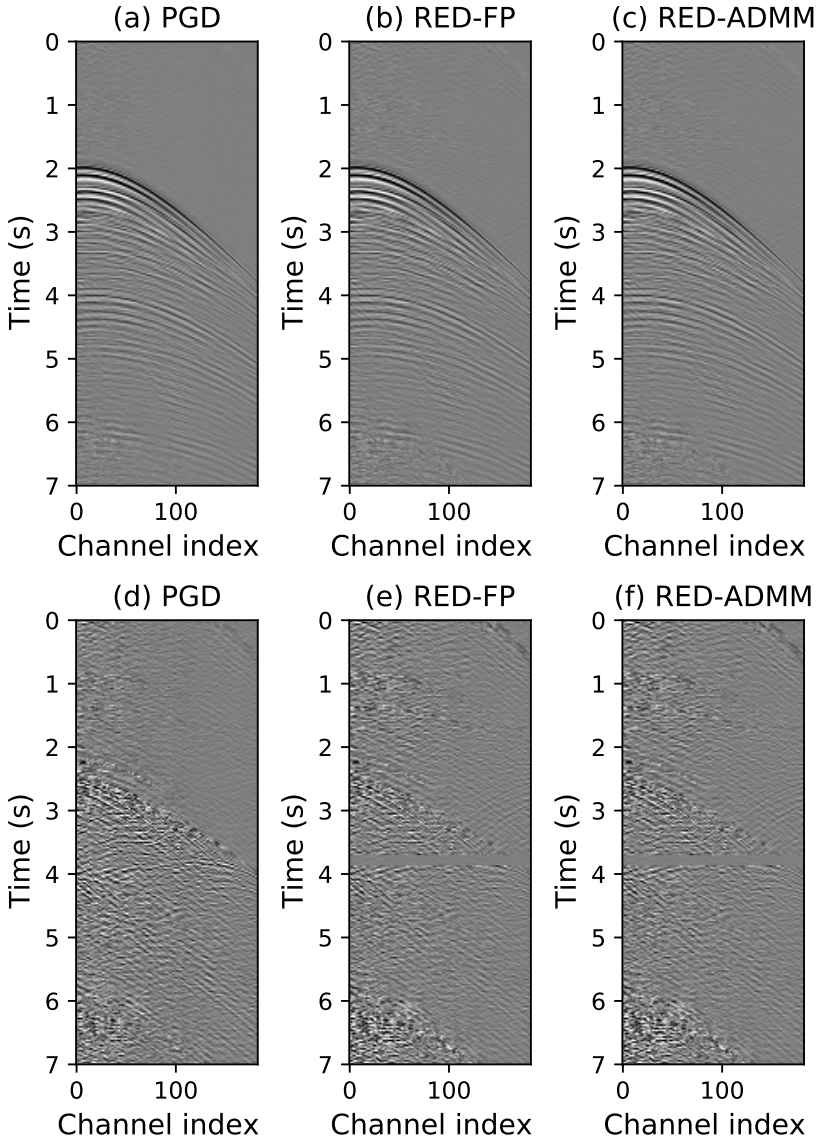

Figure 5.26: Common-shot gather view of the deblended results with the SSA denoiser in (a) PGD, (b) RED-FP, and (c) RED-ADMM ($I = 2$). The respective difference with the ideal data are scaled by 5.0 and given in panels (d)-(f).

5.3 Final Remarks

5.3.1 Discussion

The computational running times of REDeblending (RDB) are not primarily affected by the inner linear problems in each of its solvers, being mainly dictated by the number of activations of the denoiser. For instance, a second application of the FKT filter is not noticeable as it is an efficient filter based on fast Fourier transforms. In contrast, when running experiments with two applications of the SSA filter in RED-ADMM, the difference in running times is noticeable. In the case of vector-valued datasets, this goes beyond the fact that the volume of data is three times larger; the quaternionic formulations suffer more for being based on a heavier algebra, as discussed throughout this thesis.

The separation results obtained with RED are arguably superior to those obtained with the PGD algorithm for both scalar and vector-valued datasets. This difference is more noticeable in the 3D synthetic examples, although the 2D application using numerically blended field data also shows superior results with RED. For both cases, however, the differences in the final results are more visible when contrasting denoisers than the solvers being used. Namely, compared with the SSA and QSSA filters, the FKT and QFKT results also turn out to be superior in deblending quality. This conclusion can also be drawn from quantitative inspections using the prediction gains (equation 5.27).

One worthy difference between the solvers is how the prediction gains evolve per iteration, as shown in Figure 5.27a and 5.27b for the FKT and SSA denoisers, respectively. These also compare the scalar and vector-valued deblending quality of the data's vertical component, which improved all the results obtained with the QFKT denoiser. It is also visible that RED outperformed the PGD in all instances of this particular example.

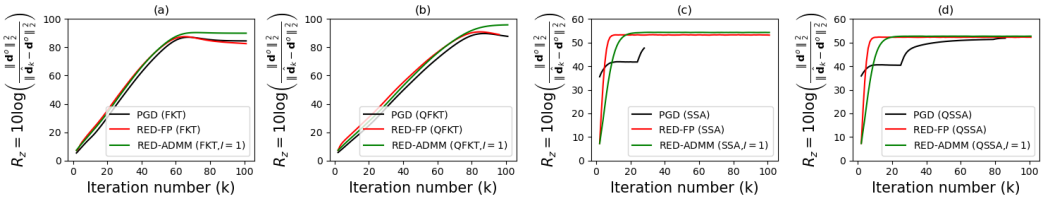

Figure 5.27: The prediction gain evolution with iteration for the vertical component of the 3D-3C VSP data using (a) and (b) FKT and QFKT and (c) and (d) SSA and QSSA denoisers, respectively. Notice the difference in scale between the plots (a)-(b) against (c)-(d).

Using the 2D numerically blended field dataset, on the other hand, smaller differences were observed. Still, as shown in Figure 5.28a-5.28c and 5.29a-5.29c for the FKT and SSA

denoisers, respectively, it is possible to observe superior performance in the prediction gains obtained with RDB at each iteration. In this case, for the FKT denoiser, the PGD could marginally outperform RDB only for the blending factor of 8. Also, it is possible to notice how the second activation of the denoising engine was beneficial in all blending factors for the FKT. The same was not valid for the SSA filter, in which the second application of the denoiser was redundant for all blending factors, although the method converged towards the final solution using fewer iterations. An additional marker was used to indicate the overlapping curves in Figure 5.29 for better visualization. Finally, when used with the SSA filter, the PGD algorithm was constantly outperformed by RED.

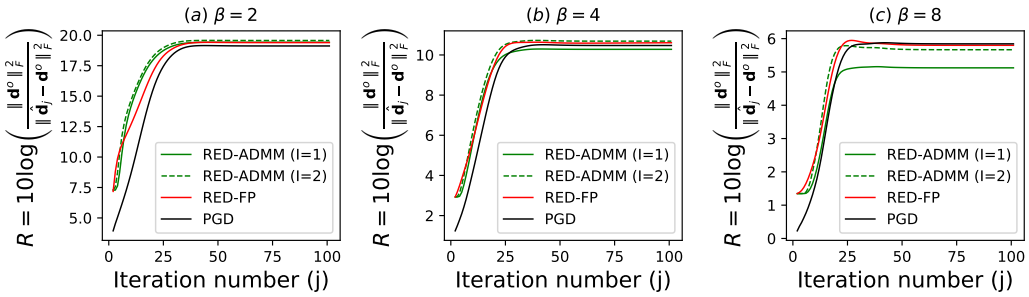

Figure 5.28: The prediction gain evolution with iteration for all solvers using the FKT operator. The blending factors (β) are 2 (a), 4 (b), and 8 (c). Note the difference in vertical scale.

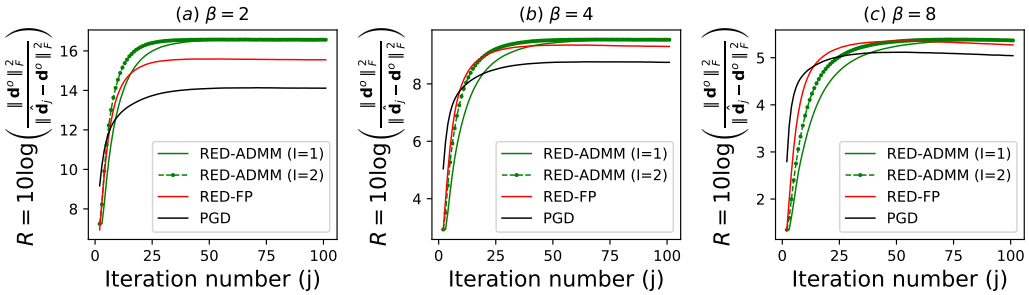

Figure 5.29: The prediction gain evolution with iteration for all solvers using the SSA operator. The blending factors (β) are 2 (a), 4 (b), and 8 (c). Note the difference in the vertical scale and the additional marker for RED-ADMM ($I = 2$).

Even though there are numerous similarities between RED and the PGD, these approaches are conceptually different. In the light of existing work in deblending, such as that of Mahdad et al. (2011), the PGD algorithm solves a constrained inversion problem in which the projection operator is implemented as the action of a denoiser. RED, however, solves a regularized inverse problem in which the regularization term explicitly exploits a denoiser

in its definition. Consequently, the algorithms devised for solving either problem differ significantly; RED-ADMM is the clearest example of such differences. It should be expected, however, that the results obtained with PGD and RED to be similar for a given denoiser because, after all, it is the action of the denoiser that imposes prior information onto the solution. The FKT, for instance, encourages sparse solutions in the frequency-wavenumber domain for all methods, thus performing well in applications such as simultaneous source separation. Overall, just like PGD, RED is a general framework for solving inverse problems with great value to the geophysics community since denoising engines have been developed to a great extent already, and more are expected to come.

5.3.2 Conclusions

This chapter exploits the regularization by denoising (RED) framework in simultaneous sources separation of scalar and vector-valued datasets. RED proposes a regularization term in its cost function that employs denoising engines of choice in its definition. Such a cost function has a clear interpretation of data fitting coupled with strong noise attenuation. Minimizing the RED cost function can be achieved via first-order optimization methods. Underlying this rather desirable workflow are the RED conditions, which a denoiser of interest would have to satisfy to validate its gradient. Together with differentiability, the conditions are local homogeneity, strong passivity, and Jacobian symmetry.

RED is a topic of active research, as it is clear from recent interest in signal processing and inversion communities. The last condition, Jacobian symmetry, was recently added to the list to make the gradient valid for a suitable denoiser. This is a potential setback on RED because many highly in-use denoisers lack Jacobian symmetry, sometimes even being non-differentiable with respect to the model of interest. Nevertheless, RED can still deliver high-quality results with such denoisers. For example, a classic denoiser such as the FKT used here, which does not fully satisfy the local homogeneity condition, still could perform well in RED for simultaneous sources separation.

RED could achieve excellent performance in simultaneous sources separation, comparable to Fourier sparse inversion results. One can use arbitrary denoisers in RED if compliance with its conditions is assumed. The second example adopted in this paper was the SSA filter, which was also employed as a spatial coherency constraint for deblending within the projected gradient descent algorithm. RED could achieve good performance in all instances, and the results are satisfactory. When comparing the results among denoisers, the FKT operator outperformed the SSA filter, but one can relate it to the denoisers' setup. As discussed, the performance of both PGD and RED highly depends on the denoiser of choice, and a smartly formulated denoiser is likely to perform better too.

CHAPTER 6

Conclusions

6.1 Summary

Multicomponent seismology offers a better characterization of the subsurface as it samples the three orthogonal components of the vector-valued wavefield. The acquired components still suffer from all issues faced in classic (scalar) seismic data processing, such as noise and missing observations. Processing vector-valued datasets impose a series of physical constraints onto the data that are usually ignored in the commonly used componentwise processing, which treats each component as its scalar entity. Such an approach completely disregards any possible relationship between the acquired data components. Vectorial processing approaches aim to include all possible useful information from the data components into steps such as data denoising and recovery. This thesis exploited the noncommutative quaternion algebra to formulate vector-valued data processing techniques for multicomponent seismic data. The methods developed here exploit the correlation between data components following physically meaningful vector-valued approaches. For such, Chapter 2 provides the necessary introduction to quaternion numbers and their algebra, along with the extension of well-known data processing techniques, such as the Fourier transform (FT) and Singular Value Decomposition (SVD), which pave the way to quaternionic signal processing methods. Also importantly, the second-order statistics of quaternion random vectors were introduced to exploit the correlation between quaternion components through the basis augmentation given by the quaternion involutions.

Multidimensional seismic data reconstruction is a common problem in seismic data processing. There are several methods to perform seismic data reconstruction, one of them being the frequency-space ($f-x$) Singular Spectrum Analysis (SSA) filter, extended in Chapter 3

to the case of multicomponent datasets using its quaternionic version (QSSA). The extension requires defining quaternion-valued seismic data in the time domain through the vector components of displacement fields. Its frequency-domain representation is obtained using the Quaternion Fourier transform (QFT), which requires the definition of an eigenaxis and side for the transformation. Following the advances in augmented statistics, an augmented version of the QSSA (AQSSA) can exploit the additional information between data components. The usual SSA steps are still the same in its quaternionic formulation, but the classically expensive SVD has to be replaced by the prohibitive QSVD. The efficient formulations of rank-reduction can be readily presented in the quaternion domain as in Appendix A of this thesis. Given the Hankel structure of the matrices used by the SSA method, spectral-domain convolutions in the quaternion domain can also be used to increase the feasibility of these approaches for demanding applications such as multidimensional seismic data reconstruction. The methods were tested in reconstructing multidimensional synthetic datasets, thus showing the effectiveness of the vector approaches. Both QSSA and AQSSA could improve upon the results from the componentwise reconstruction.

Chapter 4 exploits the underlying models explaining the low-rank character of the Hankel matrices in Chapter 3, known as auto-regressive models. These models are used to define classic seismic data processing techniques, such as the so-called prediction filters (or FXDECON), known to be efficient and effective denoisers. The low-rankness of quaternion Hankel matrices thus suggests that a low-order quaternion-valued auto-regressive model can be used to develop a quaternionic predictive filter for vector-valued seismic data. Similar to the QSSA, the method is defined in the frequency-space domain, and the data is required to be quaternion Fourier transformed. Each frequency slice is processed independently on both sides of the spectrum, giving rise to a technique named QFXDECON. For complete second-order characterization of the quaternion signal, a widely-linear auto-regressive model is used to define the WL-QFXDECON capable of exploiting additional useful information from the correlation between data components. In their applications to synthetic and field multicomponent data examples, the methods were effective in signal preservation.

While Chapters 3 and 4 introduce denoising and reconstruction methods based on the low-rankness and predictability of plane waves in the frequency-space domain, Chapter 5 argues that those priors persist in the definition of many denoising algorithms that can be further exploited with the Regularization by Denoising (RED) framework in diverse applications. As an example, Chapter 5 solves the simultaneous sources separation problem through RED in a formulation that permits developing deblending algorithms where the user can select any denoising engine that satisfies RED conditions discussed in Appendix B. Two popular denoisers in the seismic processing community are tested: frequency-wavenumber thresholding (FKT) and singular spectrum analysis (SSA). The results show that RED can deliver high-quality results for scalar and vector-valued seismic deblending, comparable or

superior to those obtained by state-of-the-art techniques such as sparse inversion.

6.2 Contributions

One of the contributions of this thesis consists of the implementation of physically meaningful vector-valued data processing techniques tailored for multicomponent seismic data. Thanks to the developments in quaternion-based signal processing, no intermediate scalar steps were necessary to formulate the techniques presented in this thesis. These were formulated based on the quaternionic counterparts of classic data processing techniques, which are discussed in Chapter 2. This chapter provided an overview of quaternion numbers, quaternionic signals, and their processing. Altogether, it represents the description of a fully vector-based signal processing framework upon which many techniques can be based.

One example of such techniques is the extension of the QSSA to the case of vector-valued seismic data presented in Chapter 3. The classic (scalar) SSA filter operates in the $f - x$ domain and exploits the low-rank character of Hankel matrices built from the frequency slices of the data. It can be shown that the rank of these matrices corresponds to the number of plane waves present in the time domain data. The quaternion-signal model represents the multicomponent seismic data in the time domain, thus avoiding any scalar processing of vector quantities. Its $f - x$ representation is obtained through the QFT, and the low-rank character of the Hankel matrices built from the quaternionic frequency slices is shown to be preserved. The QSSA can thus be used for multicomponent seismic data denoising and reconstruction. However, there is no guarantee that the direct extension of classic signal processing techniques to their quaternion counterparts will exploit the correlation between data components. The inclusion of augmented second-order statistics in the QSSA represents an essential contribution to processing vector-valued seismic data. The resulting AQSSA provides an explicit way to exploit second-order information from quaternionic signals, improving results over the SSA and QSSA. One known drawback of rank-reduction techniques, however, is their computational requirements. With the heavier quaternion algebra, developing seismic data processing methods based on rank-reduction is further challenging. Following the scalar case, one possible solution is to exploit the structure of the Hankel matrix to implement the QSSA filter efficiently. Such implementation is based on the spectral convolution theorem, which has a non-trivial extension to the quaternion domain. Fast Fourier transforms (FFTs) significantly improve the method's efficiency in the vector case. These efficient extensions were vital in making it possible to apply the method to multidimensional vector-valued seismic data reconstruction.

As for the SSA, the scalar FXDECON is a classic seismic data processing technique commonly present in industrial workflows. Using quaternion auto-regressive (Q-AR) models,

again in conjunction with the QFT, a hypercomplex version of the $f - x$ prediction filter, the QFXDECON, was formulated for the case of vector-valued seismic data in Chapter 4. Again, the method is fully vectorial and does not require intermediate scalar steps. For complete second-order signal characterization, where the correlation between data components can be fully exploited, augmented statistics have to be considered. Similar to the AQSSA, the quaternion involutions are used as an additional basis for describing the observations, thus leading to a widely-linear model of the quaternionic frequency slices. The contribution is introduced as the WL-QFXDECON and includes additional useful information from the data components in its formulation. Its underlying augmentation step creates a four-fold redundancy in the data samples for signal and noise alike, which can be helpful for better signal preservation. However, as the noise is augmented, the results will look noisier than their strictly linear counterpart. The methods retain the FXDECON's simplicity and efficiency and exemplify the possibility of developing vector-valued data processing methods that are physically consistent and make the most out of the acquired data.

Chapter 5 introduces RED for applications in scalar and vector-valued seismic data processing. More specifically, RED was used to solve the simultaneous sources separation problem. The formulation leads to deblending algorithms that can explicitly employ denoising techniques, including those developed in Chapters 3 and 4 of this thesis, capable of delivering solutions comparable to state-of-the-art methods. The denoisers are used to implicitly impose desired priors, such as sparsity or low-rankness in transform domains, as exploited in this chapter through the FKT and SSA, respectively. Their choice is steered by the underlying character of plane waves described by low-rank, predictable, and sparse models in transform domains. These are favorable priors to applications such as deblending and reconstruction. Given the extension of quaternion-based denoising and reconstruction techniques for vector-valued data, Chapter 5 also presents synthetic applications of vector-valued deblending in an ad-hoc fashion. Notice that defining a quaternion-based RED cost function inherits its conceptual difficulties already from the real case, which are discussed here in the light of the RED conditions. These conditions and their corresponding numerical tests are summarized and discussed for the real case only. Arguably, RED is still a developing method, and this chapter summarizes its developments, from its conception up to its clarification. Such a contribution is expected to drive further interest in using RED for other applications. The examples with synthetic and real numerically blended data showed that RED performs competitively with leading current methods for both scalar and vector cases.

6.3 Future Work

The recent development in quaternion-based signal processing has equipped the field with enough tools to fuel a range of applications in processing color images and other vector-valued signals. These tools introduce additional parameters to the processing routines, such as the eigenaxis and side of the QFT. The techniques presented in this thesis did not explore any particular way of defining them. Further efforts can be made in that direction to boost quaternion-based techniques further. The methods presented here are extensions of classic scalar seismic data processing techniques to their vector-valued case. Therefore, they are defined based on plane-wave assumptions which are approximated by data windowing. No other multicomponent processing was conducted for futher appraisal, and the assessment was based on the reconstruction quality as routinely done in the signal processing community. Futhermore, many quaternion-based extensions can be proposed to process multicomponent seismic data. For example, in the case of irregular vector-valued seismic data, a quaternionic anti-leakage Fourier transform (Xu et al., 2005) can be promptly proposed. Likewise, the I-MSSA proposed by Carozzi and Sacchi (2021) also has a trivial quaternionic extension that can be exploited to reconstruct irregular multicomponent traces. Projection filters (Soubaras, 1994), which represent an improvement over the FXDECON, also stand as a direction for future research with quaternions since the predictive models were particularly effective in signal predictability. These filters provide a possible way to reduce the noise levels in the results obtained with widely-linear prediction models used in the WL-QFXDECON.

Simultaneous source separation and seismic data reconstruction can be solved through similar methods based on the related sparse and low-rank priors developments. For instance, both problems can be solved through the projection-onto-convex-sets (POCS) algorithm as given by Abma and Kabir (2006) and Abma et al. (2015) for reconstruction and deblending, respectively, which exploits sparsity priors in its formulations. In addition, it is also usual for these problems to be solved simultaneously (Cheng and Sacchi, 2015). In this thesis, RED was introduced as a flexible framework that allows the inclusion of well-established denoising techniques within the regularization term of the cost function for general inverse problems. RED was tested in different blending noise conditions, but examples presented here do not include issues such as strong first arrivals that highly mask weaker signals from earlier sources. While a Gaussian fidelity term is still welcome to fit the blended observations, the denoisers employed require stronger noise attenuation and signal recovery capabilities. Methods based on robust statistics and redescending operators (Maronna et al., 2019) have shown good performance in these conditions and also represent a further point of improvement for the methods presented in this thesis. Finally, the priors mentioned above, being successful in REDeblending, also suggest RED as a powerful reconstruction framework.

Bibliography

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

Abma, R., and M. S. Foster, 2020, Simultaneous source seismic acquisition: SEG.

Abma, R., D. Howe, M. Foster, I. Ahmed, M. Tanis, Q. Zhang, A. Arogunmati, and G. Alexander, 2015, Independent simultaneous source acquisition and processing: Geophysics, **80**, no. 6, WD37–WD44.

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

Akerberg, P., G. Hampson, J. Rickett, H. Martin, and J. Cole, 2008, Simultaneous source separation by sparse Radon transform: SEG Technical Program Expanded Abstracts 2008, Society of Exploration Geophysicists, 2801–2805.

Anagaw, A. Y., and M. D. Sacchi, 2019, Edge-preserving FWI via Regularization by Denoising: Presented at the GeoConvention 2019.

Andersson, F., D.-J. V. Manen, J. Wittsten, K. Eggenberger, and J. O. A. Robertsson, 2017, *in* Quaternion dealising for simultaneous source separation: 4322–4327.

Bahia, B., I. Papathanasaki, and M. D. Sacchi, 2020, *in* Ground-roll attenuation through quaternionic inversion with sparsity constraints: 3254–3258.

Bahia, B., and M. Sacchi, 2019a, Robust singular spectrum analysis via the bifactored gradient descent algorithm: SEG Technical Program Expanded Abstracts 2019, Society of Exploration Geophysicists, 4640–4644.

Bahia, B., and M. D. Sacchi, 2019b, Quaternionic rank-reduction methods for vector-field seismic data processing: Digital Signal Processing, **87**, 178 – 189.

Beasley, C. J., R. E. Chambers, and Z. Jiang, 1998, A new look at simultaneous sources: SEG Technical Program Expanded Abstracts 1998, SEG, 133–135.

Berkhout, A. J. G., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, **27**, no. 7, 924–938.

Bezanson, J., S. Karpinski, V. B. Shah, and A. Edelman, 2012, Julia: A fast dynamic language for technical computing: arXiv preprint arXiv:1209.5145.

Blumensath, T., and M. E. Davies, 2008, Iterative thresholding for sparse approximations: Journal of Fourier Analysis and Applications, **14**, no. 5-6, 629–654.

Bonar, D., and M. Sacchi, 2012, Denoising seismic data using the nonlocal means algorithm: Geophysics, **77**, A5–A8.

Boyd, S., N. Parikh, and E. Chu, 2011, Distributed optimization and statistical learning via the alternating direction method of multipliers: Now Publishers Inc.

Bro, R., 1997, Parafac. tutorial and applications: Chemometrics and intelligent laboratory systems, **38**, 149–171.

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

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

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

Carozzi, F., 2021, Rank-constrained multidimensional reconstruction of seismic data: PhD thesis, University of Alberta.

Carozzi, F., and M. D. Sacchi, 2019, Robust tensor-completion algorithm for 5D seismic-data reconstruction: Geophysics, **84**, no. 2, V97–V109.

———, 2021, Interpolated multichannel singular spectrum analysis: A reconstruction method that honors true trace coordinates: Geophysics, **86**, V55–V70.

Carozzi, F., and M. D. S. Sacchi, 2017, 5D seismic reconstruction via parallel matrix factorization with randomized QR decomposition: Presented at the 2017 SEG International Exposition and Annual Meeting, Society of Exploration Geophysicists.

Chase, M. K., 1992, Random noise reduction by 3-d spatial prediction filtering, *in* SEG Technical Program Expanded Abstracts 1992: Society of Exploration Geophysicists, 1152–1153.

Chen, K., and M. D. Sacchi, 2014, Robust reduced-rank filtering for erratic seismic noise attenuation: Geophysics, **80**, V1–V11.

Chen, Y., S. Fomel, and J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: Geophysics, **79**, no. 5, V179–V189.

Cheng, J., and M. Sacchi, 2017, Computational efficient multidimensional singular spectrum analysis, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 4312–4316.

Cheng, J., and M. D. Sacchi, 2015, Separation and reconstruction of simultaneous source data via iterative rank reduction: Geophysics, **80**, no. 4, V57–V66.

Chiron, L., M. A. van Agthoven, B. Kieffer, C. Rolando, and M.-A. Delsuc, 2014, Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry: Proceedings of the Na-

tional Academy of Sciences, **111**, 1385–1390.

Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications.

Crawley, S., R. Clapp, and J. Claerbout, 1999, Interpolation with smoothly nonstationary prediction-error filters, *in* SEG Technical Program Expanded Abstracts 1999: Society of Exploration Geophysicists, 1154–1157.

Davies, A., R. Foot, G. Joshi, and B. McKellar, 1989, Quaternionic methods in integral transforms of geophysical interest: Geophysical Journal International, **99**, 579–582.

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

Donoho, D. L., 2006, Compressed sensing: IEEE Transactions on information theory, **52**, 1289–1306.

Doulgeris, P., K. Bube, G. Hampson, and G. Blacquière, 2012, Convergence analysis of a coherency-constrained inversion for the separation of blended data: Geophysical Prospecting, **60**, no. 4, 769–781.

Eckart, C., and G. Young, 1936, The approximation of one matrix by another of lower rank: Psychometrika, **1**, 211–218.

Ell, T. A., 1992, Hypercomplex spectral transformations: PhD thesis, University of Minnesota.

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

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

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

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

Flamant, J., S. Miron, and D. Brie, 2021, A general framework for constrained convex quaternion optimization: IEEE Transactions on Signal Processing, **70**, 254–267.

Freire, S. L., and T. J. Ulrych, 1988, Application of singular value decomposition to vertical seismic profiling: Geophysics, **53**, 778–785.

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

———, 2016, 3c seismic and vsp: Converted waves and vector wavefield applications: Society of Exploration Geophysicists.

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

V30.

Garotta, R., 1999, Shear waves from acquisition to interpretation: Society of Exploration Geophysicists.

Gezerlis, A., 2020, Roots, *in* Numerical Methods in Physics with Python: Cambridge University Press, 243–310.

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

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

Gülünay, N., 1986, FXDECON and complex Wiener prediction filter, *in* SEG Technical Program Expanded Abstracts 1986: Society of Exploration Geophysicists, 279–281.

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

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

Hamilton, W. R., 1844, Ii. on quaternions; or on a new system of imaginaries in algebra: The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, **25**, 10–13.

———, 1866, Elements of quaternions: Longmans, Green, & Company.

Hennenfent, G., and F. J. Herrmann, 2008, Simply denoise: Wavefield reconstruction via jittered undersampling: Geophysics, **73**, V19–V28.

Herrmann, F. J., 2010, Randomized sampling and sparsity: Getting more information from fewer samples: Geophysics, **75**, WB173–WB187.

Herrmann, F. J., D. Wang, G. Hennenfent, and P. P. Moghaddam, 2008, Curvelet-based seismic data processing: A multiscale and nonlinear approach: Geophysics, **73**, A1–A5.

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

Hong, T., Y. Romano, and M. Elad, 2019, Acceleration of RED via vector extrapolation: Journal of Visual Communication and Image Representation, **63**, 102575.

Hong, T., I. Yavneh, and M. Zibulevsky, 2020, Solving RED with Weighted Proximal Methods: IEEE Signal Processing Letters, **27**, 501–505.

Huo, S., Y. Luo, and P. G. Kelamis, 2012, Simultaneous sources separation via multidirectional vector-median filtering: Geophysics, **77**, no. 4, V123–V131.

Ibrahim, A., and M. D. Sacchi, 2013, Simultaneous source separation using a robust Radon transform: Geophysics, **79**, no. 1, V1–V11.

Ibrahim, A., and D. Trad, 2020, Inversion-based deblending using migration operators: Geophysical Prospecting, **68**, no. 8, 2459–2470.

Jeong, W., C. Tsingas, and M. S. Almubarak, 2020, A numerical study on deblending

of land simultaneous shooting acquisition data via rank-reduction filtering and signal enhancement applications: Geophysical Prospecting, **68**, 1742–1757.

Jones, I., and S. Levy, 1987, Signal-to-noise ratio enhancement in multichannel seismic data via the karhunen-love transform: Geophysical Prospecting, **35**, 12–32.

Kamil, Y. I., P. L. Vermeer, and M. Vassallo, 2015, Noise attenuation in multimeasurement streamer data using weighted vector auto regressive operators, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 4636–4640.

Kreimer, N., and M. D. Sacchi, 2012, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation: Geophysics, **77**, no. 3, V113–V122.

Krieger, L., and F. Grigoli, 2015, Optimal reorientation of geophysical sensors: A quaternion-based analytical solution: Geophysics, **80**, F19–F30.

Kumar, R., H. Wason, and F. J. Herrmann, 2015, Source separation for simultaneous towed-streamer marine acquisition a compressed sensing approach: Geophysics, **80**, no. 6, WD73–WD88.

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

Le Bihan, N., S. Miron, and J. I. Mars, 2007, MUSIC algorithm for vector-sensors array using biquaternions: IEEE Transactions on Signal Processing, **55**, 4523–4533.

Li, C., C. C. Mosher, and Y. Ji, 2019, An amplitude-preserving deblending approach for simultaneous sources: Geophysics, **84**, no. 3, V185–V196.

Li, F., J. Gao, Y. Wang, and J. Lu, 2021, Multi-component seismic data reconstruction based on vector POCS method via complexified quaternion Fourier transform: 82nd EAGE Annual Conference & Exhibition, European Association of Geoscientists & Engineers, 1–5.

Liberty, E., F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, 2007, Randomized algorithms for the low-rank approximation of matrices: Proceedings of the National Academy of Sciences, **104**, 20167–20172.

Lin, R., and M. D. Sacchi, 2020, Separation of simultaneous sources acquired with a high blending factor via coherence pass robust Radon operators: Geophysics, **85**, no. 3, V269–V282.

Lin, T. T., and F. Herrmann, 2009, Designing simultaneous acquisitions with compressive sensing: 71st EAGE Conference and Exhibition incorporating SPE EUROPEC 2009, European Association of Geoscientists & Engineers, cp–127.

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

Liu, Y., 2013, Noise reduction by vector median filtering: Geophysics, **78**, V79–V87.

Loring, T. A., 2012, Factorization of matrices of quaternions: Expositiones Mathematicae,

30, 250–267.

Mahdad, A., P. Doulgeris, and G. Blacquiere, 2011, Separation of blended data by iterative estimation and subtraction of blending interference noise: Geophysics, **76**, no. 3, Q9–Q17.

———, 2012, Iterative method for the separation of blended seismic data: Discussion on the algorithmic aspects: Geophysical Prospecting, **60**, no. 4, 782–801.

Mandic, D. P., and V. S. L. Goh, 2009, Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models: John Wiley & Sons.

Maraschini, M., R. Dyer, K. Stevens, D. Bird, and S. King, 2012, An iterative SVD method for deblending: theory and examples: SEG Technical Program Expanded Abstracts 2012, Society of Exploration Geophysicists, 1–5.

Maraschini, M., A. Kielius, and S. Grion, 2016, Rank-reduction deblending for record length extension: The example of the Carnarvon basin: SEG Technical Program Expanded Abstracts 2016, Society of Exploration Geophysicists, 4628–4632.

Maronna, R. A., R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, 2019, Robust statistics: theory and methods (with r): John Wiley & Sons.

Mataev, G., P. Milanfar, and M. Elad, 2019, DeepRED: Deep image prior powered by RED: Presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops.

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

Naghizadeh, M., and M. Sacchi, 2012, Multicomponent f-x seismic random noise attenuation via vector autoregressive operators: Geophysics, **77**, no. 2, V91–V99.

O'Brien, M. J., 2010, Mounting an offense against poor-quality shear data: SEG Technical Program Expanded Abstracts 2010, Society of Exploration Geophysicists, 177–181.

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

Ortolani, F., D. Comminiello, M. Scarpiniti, and A. Uncini, 2017, Frequency domain quaternion adaptive filters: Algorithms and convergence performance: Signal Processing, **136**, 69–80.

Peng, C., B. Liu, A. Khalil, and G. Poole, 2013, Deblending of simulated simultaneous sources using an iterative approach: an experiment with variable-depth streamer data: SEG Technical Program Expanded Abstracts 2013, SEG, 4278–4282.

Perelli, A., and M. S. Andersen, 2019, Regularized Newton sketch by denoising score matching for computed tomography reconstruction: Presented at the 2019 Workshop on Signal Processing with Adaptive Sparse Structured Representations.

Picinbono, B., 1994, On circularity: IEEE Transactions on signal processing, **42**, 3473–3482.

———, 1996, Second-order complex random vectors and normal distributions: IEEE Trans-

actions on Signal Processing, **44**, 2637–2640.

Picinbono, B., and P. Chevalier, 1995, Widely linear estimation with complex data: IEEE transactions on Signal Processing, **43**, 2030–2033.

Porsani, M. J., 1999, Seismic trace interpolation using half-step prediction filters: Geophysics, **64**, 1461–1467.

Reehorst, E. T., and P. Schniter, 2018, Regularization by denoising: Clarifications and new interpretations: IEEE transactions on computational imaging, **5**, no. 1, 52–67.

Robertsson, J. O., L. Amundsen, and Å. S. Pedersen, 2016, Signal apparition for simultaneous source wavefield separation: Geophysical Journal International, **206**, 1301–1305.

Romano, Y., M. Elad, and P. Milanfar, 2017, The little engine that could: Regularization by Denoising (RED): SIAM Journal on Imaging Sciences, **10**, no. 4, 1804–1844.

Saad, O. M., and Y. Chen, 2020, Deep denoising autoencoder for seismic random noise attenuation: Geophysics, **85**, V367–V376.

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

Sacchi, M. D., and T. J. Ulrych, 1996, Estimation of the discrete Fourier transform, a linear inversion approach: Geophysics, **61**, no. 4, 1128–1136.

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

Sangwine, S., and N. Le Bihan, 2005, Quaternion toolbox for Matlab: http://qtfm. sourceforge. net/.

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

Schreier, P. J., and L. L. Scharf, 2010, Statistical signal processing of complex-valued data: the theory of improper and noncircular signals: Cambridge university press.

Song, J., P. Li, Z. Qian, M. Zhang, P. Sun, W. Wang, and Y. Ma, 2019, Simultaneous Vibroseis data separation through sparse inversion: The Leading Edge, **38**, no. 8, 625–629.

Soubaras, R., 1994, Signal-preserving random noise attenuation by the fx projection, *in* SEG Technical Program Expanded Abstracts 1994: Society of Exploration Geophysicists, 1576–1579.

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

Spitz, S., G. Hampson, and A. Pica, 2008, Simultaneous source separation: A prediction-subtraction approach: SEG Technical Program Expanded Abstracts 2008, SEG, 2811–2815.

Stanton, A., and M. D. Sacchi, 2013, Vector reconstruction of multicomponent seismic data:

Geophysics, **78**, V131–V145.

———, 2016, Efficient geophysical research in julia: CSEG GeoConvention 2016, 1–3.

Sun, Y., J. Liu, and U. S. Kamilov, 2020, Block coordinate Regularization by Denoising: IEEE Transactions on Computational Imaging, **6**, 908–921.

Tatham, R. H., and M. D. McCormack, 1991, Multicomponent seismology in petroleum exploration: Society of Exploration Geophysicists.

To, R., J. Cai, F. C. Loh, B. Hung, and S. Wolfarth, 2018, Deblending of high-density OBN simultaneous source acquisition offshore indonesia: SEG Technical Program Expanded Abstracts 2018, SEG, 4578–4582.

Took, C. C., and D. P. Mandic, 2009, The quaternion LMS algorithm for adaptive filtering of hypercomplex processes: IEEE Transactions on Signal Processing, **57**, 1316–1327.

———, 2011, Augmented second-order statistics of quaternion random signals: Signal Processing, **91**, 214–224.

Trad, D., T. Ulrych, and M. Sacchi, 2003, Latest views of the sparse Radon transform: Geophysics, **68**, 386–399.

Treitel, S., 1974, The complex Wiener filter: Geophysics, **39**, 169–173.

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

Ulrych, T. J., and M. D. Sacchi, 2005, Information-based inversion and processing with applications: Elsevier.

Vargas, A. A. A., 2020, Regularization by denoising applied to non-linear traveltime tomography: Master's thesis, University of Alberta.

Venkatakrishnan, S. V., C. A. Bouman, and B. Wohlberg, 2013, Plug-and-play priors for model based reconstruction: 2013 IEEE Global Conference on Signal and Information Processing, IEEE, 945–948.

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

Wason, H., F. J. Herrmann, and T. T. Lin, 2011, Sparsity-promoting recovery from simultaneous data: A compressive sensing approach: SEG Technical Program Expanded Abstracts 2011, SEG, 6–10.

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

Wu, Z., Y. Sun, J. Liu, and U. Kamilov, 2019, Online Regularization by Denoising with applications to phase retrieval: Presented at the Proceedings of the IEEE International Conference on Computer Vision Workshops.

Xu, D., Y. Xia, and D. P. Mandic, 2015, Optimization in quaternion dynamic systems: Gradient, hessian, and learning algorithms: IEEE transactions on neural networks and learning systems, **27**, 249–261.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, 2005, Antileakage Fourier transform for seismic data regularization: Geophysics, **70**, V87–V95.

Xu, W., Y. Zhou, D. Liu, X. Wang, and W. Chen, 2022, Seismic intelligent deblending via Plug and Play method with blended CSGs trained deep CNN gaussian denoiser: IEEE Transactions on Geoscience and Remote Sensing, **60**, 1–13.

Yang, H. H., and Y. Hua, 1996, On rank of block hankel matrix for 2-D frequency detection and estimation: IEEE Transactions on Signal Processing, **44**, 1046–1048.

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

Zeng, F., W. Yue, and C. Li, 2018, Quaternion-based anisotropic inversion for flexural waves in horizontal transverse isotropic formations with unmatched sources: A synthetic example: Geophysics, **83**, A69–A74.

Zhang, C., and M. van der Baan, 2018, Multicomponent microseismic data denoising by 3D shearlet transform: Geophysics, **83**, A45–A51.

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

Zhao, Q., Q. Du, Q. Yasin, Q. Li, and L. Fu, 2020, Quaternion-based sparse tight frame for multicomponent signal recovery: Geophysics, **85**, V143–V156.

APPENDIX A

Efficient Formulations of the QSSA Filter

A.1 Introduction

The increased computational time in the quaternionic processing of vector signals imposes a challenge in its development because computationally expensive techniques, such as SVD might suffer more in their hypercomplex versions. However, published methods that improve rank-reduction efficacy in the case of scalar data (Gao et al., 2013; Cheng and Sacchi, 2017) can also be readily extended to the quaternion domain.

A.1.1 Lanczos Bidiagonalization for Structured Matrices

For a $m \times n$, with $m \geq n$, general matrix $\mathbf{X}$, the Lanczos bidiagonalization can be used to find a decomposition of the form

$$\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^H, \tag{A.1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of size $m \times m$ and $n \times n$, respectively, and $\mathbf{B}$ is a real upper bidiagonal matrix, of same size as $\mathbf{X}$, with entries given by

$$\mathbf{B}_{i,i} = \alpha_i, \qquad i = 1 \dots m, \qquad \mathbf{B}_{i,i+1} = \beta_i, \qquad i = 1 \dots m-1. \tag{A.2}$$

Equation A.1 can be used to find a low-rank approximation to the matrix $\mathbf{X}$ as

$$\mathbf{X}_k = \mathbf{U}_k\mathbf{B}_k\mathbf{V}_k^H, \tag{A.3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are truncated to its first k columns, and the elements in $\mathbf{B}_k$ are given by

$$\mathbf{B}_{i,i} = \alpha_i, \qquad i = 1 \dots k, \qquad \mathbf{B}_{i,i+1} = \beta_i, \qquad i = 1 \dots k-1. \tag{A.4}$$

Here k refers to the number of Lanczos iterations to be performed which, although correlated to, it is not the same as the desired rank. The columns of the unitary matrices, $\mathbf{U}_k$ and $\mathbf{V}_k$, and the diagonals of $\mathbf{B}_k$ can be computed iteratively via algorithm 5.

Algorithm 5 Lanczos bidiagonalization with selective orthogonalization

Set: $\mathbf{z}_1 \neq 0, \quad \mathbf{v}_1 = \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|}$,
$\mathbf{y}_1 = \mathbf{X}\mathbf{v}_1, \quad \alpha_1 = \| \mathbf{y}_1 \|, \quad \mathbf{u}_1 = \frac{\mathbf{y}_1}{\alpha_1}$
for $i = 1$ to $k-1$ **do**
  $\mathbf{z}_{i+1} = \mathbf{X}^H \mathbf{u}_i - \alpha_i \mathbf{v}_i$
  !Reorthogonalize
  $\mathbf{z}_{i+1} = \mathbf{z}_{i+1} - (\mathbf{z}_{i+1}^H \mathbf{v}_j)\mathbf{v}_j, \quad j = i, i-1, \dots, i-s$
  $\beta_i = \| \mathbf{z}_{i+1} \|$
  $\mathbf{v}_{i+1} = \frac{\mathbf{z}_{i+1}}{\beta_i}$
  $\mathbf{y}_{i+1} = \mathbf{X}\mathbf{v}_{i+1} - \beta_i \mathbf{u}_i$
  !Reorthogonalize
  $\mathbf{y}_{i+1} = \mathbf{y}_{i+1} - (\mathbf{y}_{i+1}^H \mathbf{u}_j)\mathbf{u}_j, \quad j = i, i-1, \dots, i-s$
  $\alpha_{i+1} = \| \mathbf{y}_{i+1} \|$
  $\mathbf{u}_{i+1} = \frac{\mathbf{y}_{i+1}}{\alpha_{i+1}}$
end for

Notice that the orthonormal vectors in $\mathbf{U}$ and $\mathbf{V}$ suffer from loss of orthogonality due to floating point arithmetic inaccuracies and have to be reorthogonalized to ensure numerical stability to the Lanczos algorithm. The Lanczos Algorithm given in 5 uses selective orthogonalization, where s refers to the number of preceding vectors used in the orthogonalization step. However, in this paper, complete orthogonalization ($j = i, \dots, 1$) was preferred since only k vectors are being computed.

The computational cost of algorithm 5 is mainly dictated by the matrix-vector multiplications. Fortunately, the structure of the trajectory matrix can be exploited to improve the efficacy of the Lanczos bidiagonalization using FFTs to perform these operations. The efficiency gain comes from the fact that the result of the product of a circulant matrix $\mathbf{C}$ by a vector $\mathbf{x}$ can be treated as the convolution of two sequences $\mathbf{c}$ and $\mathbf{x}$, where $\mathbf{c}$ denotes the first column of the circulant matrix $\mathbf{C}$. The Hankel structure of the trajectory matrix can then be exploited by reversing its columns to form a Toeplitz matrix which can be expanded into a circulant matrix. In the case of real or complex arrays, the convolution theorem allows one to perform this matrix-vector multiplication much faster and with lower memory requirements via FFTs. This efficient methodology can also be used for higher-level Hankel/Toeplitz matrices using multidimensional FFTs (Gao et al., 2013).

Quaternionic Extension

Finally, the extension of the Lanczos bidiagonalization algorithm to the quaternion domain is rather straightforward, where only the definition of the norm has to be updated to accommodate the additional components of a quaternion. In the quaternion domain, the efficient matrix-vector multiplication can be accomplished via QFTs and the spectral-domain *formulae* discussed in Chapter 2, section 2.3.4, thus allowing one to exploit the structure of multilevel block Hankel/Toeplitz matrices with quaternion entries. Furthermore, notice that in AQSSA $\mathbf{X}$ would be replaced by the augmented trajectory matrix $\mathbf{X}^a = [\mathbf{X}; \mathbf{X}^i; \mathbf{X}^j; \mathbf{X}^k]$ (equation 3.10). This matrix has a block-Hankel structure since the involution matrices are still Hankel. Therefore, the same routine for the efficient product of Hankel (or Toeplitz) matrices by vectors can be repeatedly used to perform the products $\mathbf{X}^a\mathbf{v}_{i+1}$ and $\mathbf{X}^{aH}\mathbf{u}_{i+1}$. The efficient formulation of the matrix-vector products for quaternion matrices provided a substantial gain in processing time (figure 3.8). For instance, the processing time for a 80×80 frequency slice through the QSVD with $k = 1$ is 738.94 seconds, and through Lanczos bidiagonalization with $k = 4$ iterations is 1.59 seconds.

A.1.2 Randomized QR Decomposition

Computing the rQR of a given matrix entails projecting it over a set of p random unit vectors, usually denoted by Ω, as to reduce the dimension of the initial matrix. For instance, the level-n Hankel matrix ($\mathbf{H}^{(n)}$) of size $L_1 \dots L_n \times K_1 \dots K_n$ is projected over the set of random vectors (Ω) of size $K_1 \dots K_n \times p$ as

$$\mathbf{W} = \mathbf{H}^{(n)}\Omega, \tag{A.5}$$

where $p \ll K_1 \dots K_n$. Of course, the matrix $\mathbf{W}$, of size $L_1 \dots L_n \times p$, is much smaller then $\mathbf{H}^{(n)}$, and represents a subsampling of the matrix associated with the data. The resulting matrix $\mathbf{W}$ is then decomposed via the QR decomposition

$$[\mathbf{Q}, \mathbf{R}] = \mathtt{qr}(\mathbf{W}), \tag{A.6}$$

yielding a reduced-rank orthonormal basis of $\mathbf{H}^{(n)}$, the matrix $\mathbf{Q}$ (of same size as $\mathbf{W}$). The low-rank approximation $\mathbf{H}_p^{(n)}$ is obtained by

$$\mathbf{H}_p^{(n)} = \mathbf{Q}\mathbf{Q}^H\mathbf{H}^{(n)}, \tag{A.7}$$

which is a projection of $\mathbf{H}^{(n)}$ on the orthonormal basis $\mathbf{Q}$. Notice that this is a rank-p approximation to the initial matrix, which correlates to the rank-k approximation that

would be obtained with the SVD methodology. In fact, the number of random projections p is usually interpreted as relaxation of the rank k used in SVD, as a wider range of p is still able to recover accurate low rank approximations to the given matrix (Chiron et al., 2014; Cheng and Sacchi, 2017). However, notice that the SVD provides a least-squares approximation to $\mathbf{H}^{(n)}$, and that this is the optimum result that could be obtained.

Further improvements might still be accomplished if one employs FFTs to perform the projections given in equations A.5 and A.7. The procedure refers to the matrix-vector products involving the Hankel matrix, similarly to the Lanczos bidiagonalization above. Notice that the fast Hankel matrix product has to be used p times in equation A.5, and it can also be used to perform the product $\mathbf{Q}^H\mathbf{H}^{(n)}$ in A.7 efficiently. Finally, the reader can refer to Chiron et al. (2014) or Cheng and Sacchi (2017) for a proof that equation 3.9 can also be computed through convolutions, which is efficiently carried out via FFTs.

Quaternionic Extension

The extension of the QR algorithm to quaternion matrices given by Bunse-Gerstner et al. (1989) relies on its associated complex adjoint matrix (Zhang, 1997). Its computation, however, can be obtained via the Householder transformations, as in the QSVD (Sangwine and Le Bihan, 2005). One can perform the stochastic version of the QR decomposition of quaternion matrices by following the description given earlier in this paper. For completeness, it is important to mention that there was no noticeable difference when making the random projection vectors, Ω, as real or quaternion numbers, but real numbers were preferred due to their lower computational requirements when compared with quaternions. Also, one can promptly employ the quaternionic convolution theorem given above to perform the matrix-vector products and anti-diag averaging via QFTs, similarly to the complex case.

APPENDIX B

Testing for RED Conditions

B.1 RED Conditions

As previously discussed, one needs to ensure that RED conditions are satisfied for the RED gradient to hold. Romano et al. (2017) and Reehorst and Schniter (2018) provide numerical tests to ensure a given denoiser of choice satisfies the conditions, validating its gradient. The tests are numerical and relatively simple, as discussed below.

B.1.1 Numerical Tests

For the local homogeneity condition, one can test a given denoiser by scaling the input data by $c = 1+\varepsilon$, where ε is a small perturbation here set to 0.01. The denoiser is applied to both original and scaled inputs, and the output of the original image is then scaled by the same factor c. If local homogeneity holds, a scatter plot of the outputs delivers a straight line. The FKT denoiser was tested for variable thresholding levels, where the first 99.9% of the Fourier coefficients are thresholded, followed by 80%, 60%, 40%, 20% and 0.01%. The results are shown in Figure B.1, where it is possible to see that while FKT does satisfy the local homogeneity condition except for more restrictive thresholding of 99.9% of its coefficients. The SSA filter was also tested in similar conditions, where the rank varied as 1, 2, 5 and 10, as shown in Figure B.2. Unlike the FKT, SSA fully satisfied the local homogeneity tests.

For the Jacobian symmetry condition, another numerical test was given by Reehorst and Schniter (2018). It consists in obtaining the Jacobian of the denoiser $f(\cdot)$ via a central differenciation approach, followed by a comparison of the obtained Jacobian with its transpose.

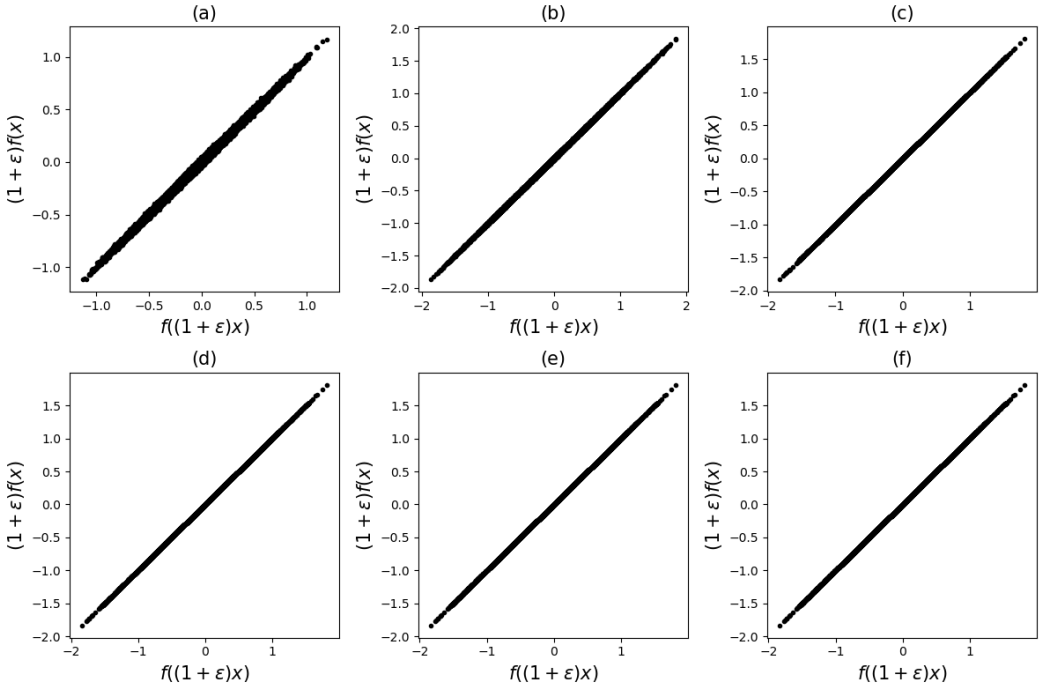

Figure B.1: Local homogeneity test results for the FKT filter with threshold percentages (a) 99.9%, (b) 80%, (c) 60%, and (d) 40%, (e) 20%, and (f) 0.01%.

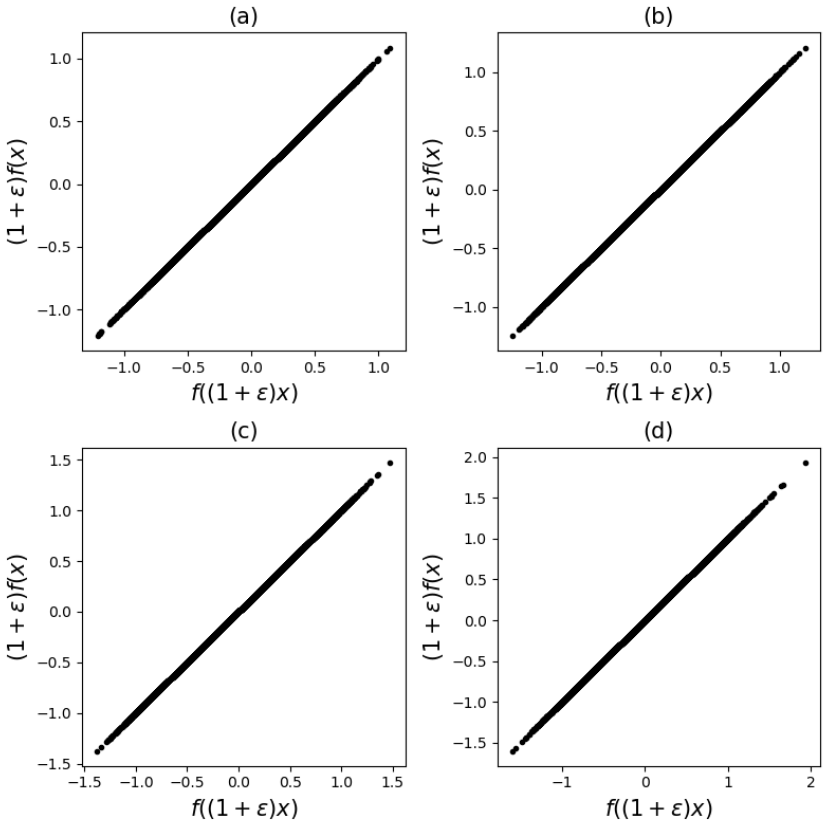

Figure B.2: Local homogeneity test results for the SSA filter with rank (a) 1, (b) 2, (c) 5, and (d) 10.

The Jacobian can be numerically approximated as

$$\nabla f(\mathbf{x})_{i,j} = \frac{f_i(\mathbf{x} + \varepsilon \mathbf{e}_j) - f_i(\mathbf{x} - \varepsilon \mathbf{e}_j)}{2\varepsilon}, \tag{B.1}$$

with ε a small perturbation and $\mathbf{e}_j$ a basis to modify the j-th sample. Its symmetry can be quantified by

$$\delta_f = \frac{\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})^T\|_F^2}{\|\nabla f(\mathbf{x})\|_F^2}. \tag{B.2}$$

Such a test is more demanding than the previous conditions since each sample j in the image to be denoised has to be perturbed. The FKT denoiser was tested with the 99.9% of thresholding values, since it failed the local homogeneity property for it. As given by Table B.1, estimated in single precision, the FKT satisfies the Jacobian symmetry condition. Moreover, Reehorst and Schniter (2018) already clarify that transform-domain thresholding denoisers satisfy the Jacobian symmetry condition analytically. The aforementioned test is thus useful for those denoisers whose symmetry one cannot easily study its symmetry, such as the SSA filter. For the latter, the Jacobian symmetry test was not positive regardless of the rank used (the reported value corresponds to rank of 3).

Jacobian symmetry

$f(\cdot)$	FKT	SSA
δ_f	0.0	0.403

Table B.1: Output results for the Jacobian symmetry tests for the FKT and SSA denoisers.

Finally, for the strong passivity condition, the test entails a power method iteration to determine the denoiser's spectral radius. For further details, the reader is pointed to Romano et al. (2017). The test should converge to a spectral radius smaller than, or equal to, 1. Both the FKT and SSA denoisers satisfied this condition using the same threshold and rank values as above.

B.1.2 The RED Gradient

By assuming a denoiser that satisfies the Jacobian symmetry condition, one can use local homogeneity to derive the RED gradient shown in equation 5.16. Recall that the denoiser applied to a positively scaled image should result in a scaled version of the original output, i.e.

$$f(c\mathbf{x}) = cf(\mathbf{x}),$$

for c close to 1. The direct implication of this property relates to the directional derivatives of $f(\cdot)$ along $\mathbf{x}$ as

$$\nabla f(\mathbf{x})\mathbf{x} = \frac{f(\mathbf{x}+\varepsilon\mathbf{x})-f(\mathbf{x})}{\varepsilon} \tag{B.3}$$

$$= \frac{f((1+\varepsilon)\mathbf{x})-f(\mathbf{x})}{\varepsilon}, \tag{B.4}$$

and by invoking the local homogeneity condition, one rewrites

$$\nabla f(\mathbf{x})\mathbf{x} = \frac{(1+\varepsilon)f(\mathbf{x})-f(\mathbf{x})}{\varepsilon} \tag{B.5}$$

$$\nabla f(\mathbf{x})\mathbf{x} = f(\mathbf{x}). \tag{B.6}$$

APPENDIX C

Software for Quaternion Signal Processing

The developments in this thesis are based on the Julia language (Bezanson et al., 2012), and it relies on the `SeismicJulia` packages (Stanton and Sacchi, 2016) for tools for reading and processing seismic data. The quaternionic development presented here is mainly based on the Quaternion Toolbox For MATLAB (QTFM) (Sangwine and Le Bihan, 2005), which is a wonderful contribution. Chapter 4 was developed using the QFTM in conjunction with the `SeismicLab` toolbox.

The QTFM is a convenient way to explore quaternion-based signal processing, but the Julia language allows the definition of abstract data types with more ease. Defining a quaternion data type that inherited many methods already defined for complex-valued data was convenient thanks to Julia's multiple dispatch. There are packages[1] that define quaternion functionalities which were useful when extending key QFTM functions to the Julia language, as the Quaternion Fourier Transform which was the main algorithmic tool used throughout the whole thesis. Finally, some examples used in this thesis for Chapters 3 and 5 can be found in https://github.com/bbahia-phd/HCDSP.

[1] Such as https://github.com/JuliaGeometry/Quaternions.jl and https://github.com/peakbook/Quaternions.jl