



**Master of Science in Applied Geophysics
Thesis**

PLANE-WAVE MARCHENKO IMAGING METHOD: APPLICATIONS

Mohammed Almobarak

August 16, 2021

PLANE-WAVE MARCHENKO IMAGING METHOD: APPLICATIONS

Master of Science Thesis
for the degree of Master of Science in Applied Geophysics at
Delft University of Technology
by

Mohammed Almobarak

August 16, 2021

Master of Science in Applied Geophysics
Delft University of Technology

Supervisor(s):

Prof. dr. ir. CPA (Kees) Wapenaar

Johno van IJsseldijk

Committee Members:

Prof. dr. ir. CPA (Kees) Wapenaar

Dr. ir. Eric Verschuur

Dr. ir. Jan Thorbecke

Dr. ir. Deyan Draganov

Johno van IJsseldijk



Copyright © 2021 by Delft University of Technology

All rights reserved.

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or by any information storage and retrieval system, without permission from this publisher.

An electronic version of this dissertation is available at

<http://repository.tudelft.nl/>.

To my beloved wife

ACKNOWLEDGMENT

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Kees Wapenaar and John van IJsseldijk, who have been more than generous with their expertise and precious time, and without whom, completing this thesis would not have been possible. A special thanks to Dr. Jan Thorbecke, who has never hesitated in providing support, guidance, and encouragement throughout this project. I also would like to thank Prof. Evert Slob and Dr. Deyan Draganov for helping me at different levels in my journey at TU Delft.

I also would like to thank my parents and my family members for supporting me during such exceptional circumstances. Finally I would like to thank Saudi Aramco for giving me the opportunity to pursue my degree.

CONTENTS

Acknowledgment	ii
List of Figures	v
Summary	ix
1 Introduction	1
2 Conventional Marchenko Method	3
2.1 Notation	3
2.2 Green's Functions	3
2.3 Focusing Functions	4
2.4 Marchenko Equations	6
2.5 Input Requirements	9
2.6 Marchenko Redatuming	10
2.7 Marchenko Imaging	10
3 Plane-Wave Marchenko Methods	13
3.1 Generation of Dipping Plane-Wave	13
3.2 Plane-Wave Marchenko Equations	14
3.3 Plane-Wave Receiver-Redatuming and Imaging	17
3.4 Plane-Wave Marchenko Imaging Workflow	18
4 Application I: Simple Model	24
4.1 Model Description	24
4.2 Imaging Results	25
5 Application II: Complex Model	27
5.1 Model Description	27
5.2 Imaging Results	28
6 Application III: Salt Model	31
6.1 Model Description	31
6.2 Imaging Results	31
7 Application IV: Field Example	36
7.1 Dataset Description	36
7.2 Pre-processing Steps	36
7.3 Imaging Results	38
8 Discussion and Conclusion	43
References	46
A Marchenko-Related Plane Wavefield	48
B $\tilde{\mathbf{G}}^{-+}$ Truncation Prior to Back-Propagation	50
C Scaling Tests For The Vøring Dataset	52
D Python Codes	55
D.1 Image Extraction for Back-Propagation by Finite Difference Extrapolation (Without Phase-Shift)	55

D.2	Image Extraction for Back-Propagation by Finite Difference Extrapolation (With Phase-Shift)	58
D.3	Cross-Correlation with Point-Source Green's Functions.	61
D.4	Image Extraction for Back-Propagation by Cross-Correlation (With Phase-Shift)	63

LIST OF FIGURES

2.1	A single-sided acquisition configuration. The white star indicates the location of a point-source at $\mathbf{x}_S = (\mathbf{x}_{H,S}, x_{3,0} = 0)$, along the acquisition surface \mathbb{S}_0 . The green rays represent the two-way reflection response, received at \mathbb{S}_0 . The light green arrays represent internal multiples.	4
2.2	The truncated medium where focusing functions are defined. It is identical to the physical medium above the focal depth \mathbb{S}_A and homogeneous below it. The red lines represent a down-going focusing function $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$. The blue lines represent an up-going focusing function $f_1^-(\mathbf{x}, \mathbf{x}_A, t)$. The white star represents a virtual source at \mathbf{x}_A , from which energy radiates as illustrated by the green lines.	5
2.3	Illustrative representation of Equation 2.15. The red lines represent the down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$. The blue lines represent the up-going focusing function $f_1^-(\mathbf{x}_S, \mathbf{x}_A, t)$. The green lines represent the down-going Green's function $G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t)$ associated with a virtual source (white star) at \mathbf{x}_A	7
2.4	Illustrative representation of Equation 2.16. The red lines represent the time-reversed down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, -t)$. The blue lines represent the time-reversed up-going focusing function $f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t)$. The orange lines represent the up-going Green's function $G^{-,-}(\mathbf{x}_R, \mathbf{x}_A, t)$ associated with a virtual source (white star) at \mathbf{x}_A . The dashed orange lines represent the overlap between the direct arrival of $f_1^+(\mathbf{x}_S, \mathbf{x}_A, -t)$ and $G^{-,-}(\mathbf{x}_R, \mathbf{x}_A, t)$	8
2.5	The reflection response $\tilde{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$, defined in the complementary truncated medium.	11
2.6	The reflection response $R_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$, defined in the physical medium.	11
3.1	(a) Modelling a dipping plane-wave by exciting horizontally-situated time-delayed point-sources. (b) Modelling a dipping plane-wave by exciting point-sources of varying vertical coordinates. The red arrows indicate the propagation direction of the plane wavefront.	14
3.2	The plane-wave focusing function is defined as the integral of all possible point-source focusing functions (blue and red lines) along the focal depth \mathbb{S}_A . Focusing is accomplished in space along the focusing depth \mathbb{S}_A , but at different times (dashed green lines).	15
3.3	The down-going plane-wave Green's function $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ associated with plane-wave virtual source at the focal depth \mathbb{S}_A	16
3.4	The up-going plane-wave Green's function $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}_A, t)$ associated with plane-wave virtual source at the focal depth \mathbb{S}_A	17
3.5	The adopted plane-wave Marchenko imaging workflow. The different approaches of performing the cross-correlation step are illustrated in Figure 3.7.	18
3.6	Examples for plane-wave Green's functions modelled via finite difference modelling for (a) angle= 15° , and (b) angle= -15° . (c), (d) are the muted versions of (a) and (b), respectively. The complex conjugates of (c) and (d) represent the initial focusing function used to initiate the Marchenko iterative scheme.	19

3.7	A workflow of the different approaches of performing the back-propagation of the down-going plane-wave Green's function $\tilde{G}^{-,+}$. The green lines indicate the back-propagation by finite difference extrapolation approach. The red lines indicate the back-propagation by cross-correlation with point-source Green's functions modelled by finite difference modelling. The blue lines indicate the back-propagation by cross-correlation with point-source Green's functions modelled with an Eikonal solver. The purple lines indicate the functions and processing step common to the latter two approaches. Note that for the first approach, the cross-correlation step is taken into account by the finite difference extrapolation process.	20
3.8	(a) The imaging condition associated with back-propagation by cross-correlation. (b) The imaging condition associated with back-propagation by finite difference extrapolation. The green line represents t_0 which is equal to the duration of the Green's function $\tilde{G}^{-,+}$ (coincides with $t=0$), used as an input to the back-propagation process. The blue line represents the maximum plane-wave time delay, $\max(\mathbf{p} \cdot \mathbf{x}_{H,B})$. The red line represents the linear plane-wave time delay, $\mathbf{p} \cdot \mathbf{x}_{H,B}$. The magenta line represents the shift of the plane-wave time delay by half the wavelength of the wavelet denoted by c	21
3.9	Schematic representation of the effect of the plane-wave incident angle α on the recorded wavelength λ_z for a horizontal receivers array.	22
3.10	(a) The imaging condition expressed in Equation 3.20, which extracts the image along a slanted line, resulting in a jittered image (c). (b) The phase-shifted imaging condition, expressed in Equation 3.23, which extracts the image along a horizontal line, resulting in a smooth and jitter-free image (d). The red arrows indicate imaging artefacts, which can be attenuated using a dip filter.	23
4.1	The velocity (a) and density (b) models used for computing the reflection response.	25
4.2	The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.	25
4.3	A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane wave image. The images are displayed using the same scaling factor. The red arrows indicate internal multiples that were suppressed using the plane-wave Marchenko imaging method. The blue arrow indicate a deep, dipping reflector that was poorly illuminated by horizontal plane-waves.	26
5.1	The velocity (a) and density (b) models used for computing the reflection response.	27
5.2	The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.	28
5.3	A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane-wave image. Both images are displayed using the same scaling factor. The red arrows indicate internal multiples that are mostly suppressed using the plane-wave Marchenko imaging method.	29
5.4	The individual Marchenko-based dipping plane-wave images associated with (a) -5° , (b) 5° , (c) -10° , (d) 10° , (e) -15° , and (f) 15° . All images are displayed using the same scaling factor. The red and green boxes indicate the location of the event discontinuities observed in the Marchenko-based horizontal plane-wave image (Figure 5.3).	29
5.5	A comparison between (a) the Marchenko-based horizontal plane-wave image, and (b) the stack of the Marchenko-based dipping plane-waves images, with incident angles ranging from -15° to 15° , inclusive, with an increment of 5° . Both images are displayed using the same scaling factor.	30
6.1	The velocity (a) and density (b) models used for computing the reflection response.	31

6.2	The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.	32
6.3	A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane-wave image. Both images are displayed using the same scaling factor. The red arrows indicate residual internal multiples that Marchenko plane-wave imaging method failed to suppress. Note the similarity in terms of character between the primary and multiple events.	33
6.4	The Marchenko-based dipping plane-wave images associated with angles (a) -10° , (b) 10° , and (c) 25° . All images are displayed using the same scaling factor. The red arrows indicate the locations where the image associated with angle 25° shows degradation of the lateral resolution.	34
6.5	The stacking results of different Marchenko-based plane-wave images. (a) the stack of images associated with angles -10° , 0° , and 10° , (b) the stack of images associated with -10° , 0° , 10° , and 25° , and (c) the stack of images associated with angles 0° , and 25° . All images are displayed using the same scaling factor. The red arrows indicate the deterioration of the lateral resolution as a results of including the image associated with angle 25°	35
7.1	(a) The macro velocity model provided by Equinor ASA for the Vøring dataset. The red box indicates the imaged area of interest. (b) A pre-processed shot gather from the middle of the 2D line.	37
7.2	The source and receiver locations of the Vøring dataset used to apply the plane-wave Marchenko imaging method. The red stars represent the shot location. The green dots represent the acquisition receiver locations. The yellow dots represent the receiver locations added by source-receiver reciprocity. The blue dots represent the dummy traces added to unify the number of traces per shot location.	38
7.3	The imaging results of the SAGA A.S. dataset over the Vøring basin, the North Sea. (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. Both images are the results of summing plane wave images with incident angles ranging from -10° to 10° with an increment of 5° . Images are scaled using the same scaling factor. The red and green boxes highlight areas that are magnified in Figure 7.4 and Figure 7.5 for a more detailed comparison.	40
7.4	Magnification of the red box in Figure 7.3. (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. The red ellipsoid and arrow indicate internal multiples that have been suppressed by the plane wave Marchenko imaging method. All images are scaled using the same scaling factor.	41
7.5	Magnification of the green box in Figure 7.3. (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. The red arrows indicate internal multiples that have been suppressed by the plane wave Marchenko imaging method. The green arrows indicate events that were recovered by suppressing destructively interfering internal multiples. All images are scaled using the same scaling factor.	41
7.6	The image stacks associated with (a) back-propagation by finite difference extrapolation, and (b) back-propagation by cross-correlation. The red boxes indicate edges of the image for which the amplitude of the cross-correlation image is higher than that of the finite difference extrapolation image. The red arrows indicate the residual free-surface multiples present in the cross-correlation image.	42
A.1	A schematic illustration of the dip of different Marchenko-related plane wavefields.	49

- B.1 The truncation of the down-going plane-wave Green's function $\tilde{G}^{-,+}$ associated with a plane-wave incident angle of 15° , using a truncation time that is (a) harsh and parallel to the first event, and (d) not parallel to the first event, and (e) not truncated. The images corresponding to (a), (c), and (e) are displayed in (b), (d), and (f), respectively. The red boxes indicate the part of the imaged events affected by the different truncation designs. 51
- C.1 The focusing functions associated with the scaling factors listed in Table C.1. The left column contains the focusing functions for $z = 1500\text{m}$, whereas the right column contains the focusing functions for $z = 3000\text{m}$. Figures (a) and (b) were computed using a scaling factor of 12, figures (c) and (d) were computed using a scaling factor of $8.5e^t$, figures (e) and (f) were computed using a scaling factor of $9e^{0.9t}$, and figures (g) and (h) were computed using a scaling factor of $7.2e^t$. The red boxes in (d) indicate anomalous amplitudes. 54

SUMMARY

Seismic imaging is often used to interpret subsurface formations. However, images obtained by conventional methods are contaminated with internal multiples. The Marchenko method provides the means to obtain multiple-free subsurface images. Due to the high computational cost of the conventional point-source Marchenko imaging method, the less expensive plane-wave Marchenko imaging method can be used to produce subsurface images along planes. This method can be repeated for different incident angles to produce images that account for the variable dip of the subsurface structures. In this work, we present the results of applying the plane-wave Marchenko imaging method to a range of datasets of different reflector geometries, including a 2D marine dataset acquired over the Vøring basin, the North Sea. The results show that, in comparison to the conventional plane-wave image, the plane-wave Marchenko imaging method successfully suppressed internal multiples, resulting in improvements in both the amplitude and continuity of the seismic events.

1

INTRODUCTION

In exploration geophysics, reflection seismology is often the preferred method to explore the Earth's subsurface. Seismic reflection data are acquired by exciting active sources at the acquisition surface, and record the reflected wavefield by a receiver array at the same acquisition surface. By applying seismic migration, such data can be used to extract images of the subsurface reflectors, which are used to identify potential prospects. A persistent problem with seismic images is the presence of internal multiples from heterogeneities in the overburden, which often distort the subsurface events, and add ambiguity to the event interpretations, especially in the absence of additional information such as VSP profiles or checkshots.

Generally, imaging involves redatuming the reflection response to the imaging depth. Redatuming is defined as the process of virtually relocating sources and receivers from the acquisition surface to a new datum inside the medium. For conventional imaging methods, such as RTM ([Berkhout, 1982](#); [Berryhill, 1984](#)), redatuming is performed by means of inverse wavefield extrapolation, which assumes that the recorded wavefield consists of primary events only, based on the first-order Born approximation. Consequently, images obtained by these redatuming methods are contaminated with internal multiple energy.

To redatum the complete wavefield, the Green's function of a virtual source needs to be obtained by either of two methods: seismic interferometry, and the Marchenko method. Source redatuming via seismic interferometry ([Bakulin and Calvert, 2006](#); [Schuster, 2009](#)) requires the presence of a source at the virtual location, and receivers surrounding the medium. While this method does not need knowledge of the medium properties, its requirements pose a practical limitation. Alternatively, solving the Marchenko equations ([Rose, 2001](#); [Wapenaar et al., 2012](#); [Broggini et al., 2014](#); [Wapenaar et al., 2014a](#)) allows for the retrieval of the virtual source Green's functions from single-sided reflection data, which represents an advantage over seismic interferometry. In addition, detailed knowledge of the medium is not needed, except for a macro velocity model. Upon obtaining the Green's functions, redatuming is achieved by either multi-dimensional deconvolution (MDD) ([Wapenaar et al., 2011](#); [Broggini et al., 2014](#)) or by double-focusing ([Wapenaar et al., 2021](#)). Then, the Marchenko-based redatumed data can be used to extract multiple-free images.

In addition to redatuming, the Marchenko methods can be used to perform virtual seismology experiments, for which the virtual response between any two points inside the medium is retrieved ([Wapenaar and Thorbecke, 2017](#)). Moreover, by projecting the virtual sources and receivers to the acquisition surface, multiples can be eliminated in the data domain without redatuming ([Zhang, 2019](#)), which can be performed as a processing step prior to imaging.

Applying the conventional point-source Marchenko imaging methods has proven to be computationally expensive as it requires imaging each lateral position at each imaging depth. Alternatively, the plane-wave Marchenko imaging method can be applied ([Meles et al., 2018](#)). This method images the subsurface along planes, reducing the imaging outputs to the number of depth levels times the number of illumination angles. Consequently, the plane wave Marchenko method

allows for obtaining images of the subsurface with less computational expense, while inheriting most of the standard Marchenko method advantages.

Following a similar approach to [Meles et al. \(2018\)](#), the plane-wave Marchenko imaging workflow is established in [chapter 3](#). In [chapter 4](#) and [chapter 5](#), the images presented in [Meles et al. \(2018\)](#) are reproduced, taking a different approach to reduce the event discontinuities observed in the horizontal plane-wave image. In [chapter 6](#) the performance of the plane-wave Marchenko imaging method is assessed, highlighting potential limitations of summing images associated with different incident angles. Then, in [chapter 7](#), the plane-wave Marchenko imaging method is applied to a field dataset acquired over the Vøring basin, the North Sea, and the results of two back-propagation approaches are compared. Finally, in [chapter 8](#) a theoretical comparison between the conventional and plane-wave Marchenko imaging methods is given, potential solutions to the plane-wave Marchenko imaging method are presented, and causes of the imaging performance differences between the plane-wave Marchenko imaging method and the Marchenko multiple elimination are highlighted.

2

CONVENTIONAL MARCHENKO METHOD

2.1 Notation

For this thesis, the coordinates vector \mathbf{x} is defined such that it refers to 3D location of a point in the Cartesian system:

$$\mathbf{x} = (\mathbf{x}_H, x_3) \quad (2.1)$$

where $\mathbf{x}_H = (x_1, x_2)$ refers to the horizontal coordinates in the x - and y -directions, and x_3 refers to the coordinate in the z -direction.

2.2 Green's Functions

Consider a single sided illumination configuration, illustrated by [Figure 2.1](#). The surface \mathbb{S}_0 divides the medium into two half-spaces: the lower half space, which is assumed to be inhomogeneous and lossless, and the upper half space which is assumed to be homogeneous, and hence it does not contribute to the recorded wavefield. In addition, \mathbb{S}_0 , is assumed to be transparent. Consequently, free-surface multiples are assumed to be absent. This configuration represents an acquisition setup, with \mathbb{S}_0 representing the surface of the earth.

With reference to the configuration described earlier, the causal Green's function $G(\mathbf{x}, \mathbf{x}_S, t)$ is defined as the causal solution of the scalar wave equation ([Born and Wolf, 1965](#); [Wapenaar et al., 2014b](#); [Wapenaar and Thorbecke, 2017](#)):

$$\rho(\mathbf{x}) \nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla G(\mathbf{x}, \mathbf{x}_S, t) \right) - \frac{1}{c(\mathbf{x})^2} \frac{\partial^2 G(\mathbf{x}, \mathbf{x}_S, t)}{\partial t^2} = -\rho(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}_S) \frac{\partial \delta(t)}{\partial t} \quad (2.2)$$

where $\rho(\mathbf{x})$ is the mass density, $c(\mathbf{x})$ is the propagation velocity, δ is the Dirac delta function, \mathbf{x}_S is the coordinate of an impulsive point source of volume injection rate at $\mathbf{x}_S = (\mathbf{x}_{H,S}, x_{3,0} = 0)$, \mathbf{x} is the observation point coordinates, anywhere in the medium. [Equation 2.2](#) states that the causal Green's function $G(\mathbf{x}, \mathbf{x}_S, t)$ is the response of an impulsive source located at point \mathbf{x}_S , and observed at point \mathbf{x} as a function of time. The causality of this Green's function dictates that the wavefield is absent for time $t < 0$. Considering the losslessness of the medium, the wave equation is symmetric in time, and hence, the time-reversed Green's function $G(\mathbf{x}, \mathbf{x}_S, -t)$ can be defined according to the Principle of time-reversal acoustics ([Fink, 1999](#)) as the wavefield propagating from the observation points back to the source location. Hence, it is only defined for time $t < 0$.

Utilizing the principle of superposition, the Green's function can be decomposed into its one-way pressure-normalized up- and down-going constituents ([Wapenaar et al., 2014b](#)):

$$G(\mathbf{x}, \mathbf{x}_S, t) = G^{+,+}(\mathbf{x}, \mathbf{x}_S, t) + G^{-,+}(\mathbf{x}, \mathbf{x}_S, t) \quad (2.3)$$

where $G^{+,+}$ refers to a Green's functions that is excited as a down-going wavefield and received as a down-going wavefield, whereas $G^{-,+}$ refers to a Green's functions that is excited as a down-going

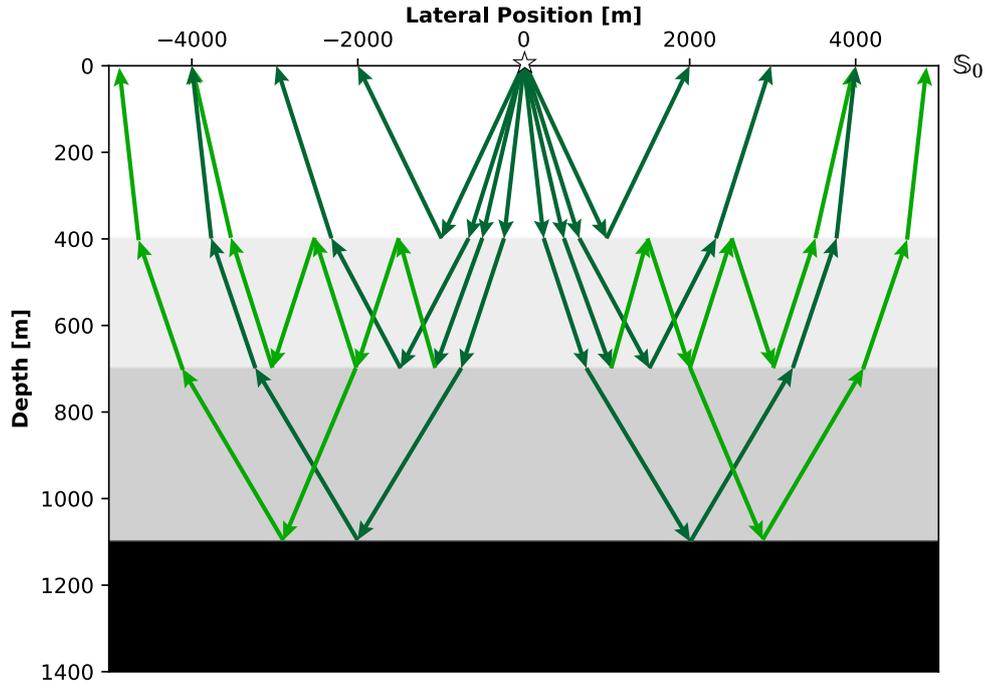


Figure 2.1: A single-sided acquisition configuration. The white star indicates the location of a point-source at $\mathbf{x}_S = (\mathbf{x}_{H,S}, x_{3,0} = 0)$, along the acquisition surface \mathbb{S}_0 . The green rays represent the two-way reflection response, received at \mathbb{S}_0 . The light green arrays represent internal multiples.

wavefield and received as an up-going wavefield. According to [Wapenaar et al. \(2017\)](#), the vertical derivative of the down-going Green's function at an observation point \mathbf{x} just below the source location \mathbf{x}_S is defined as:

$$\partial_3 G^{+,+}(\mathbf{x}, \mathbf{x}_S, t)|_{x_3=0} = -\frac{1}{2} \rho(\mathbf{x}_S) \delta(\mathbf{x}_H - \mathbf{x}_{H,S}) \frac{\partial \delta(t)}{\partial t} \quad (2.4)$$

Similarly, the vertical derivative of the up-going Green's function just below \mathbf{x}_S is defined as:

$$\partial_3 G^{-,+}(\mathbf{x}, \mathbf{x}_S, t)|_{x_3=0} = \frac{1}{2} \rho(\mathbf{x}_S) \frac{\partial R(\mathbf{x}_R, \mathbf{x}_S, t)}{\partial t} \quad (2.5)$$

where $R(\mathbf{x}_R, \mathbf{x}_S, t)$ is the reflection response of the inhomogeneous medium below the acquisition surface \mathbb{S}_0 .

2.3 Focusing Functions

In general, wavefield focusing can be achieved either by physical focusing, such as in medical imaging, or by synthetic focusing such as in exploration geophysics. Considering the linearity of the wave equation, these two approaches have similar properties. However, one main difference between the two approaches is that while physical focusing is achieved at the time of acquiring the data, synthetic focusing is performed as processing step ([Wapenaar and Thorbecke, 2017](#)). For both approaches, wavefields are emitted such that they focus at the focal point and then diverge away from it, and hence the focal point is acting as a virtual source. The diverging wavefield mimics the situation as if a physical source is placed inside the medium, while receivers are placed at the acquisition surface. Similarly, by source-receiver reciprocity, the redatumed response can be thought of as if a physical receiver is placed inside the medium, while sources are placed at the acquisition surface. Furthermore, the two scenarios can be combined such that both sources and receivers are redatumed to virtual locations inside the medium.

Classical redatuming methods, such as the work of [Berkhout \(1982\)](#), are based on the Born approximation, which assumes that the scattered wavefield consists of primary events only.

Consequently, these methods do not take into account internal multiples generated at interfaces above the focal point. As a result, the redatumed data suffers from artefacts as the focusing field creates ghost foci in addition to the virtual source. Furthermore, numerical experiments presented in the work of [Wapenaar and Thorbecke \(2017\)](#) indicate that single-sided reverse-time redatuming leads to anisotropic virtual sources, characterized by unidirectional propagation. On the contrary, the Marchenko-based focusing functions take into account internal multiples while redatuming, resulting in multiple-free redatumed data.

In the context of the Marchenko method, the focusing functions are defined in a truncated medium, which is defined as a medium that is identical to the actual medium above the focusing depth \mathbb{S}_A , and homogeneous below it ([Wapenaar et al., 2014a](#)), as shown in [Figure 2.2](#).

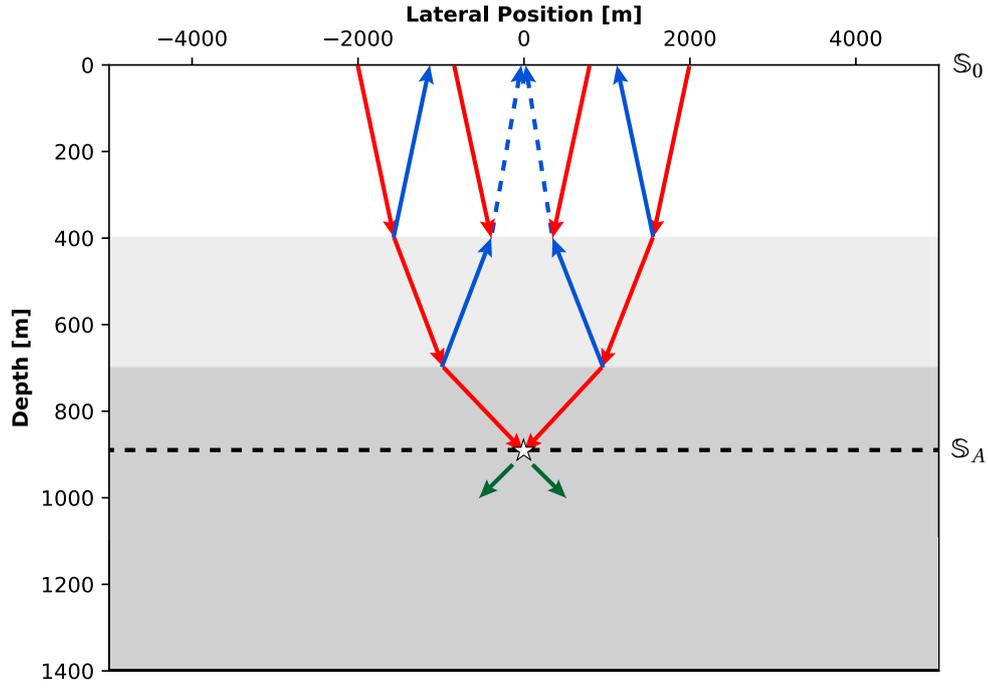


Figure 2.2: The truncated medium where focusing functions are defined. It is identical to the physical medium above the focal depth \mathbb{S}_A and homogeneous below it. The red lines represent a down-going focusing function $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$. The blue lines represent an up-going focusing function $f_1^-(\mathbf{x}, \mathbf{x}_A, t)$. The white star represents a virtual source at \mathbf{x}_A , from which energy radiates as illustrated by the green lines.

Similar to [Equation 2.3](#), [Wapenaar et al. \(2014b\)](#) showed that the two-way focusing function $f_1(\mathbf{x}, \mathbf{x}_A, t)$ can be decomposed into its pressure-normalized up- and down-going components:

$$f_1(\mathbf{x}, \mathbf{x}_A, t) = f_1^+(\mathbf{x}, \mathbf{x}_A, t) + f_1^-(\mathbf{x}, \mathbf{x}_A, t) \quad (2.6)$$

where \mathbf{x}_A is the coordinate of the focal point at $\mathbf{x}_A = (\mathbf{x}_{H,A}, x_{3,A})$ along the focal depth \mathbb{S}_A , and \mathbf{x} is the observation point coordinates anywhere in the medium. The superscript (+) refers to a down-going wavefield observed at point \mathbf{x} , and the superscript (-) refers to an up-going wavefield observed at point \mathbf{x} . The down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$ is defined as the inverse of the transmission response of the truncated medium $T(\mathbf{x}_A, \mathbf{x}_S, t)$ ([van der Neut et al., 2015](#))

$$\delta(\mathbf{x}'_{H,A} - \mathbf{x}_{H,A})\delta(t) = \int_{\mathbb{S}_0} \int_{-\infty}^t f_1^+(\mathbf{x}_S, \mathbf{x}_A, t - t') T(\mathbf{x}'_A, \mathbf{x}_S, t') dt' d\mathbf{x}_S. \quad (2.7)$$

With reference to [Figure 2.2](#), the up-going focusing functions $f_1^-(\mathbf{x}_R, \mathbf{x}_A, t)$ is defined as the response of the truncated medium to the down-going focusing functions $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$ ([van der Neut et al., 2015](#))

$$f_1^-(\mathbf{x}_R, \mathbf{x}_A, t) = \int_{\mathbb{S}_0} \int_{-\infty}^{\infty} f_1^+(\mathbf{x}_S, \mathbf{x}_A, t - t') R(\mathbf{x}_R, \mathbf{x}_S, t') dt' d\mathbf{x}_S. \quad (2.8)$$

The focusing conditions at the focal point \mathbf{x}_A are given by (Wapenaar et al., 2021):

$$f_1^+(\mathbf{x}'_A, \mathbf{x}_A, t) = \delta(\mathbf{x}'_{H,A} - \mathbf{x}_{H,A}) \delta(t), \quad (2.9)$$

$$f_1^-(\mathbf{x}'_A, \mathbf{x}_A, t) = 0. \quad (2.10)$$

Equation 2.9 states that once the injected down-going focusing function $f_1^+(\mathbf{x}'_A, \mathbf{x}_A, t)$ reaches the focal point \mathbf{x}_A , it forms an impulsive point-source, as denoted by delta functions at the right-hand-side. In addition, Equation 2.10 states that at the focal point \mathbf{x}_A , the up-going focusing functions $f_1^-(\mathbf{x}'_A, \mathbf{x}_A, t)$ is equal to zero, and hence it does not contribute to the recorded up-going wavefield. Furthermore, note that the spatial delta function at the left-hand-side of Equation 2.7 is band-limited. The band limitation is attributed to exclusion of evanescent wavefields. Further, numerical examples presented in the work of Wapenaar et al. (2014b) show that the focus could be approximated by a sinc function, with the main lobe width that is proportional to the maximum incident angle. Hence, the maximum incident angle also plays a role in the band limitation of the focus.

Wapenaar et al. (2014b) showed that the down-going focusing function $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$ can be decomposed into a direct wave and a coda

$$\begin{aligned} f_1^+(\mathbf{x}, \mathbf{x}_A, t) &= f_{1,d}^+(\mathbf{x}, \mathbf{x}_A, t) + M^+(\mathbf{x}, \mathbf{x}_A, t) \\ &= T_d^{\text{inv}}(\mathbf{x}, \mathbf{x}_A, t) + M^+(\mathbf{x}, \mathbf{x}_A, t), \end{aligned} \quad (2.11)$$

where $f_{1,d}^+(\mathbf{x}, \mathbf{x}_A, t)$ denotes the direct arrival of $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$, and $M^+(\mathbf{x}, \mathbf{x}_A, t)$ is the unknown causal coda. Taking Equation 2.7 into consideration, Equation 2.11 states that $f_{1,d}^+(\mathbf{x}, \mathbf{x}_A, t)$ is equal to the direct arrival of the inverse of the transmission response, denoted by $T_d^{\text{inv}}(\mathbf{x}, \mathbf{x}_A, t)$.

Under the direct wave assumption, T_d^{inv} can be approximated by (Wapenaar et al., 2014b):

$$T_d^{\text{inv}}(\mathbf{x}, \mathbf{x}_A, t) \approx G_d(\mathbf{x}, \mathbf{x}_A, -t), \quad (2.12)$$

where $G_d(\mathbf{x}, \mathbf{x}_A, -t)$ denotes the time-reversed direct arrival of the Green's function associated with a source at \mathbf{x}_A .

2.4 Marchenko Equations

The formulation of the Marchenko equations is based on the pressure-normalized one-way reciprocity theorems of the convolution and correlation types. The convolution type one-way reciprocity theorem is defined in the frequency domain as (Slob et al., 2014; Wapenaar et al., 2014b):

$$-\int_{\mathbb{S}_0} \rho^{-1} \{p_A^+(\partial_z p_B^-) + p_A^-(\partial_z p_B^+)\} d\mathbf{x}_0 = \int_{\mathbb{S}_A} \rho^{-1} \{(\partial_z p_A^+) p_B^- + (\partial_z p_A^-) p_B^+\} d\mathbf{x}_A, \quad (2.13)$$

where ρ is the mass density, p_A denotes the acoustic wavefield of state A, p_B denotes the acoustic wavefield of state B, \mathbb{S}_0 and \mathbb{S}_A denote the upper and lower interfaces bounding the lossless medium domain \mathbb{D} , respectively. For this type, multiplications are equivalent to convolutions in the time domain. The correlation type one-way reciprocity theorem in the frequency domain is defined as:

$$-\int_{\mathbb{S}_0} \rho^{-1} \{(p_A^+)^* (\partial_z p_B^+) + (p_A^-)^* (\partial_z p_B^-)\} d\mathbf{x}_0 = \int_{\mathbb{S}_A} \rho^{-1} \{(\partial_z p_A^+)^* p_B^+ + (\partial_z p_A^-)^* p_B^-\} d\mathbf{x}_A, \quad (2.14)$$

where $*$ denotes complex conjugation. Hence, for this type, multiplications are equivalent to cross-correlations in the time domain. These relations are valid assuming that sources are absent within the medium domain \mathbb{D} , and the medium properties between \mathbb{S}_0 and \mathbb{S}_A are identical for states A and B. Moreover, in Equation 2.14 evanescent waves are neglected.

State A is substituted with the focusing function such that $p_A^\pm(\mathbf{x}, \omega) = f_1^\pm(\mathbf{x}, \mathbf{x}_A, \omega)$, which, as mentioned in section 2.3, is defined in the truncated medium. The up-going component of state A

is absent at the lower boundary, hence $p_A^- = 0$ at \mathbb{S}_A . State B is substituted with Green's functions such that $p_B^\pm(\mathbf{x}, \omega) = G^{\mp, \pm}(\mathbf{x}, \mathbf{x}_A, \omega)$, and is defined in the actual medium. Making these substitutions leads to the Green's function representations, which are expressed after applying the inverse Fourier transform as (Slob et al., 2014; Wapenaar et al., 2014b):

$$G^{\mp, +}(\mathbf{x}_R, \mathbf{x}_A, t) + f_1^-(\mathbf{x}_R, \mathbf{x}_A, t) = \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^+(\mathbf{x}_S, \mathbf{x}_A, t') dt', \quad (2.15)$$

$$G^{\mp, -}(\mathbf{x}_R, \mathbf{x}_A, t) + f_1^+(\mathbf{x}_R, \mathbf{x}_A, -t) = \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t') dt'. \quad (2.16)$$

From Equation 2.15 and Equation 2.16, It can be seen that these representations provide a relationship between the Green's functions, the focusing functions and the reflection data. With reference to Figure 2.3, Equation 2.15 states that the reflection response to the down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$ consists of the up-going focusing function $f_1^-(\mathbf{x}_R, \mathbf{x}_A, t)$, and the down-going Green's function $G^{\mp, +}(\mathbf{x}_R, \mathbf{x}_A, t)$. Similarly, Equation 2.16 states that the reflection response of the actual medium to the time-reversed up-going focusing function $f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t)$ consists of the time-reversed down-going focusing function $f_1^+(\mathbf{x}_R, \mathbf{x}_A, -t)$ and the up-going Green's function $G^{\mp, -}(\mathbf{x}_R, \mathbf{x}_A, t)$, as illustrated by Figure 2.4. Note that the Green functions $G^{\mp, -}(\mathbf{x}_R, \mathbf{x}_A, t)$ and $G^{\mp, +}(\mathbf{x}_R, \mathbf{x}_A, t)$ represent the up- and down-going wavefields associated with a virtual source at \mathbf{x}_A , respectively.

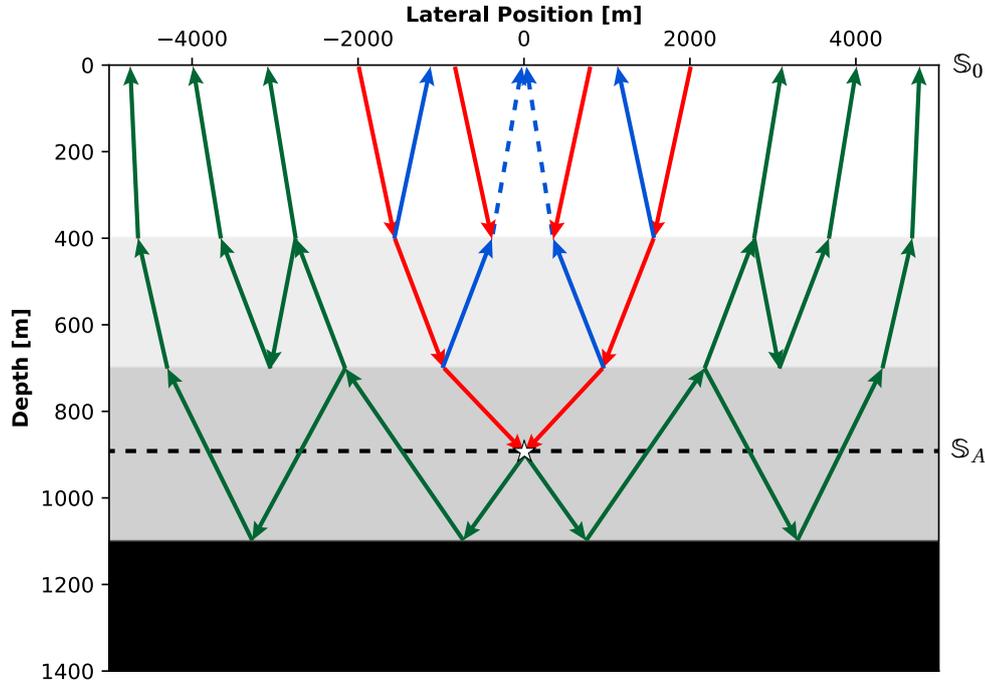


Figure 2.3: Illustrative representation of Equation 2.15. The red lines represent the down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$. The blue lines represent the up-going focusing function $f_1^-(\mathbf{x}_R, \mathbf{x}_A, t)$. The green lines represent the down-going Green's function $G^{\mp, +}(\mathbf{x}_R, \mathbf{x}_A, t)$ associated with a virtual source (white star) at \mathbf{x}_A .

Equation 2.15 and Equation 2.16 form an under-determined system of equations, which consists of two equations and four unknowns. This system cannot be solved directly, and hence an additional constraint is needed. The constraint is based on the idea that Green's and focusing functions have different causality conditions (Broggini et al., 2014; Wapenaar et al., 2014b). Introducing the time window $\Theta(\mathbf{x}_R, \mathbf{x}_A, t)$ (Thorbecke et al., 2017; Wapenaar et al., 2021):

$$\Theta(\mathbf{x}_R, \mathbf{x}_A, t) = \theta(t_d - \epsilon - t), \quad (2.17)$$

where $\theta(t)$ is the Heaviside step function, t_d is the direct travel time of the transmission response from the focusing point \mathbf{x}_A to points \mathbf{x}_R at the surface \mathbb{S}_0 , and ϵ denotes a constant that accounts

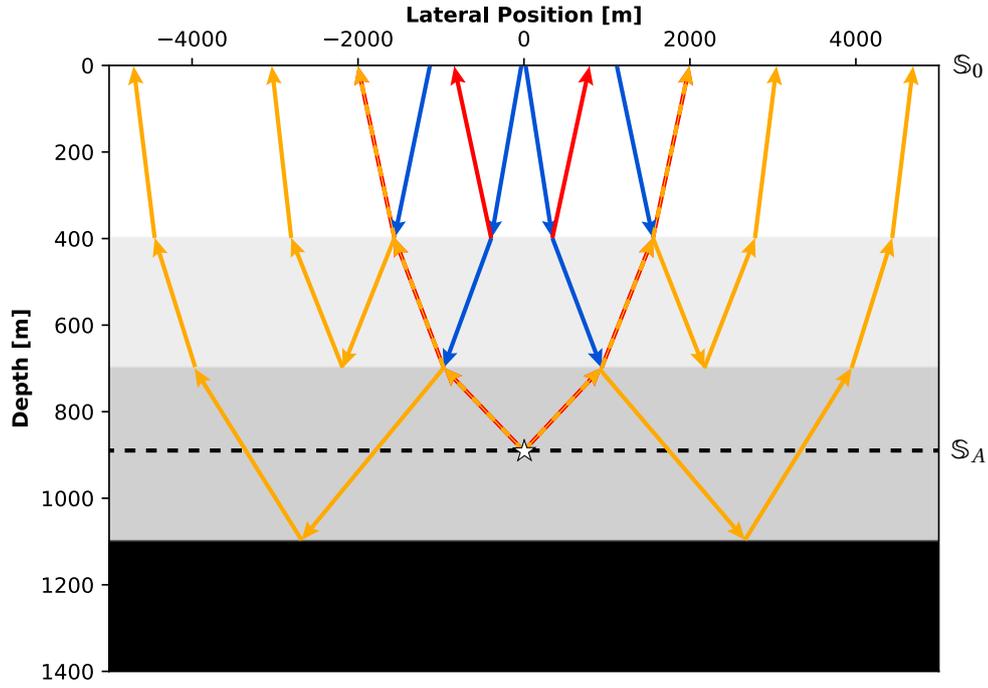


Figure 2.4: Illustrative representation of Equation 2.16. The red lines represent the time-reversed down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, -t)$. The blue lines represent the time-reversed up-going focusing function $f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t)$. The orange lines represent the up-going Green's function $G_d^-(\mathbf{x}_R, \mathbf{x}_A, t)$ associated with a virtual source (white star) at \mathbf{x}_A . The dashed orange lines represent the overlap between the direct arrival of $f_1^+(\mathbf{x}_S, \mathbf{x}_A, -t)$ and $G_d^-(\mathbf{x}_R, \mathbf{x}_A, t)$.

for the wavelength of the wavelet. This time window is designed to mute all events of the Green's functions $G_d^-(\mathbf{x}_R, \mathbf{x}_A, t)$ and $G_d^+(\mathbf{x}_R, \mathbf{x}_A, t)$ (van der Neut et al., 2015; Wapenaar et al., 2021).

Applying Equation 2.17 to Equation 2.15 and Equation 2.16 yields:

$$0 = \Theta \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^+(\mathbf{x}_S, \mathbf{x}_A, t') dt' - f_1^-(\mathbf{x}_R, \mathbf{x}_A, t) \right\}, \quad (2.18)$$

$$0 = \Theta \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t') dt' - f_1^+(\mathbf{x}_R, \mathbf{x}_A, -t) \right\}. \quad (2.19)$$

It can be seen that the Green's functions on the left-hand-side of Equation 2.18 and Equation 2.19 vanish as a result of applying Θ . Considering the relations expressed in Equation 2.11 and Equation 2.12, it can be deduced that the direct wave of the time reversed down-going focusing function $f_{1,d}^+(\mathbf{x}_R, \mathbf{x}_A, -t)$ overlaps with the direct arrival of the down-going Green's function $G_d^-(\mathbf{x}_R, \mathbf{x}_A, t)$. Hence, it is also muted by Θ . Consequently, the muted representations are defined as:

$$f_1^-(\mathbf{x}_R, \mathbf{x}_A, t) = \Theta \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^+(\mathbf{x}_S, \mathbf{x}_A, t') dt', \quad (2.20)$$

$$M_1^+(\mathbf{x}_R, \mathbf{x}_A, -t) = \Theta \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t') dt'. \quad (2.21)$$

Equation 2.20 and Equation 2.21 are called the coupled Marchenko equations. They form a well-determined system of equations that can be solved iteratively. The iterative scheme is initiated by setting the coda $M_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$ on the right-hand-side of Equation 2.20 to zero, and use $G_d(\mathbf{x}_S, \mathbf{x}_A, -t)$ as an initial estimation of the down-going focusing function $f_1^+(\mathbf{x}_S, \mathbf{x}_A, t)$. The resulting $f_1^-(\mathbf{x}_R, \mathbf{x}_A, t)$ is then used to solve Equation 2.21 to find the first estimation of $M_1^+(\mathbf{x}_R, \mathbf{x}_A, -t)$, which is, in turn, used to update $f_1^+(\mathbf{x}_R, \mathbf{x}_A, t)$. This iterative system continues until convergence is reached. Considering that the iterative scheme is energy-minimizing (Broggini

et al., 2014), the convergence of the iterative scheme is defined by a continuous decrease in the energy of the estimated focusing functions with respect to the first iteration (Thorbecke et al., 2017).

Upon obtaining the focusing functions f_1^\pm , the Green's functions $G^{-,\pm}$ can be obtained by applying the complementary time window, defined as (Wapenaar et al., 2021):

$$\Psi(\mathbf{x}_R, \mathbf{x}_A, t) = 1 - \Theta(\mathbf{x}_R, \mathbf{x}_A, t) \quad (2.22)$$

By definition, $\Psi(\mathbf{x}_R, \mathbf{x}_A, t)$ mutes the focusing functions $f_1^\pm(\mathbf{x}_R, \mathbf{x}_A, t)$, while preserving the Green's functions, such that:

$$G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t) = \Psi \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') f_1^+(\mathbf{x}_S, \mathbf{x}_A, t') dt', \quad (2.23)$$

$$G^{-,-}(\mathbf{x}_R, \mathbf{x}_A, t) + f_{1,d}^+(\mathbf{x}_R, \mathbf{x}_A, -t) = \Psi \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') f_1^-(\mathbf{x}_S, \mathbf{x}_A, -t') dt'. \quad (2.24)$$

Note that the presence of $f_{1,d}^+(\mathbf{x}_R, \mathbf{x}_A, -t)$ on the left-hand-side of Equation 2.24 is due to its coincidence with the first arrival of the up-going Green's function $G^{-,-}(\mathbf{x}_R, \mathbf{x}_A, t)$, as illustrated by the dashed orange lines in Figure 2.4. The obtained Green's functions are wavefields associated with a virtual source at the focal point \mathbf{x}_A , redatumed from its physical location at \mathbf{x}_S , and received at \mathbf{x}_R along the acquisition surface \mathbb{S}_0 . In section 2.6, methods used to achieve receiver redatuming will be presented.

2.5 Input Requirements

The Marchenko iterative scheme requires two inputs: single-sided reflection data and a macro velocity model. To assure the convergence of the Marchenko iterative scheme and to avoid artefacts, a number of reflection data conditions need to be fulfilled. These conditions can be divided into acquisition requirements and pre-processing requirements.

In terms of the acquisition requirements, sources and receivers have to be co-located, which requires them to be situated at the same spatial position. The co-location condition arises from the fact that focusing functions on the left-hand-side of Equation 2.15, which are observed at the receiver locations, are used to compute the focusing functions at the right-hand-side of Equation 2.16, observed at the source locations. This mutual usage of source and receiver coordinates is only applicable if sources and receivers are located at the same coordinates. Furthermore, the spatial sampling should be small enough to avoid spatial aliasing, and the horizontal offset should be large enough to allow for a proper reconstruction of events in the subsurface.

In terms of processing, the reflection response should be free of surface multiples and ghost effects. This is based on the assumption that the medium is bounded by a transparent upper boundary \mathbb{S}_0 . In addition, the source wavelet has to be deconvolved resulting in a zero-phased and broad-band dataset. Furthermore, undesired signal such as noise, direct arrival, and refracted waves should be attenuated.

The amplitude accuracy of the reflection data plays an important role in terms of the convergence of the Marchenko iterative scheme. Thorbecke et al. (2017) showed that introducing a scaling factor b as an error in the reflection data $R(\mathbf{x}_R, \mathbf{x}_S, t')$ would lead to an exponential error in $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$, which grows by a factor of b^{2m+2} . Here m is the iteration number. Consequently, the iterative scheme would not converge. In addition, introducing a scaling factor in $G_d(\mathbf{x}, \mathbf{x}_A, t)$ would translate into an error in $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$ with the same scaling factor error as in $G_d(\mathbf{x}, \mathbf{x}_A, t)$. Hence, the error does not amplify as the number of iterations increases. In this case, the iterative scheme might converge, but it would generate inaccurate amplitude of the calculated $f_1^+(\mathbf{x}, \mathbf{x}_A, t)$ functions. Based on this analysis, other scaling factors such as spherical divergence and frequency-dependent absorption factors should be accounted for. Note that, regardless of the

accuracy of the Marchenko method inputs, a scaling error is always embedded as a result of adopting the direct wave assumption (Equation 2.12), which implies that transmission losses are ignored (Wapenaar et al., 2014b), which results in global and offset-dependent errors (Thorbecke et al., 2017). However, scaling errors can be corrected for while re-datuming, as will be described in section 2.6.

With reference to direct wave assumption described in section 2.3, the availability of a macro velocity model is necessary to initiate the Marchenko iterative scheme, as it is used to model the initial focusing functions $f_{1,d}^+(\mathbf{x}_R, \mathbf{x}_A, t)$.

2.6 Marchenko Redatuming

As explained in section 2.4, the Green's functions obtained by solving the Marchenko equation are the wavefields obtained as a result of redatuming the physical source to a virtual location. Receiver redatuming, on the other hand, can be achieved by applying either of two methods: multi-dimensional deconvolution (MDD), or double-focusing (Broggini et al., 2014; Wapenaar et al., 2014b).

The up- and down-going Green's functions, $G^{-,\pm}$, are related to the reflection response by (Wapenaar et al., 2014b):

$$G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t) = - \int_{\mathbb{S}_A} d\mathbf{x}_B \int_0^t G^{-,-}(\mathbf{x}_R, \mathbf{x}_B, t') \bar{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t-t') dt', \quad (2.25)$$

where $\bar{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$ is the reflection response of the complementary truncated medium, which is defined as identical to the physical medium below the focal depth \mathbb{S}_A and homogeneous above it, and the arguments \mathbf{x}_A and \mathbf{x}_B are the coordinates of the virtual sources, both at \mathbb{S}_A . By definition, $\bar{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$ does not contain internal multiples from the overburden, as illustrated in Figure 2.5. However, it does contain internal multiples from the medium below \mathbb{S}_A . $\bar{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$ is obtained by applying multi-dimensional deconvolution (MDD), which requires inverting Equation 2.25. The inverse problem involves dividing the Green's functions in the space-frequency domain, which consequently eliminates any scaling errors that are common to both functions (Minato et al., 2013). However, the inversion process is not always stable, which poses a limitation of this redatuming method.

The double focusing redatuming method is defined as (Wapenaar and Thorbecke, 2017; van der Neut et al., 2017):

$$R_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t) = \int_{\mathbb{S}_0} d\mathbf{x}_R \int_0^\infty f_1^+(\mathbf{x}_R, \mathbf{x}_B, t-t') G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t') dt', \quad (2.26)$$

where R_{tar} is the reflection response below the focusing depth level, \mathbb{S}_A . Equation 2.26 describes how the down-going focusing functions $f_1^+(\mathbf{x}_R, \mathbf{x}_B, t)$ is used to redatum the receiver side of the down-going Greens function $G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t)$ obtained from Equation 2.23. Unlike the reflection response expressed in Equation 2.25 which is defined in the complementary truncated medium, $R_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$ is defined in the physical medium. Therefore, it includes events from the the overburden, above the focusing depth level \mathbb{S}_A , in addition to the events from below it. See Figure 2.6. The presence of the events from the overburden is a disadvantage of this method in comparison to MDD. However, unlike MDD, double focusing does not involve inversion, and hence it offers a more suitable alternative, suitable for practical applications. Upon applying receiver redatuming, multiple-free subsurface images can be extracted, which will be described in detail in section 2.7.

2.7 Marchenko Imaging

Conventional seismic imaging techniques, such as Reverse-Time Migration (RTM), are based on the Born approximation. Hence, they assume that the recorded wavefield consists of primary

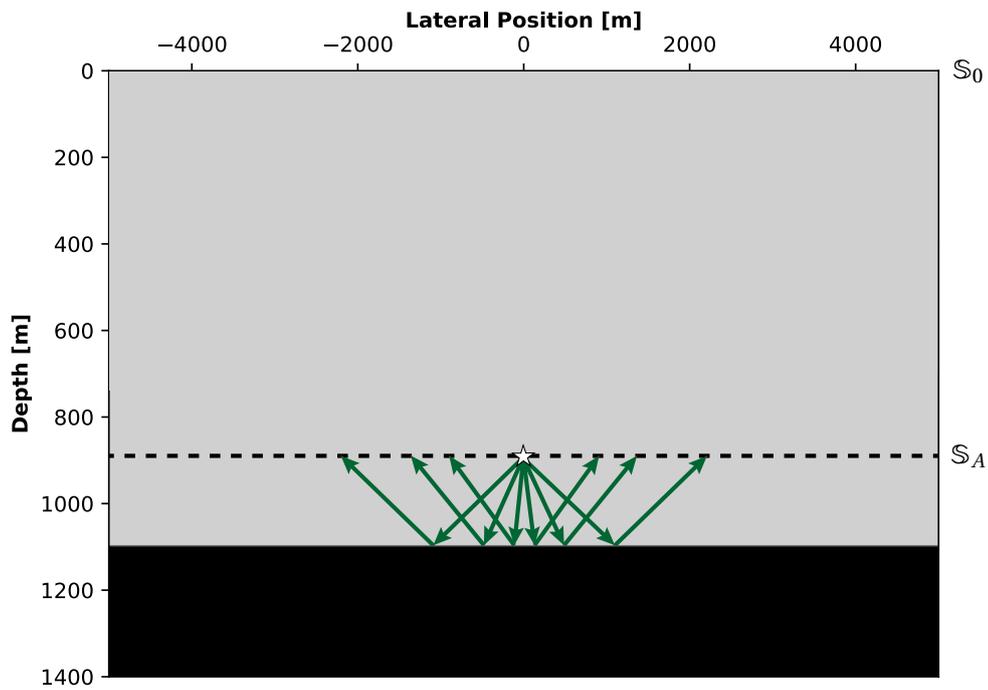


Figure 2.5: The reflection response $\bar{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$, defined in the complementary truncated medium.

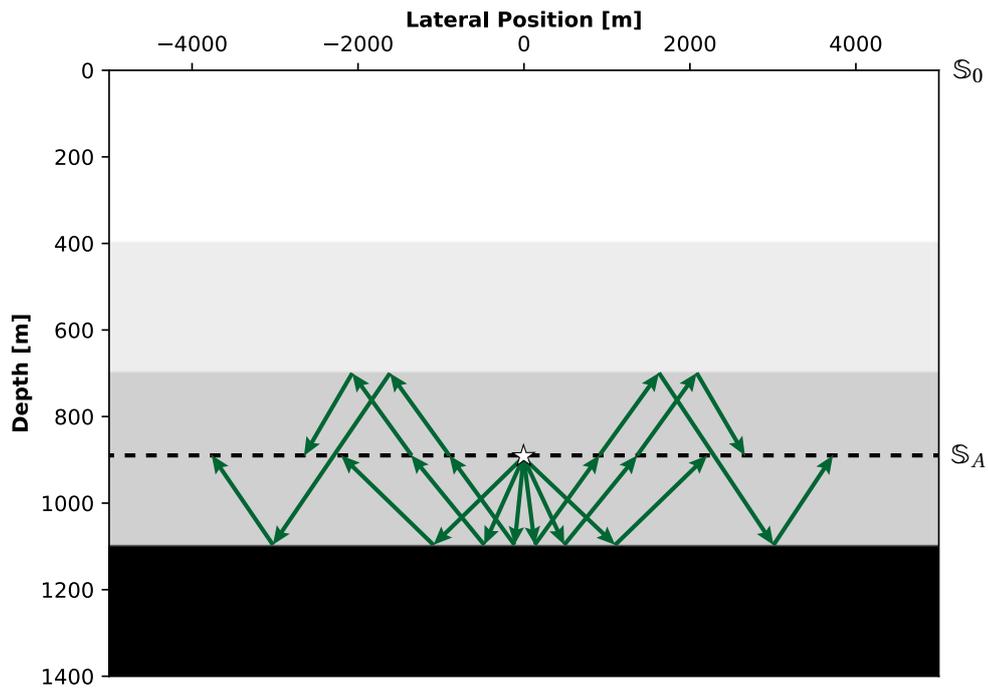


Figure 2.6: The reflection response $R_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$, defined in the physical medium.

events only. Consequently, subsurface images produced by these techniques are contaminated with internal multiples (e.g. [Berkhout, 1982](#)). On the contrary, the Marchenko method accounts for internal multiples while redatuming the reflection data, and hence the Marchenko-based images are multiple-free, which provide an advantage over the conventional methods.

Other imaging techniques such as full waveform inversion ([Virieux and Operto, 2009](#)), and full waveform migration ([Davydenko and Verschuur, 2017](#)) also provide multiples-free images. For these methods, imaging is performed recursively, while the Marchenko-based imaging is performed non-recursively, allowing for the construction of a subsurface image between any two

depth levels without the need of resolving the overburden. Hence, it provides another advantage for the Marchenko-based imaging methods.

Marchenko-based imaging can be achieved either by multi-dimensional deconvolution (MDD), or by cross correlation (Broggini et al., 2014; Wapenaar et al., 2014b). With reference to section 2.6, imaging by MDD is performed by extracting the the zero-offset ($\mathbf{x}_B = \mathbf{x}_A$) and zero-time $t = 0$ component from the MDD redatumed reflection response $\tilde{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{x}_A, t)$. This procedure is repeated for all lateral positions and for all depths within the target zone.

Imaging by cross-correlation is performed by cross correlating the up- and down-going Green's functions, $G^{\pm,+}$:

$$C(\mathbf{x}_A, \mathbf{x}_A, t) = \int_{\mathbb{S}_0} d\mathbf{x}_R \int_{-\infty}^{\infty} G^{+,+}(\mathbf{x}_A, \mathbf{x}_R, t + t') G^{-,+}(\mathbf{x}_R, \mathbf{x}_A, t') dt', \quad (2.27)$$

where $C(\mathbf{x}_A, \mathbf{x}_A, t)$ is cross correlation function. The image is then constructed by evaluating $C(\mathbf{x}_A, \mathbf{x}_A, t)$ at $t = 0$. Similar to imaging by MDD, this procedure is repeated for every lateral position and every depth level within the target zone. This is equivalent to extracting the zero-offset ($\mathbf{x}_B = \mathbf{x}_A$) and zero-time ($t = 0$) from the R_{tar} , redatumed by double-focusing (Equation 2.26).

Although both MDD- and cross-correlation-based images are free of multiples, they differ in terms of amplitude accuracy of the imaged events. Numerical examples presented in the work of Broggin et al. (2014) show that MDD-based images are more accurate both in terms of amplitude and phase, in comparison to cross-correlation-based images. The accuracy of the MDD-based images is attributed to the fact that the inversion process eliminates errors in the computed Green's functions $G^{\pm,+}$, including the transmission effects.

3

PLANE-WAVE MARCHENKO METHODS

Applying the standard point-source Marchenko method has proven to be computationally expensive as it requires computing the Marchenko solutions for each lateral position at each imaging depth. Alternatively, the plane-wave Marchenko method can be applied (Meles et al., 2018). This method bridges the RTM-based plane-wave redatuming method (see the work of Rietveld et al. (1992)) with the Marchenko method, such that it uses the Marchenko scheme outputs to redatum plane-wave sources to their virtual depths inside the medium. The main advantage of the plane-wave Marchenko method is that it reduces the Marchenko outputs to the number of depth levels times the number of illumination angles. Consequently, the plane wave Marchenko method allows for obtaining images of the subsurface with less computational expense, while inheriting most of the standard Marchenko method advantages.

3.1 Generation of Dipping Plane-Wave

Generally, dipping plane waves can be generated by two methods. The first method is by exciting horizontally situated point sources with a linear time delay between consecutive sources. Consequently, the wave-front forms a dipping plane wave (see Figure 3.1a). This method is similar to ultra-sound steering in medical imaging. The time delay for a 3D setting is given by:

$$t_{\text{delay}} = \mathbf{p} \cdot \mathbf{x}_H \quad (3.1)$$

where \mathbf{x}_H denotes the horizontal coordinates of the point sources, $\mathbf{p} = (p_1, p_2)$, p_1 and p_2 denote the horizontal ray-parameters, which are defined as (Wapenaar et al., 2021):

$$p_1 = \frac{\sin(\alpha) \cos(\beta)}{c} \quad (3.2)$$

$$p_2 = \frac{\sin(\alpha) \sin(\beta)}{c} \quad (3.3)$$

where α is the dip angle, β is the azimuth angle, and c is the propagation velocity. For horizontal plane-waves, $\alpha = 0$ and $\beta = 0$. Hence, $t_{\text{delay}} = 0$, and all point-sources are excited at the same time. The second method for generating a dipping plane wave is by exciting point sources with vertically varying coordinates at the same time (see Figure 3.1b). As the point sources are situated along a dipping plane, a time delay is not needed. For horizontal plane-waves, the point sources are situated along a horizontal plane. For the rest of this thesis, plane-waves are generated with horizontally situated and time delayed point sources.

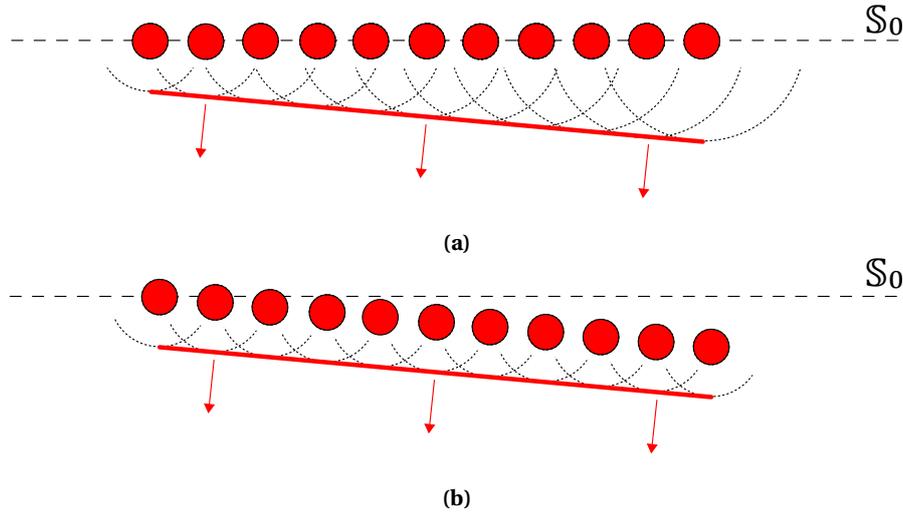


Figure 3.1: (a) Modelling a dipping plane-wave by exciting horizontally-situated time-delayed point-sources. (b) Modelling a dipping plane-wave by exciting point-sources of varying vertical coordinates. The red arrows indicate the propagation direction of the plane wavefront.

3.2 Plane-Wave Marchenko Equations

The plane-wave Marchenko focusing functions, $\tilde{f}_1^\pm(\mathbf{x}, \mathbf{p}_A, t)$, are defined in the truncated medium as (Meles et al., 2018; Wapenaar et al., 2021):

$$\tilde{f}_1^\pm(\mathbf{x}, \mathbf{p}_A, t) = \int_{\mathbb{S}_A} f_1^\pm(\mathbf{x}, \mathbf{x}_A, t - \mathbf{p} \cdot \mathbf{x}_{H,A}) d\mathbf{x}_A, \quad (3.4)$$

where the tilde ($\tilde{\cdot}$) denotes a plane wavefield, \mathbf{p}_A is the short notation of $(\mathbf{p}, x_{3,A})$, $\mathbf{p} = (p_1, p_2)$, p_1 and p_2 are the horizontal ray-parameters. Equation 3.4 states that the plane-wave focusing functions $\tilde{f}_1^\pm(\mathbf{x}, \mathbf{p}_A, t)$ are defined as the integral of all point source focusing functions $f_1^\pm(\mathbf{x}, \mathbf{x}_A, t)$ along the focusing plane \mathbb{S}_A . For an arbitrary incident angle, the plane-wave focusing conditions at the focal depth \mathbb{S}_A is defined as (Meles et al. (2018), Wapenaar et al. (2021)):

$$\begin{aligned} \tilde{f}_1^+(\mathbf{x}'_A, \mathbf{p}_A, t) &= \int_{\mathbb{S}_A} \delta(\mathbf{x}'_{H,A} - \mathbf{x}_{H,A}) \delta(t - \mathbf{p} \cdot \mathbf{x}_{H,A}) d\mathbf{x}_A \\ &= \delta(t - \mathbf{p} \cdot \mathbf{x}'_{H,A}). \end{aligned} \quad (3.5)$$

$$\tilde{f}_1^-(\mathbf{x}'_A, \mathbf{p}_A, t) = 0 \quad (3.6)$$

Equation 3.5 states that, unlike the point-source focusing condition, the plane wave focusing condition is defined by a temporal component only. Furthermore, Equation 3.5 indicates that for dipping plane-waves, focusing is accomplished in space along the focal depth \mathbb{S}_A , but at different focusing times, as indicated by green dashed lines in Figure 3.2. For horizontal plane-waves, focusing occurs along the focal depth \mathbb{S}_A at the same time ($t = 0$).

Similar to Equation 3.4, the plane-wave Green's functions are defined in the physical medium as the integration of all point-source Green's functions along the focal depth \mathbb{S}_A (Meles et al. (2018), Wapenaar et al. (2021)), satisfying

$$\tilde{G}^{-,\pm}(\mathbf{x}_R, \mathbf{p}_A, t) = \int_{\mathbb{S}_A} G^{-,\pm}(\mathbf{x}_R, \mathbf{x}_A, t - \mathbf{p} \cdot \mathbf{x}_{H,A}) d\mathbf{x}_A. \quad (3.7)$$

Here, $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ is interpreted as the response to a plane-wave source, excited as a down-going wave at the focal depth \mathbb{S}_A , and received as an up-going wave at point \mathbf{x}_R on the acquisition surface \mathbb{S}_0 (Figure 3.3). Likewise, $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}_A, t)$ is interpreted as the response to a plane-wave

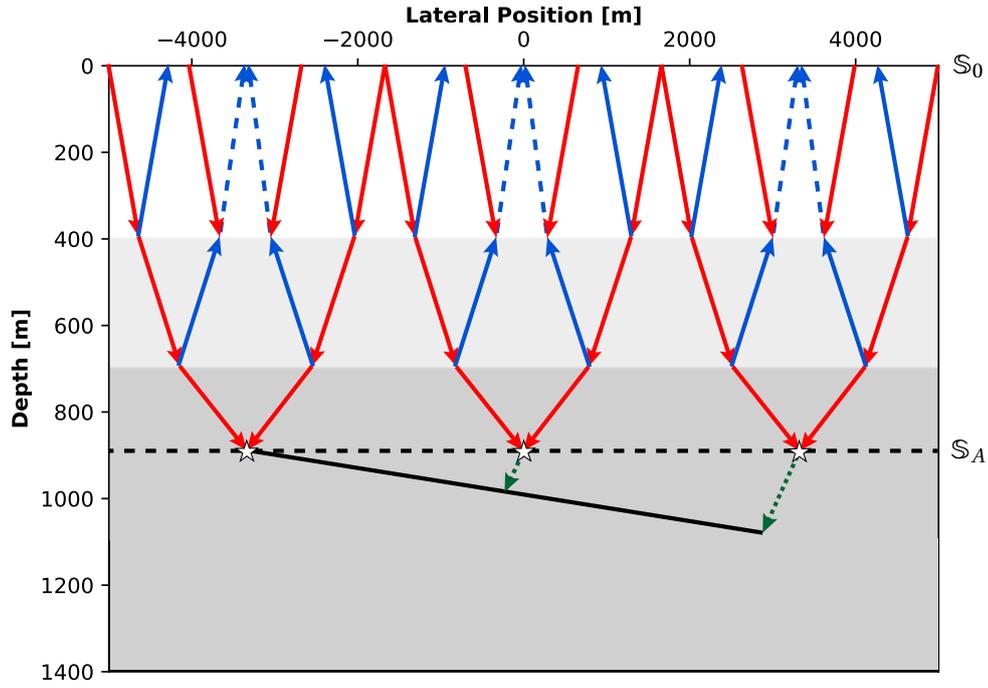


Figure 3.2: The plane-wave focusing function is defined as the integral of all possible point-source focusing functions (blue and red lines) along the focal depth \mathbb{S}_A . Focusing is accomplished in space along the focusing depth \mathbb{S}_A , but at different times (dashed green lines).

source, excited as an up-going wave at the focal depth \mathbb{S}_A , and received as an up-going wave at point \mathbf{x}_R on the acquisition surface \mathbb{S}_0 (Figure 3.4).

The plane-wave Green's function representations are defined as (Meles et al., 2018; Wapenaar et al., 2021):

$$\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t) + \tilde{f}_1^{-}(\mathbf{x}_R, \mathbf{p}_A, t) = \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') \tilde{f}_1^{+}(\mathbf{x}_S, \mathbf{p}_A, t') dt', \quad (3.8)$$

$$\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t) + \tilde{f}_1^{+}(\mathbf{x}_R, \mathbf{p}_A, -t) = \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t - t') \tilde{f}_1^{-}(\mathbf{x}_S, \mathbf{p}_A, -t') dt', \quad (3.9)$$

where \mathbf{p}'_A denotes $(-\mathbf{p}, x_{3,A})$.

To obtain the focusing functions, the causality difference between the Green's and the focusing functions is exploited by applying the plane-wave time windows $\tilde{\Theta}_{a,b}$ (Wapenaar et al., 2021) according to

$$\tilde{\Theta}_a(\mathbf{x}_R, \mathbf{p}_A, t) = \theta(\tilde{t}_d - t - \epsilon), \quad (3.10)$$

$$\tilde{\Theta}_b(\mathbf{x}_R, \mathbf{p}_A, t) \approx \theta(\tilde{t}_d - 2(\mathbf{p} \cdot \mathbf{x}_R) - \epsilon - t), \quad (3.11)$$

where $\theta(t)$ is the Heaviside step function, \tilde{t}_d is the direct travel time of the plane-wave transmission response between \mathbb{S}_A and \mathbb{S}_0 . Equation 3.11 indicates that the time window associated with the odd iterations is approximated by shifting the time window associated with even iterations (Equation 3.10) by a factor of $2(\mathbf{p} \cdot \mathbf{x}_R)$. This approximation is not valid for models with severe lateral heterogeneities. However, it is used to avoid modelling \tilde{t}_d for even and odd iterations independently. The inclusion of the term $2(\mathbf{p} \cdot \mathbf{x}_R)$ is attributed to the plane-wave modelling implementation, with the plane-wave dip in time starting at one side of the subsurface model rather than at the middle of the model. Furthermore, Equation 3.10 and Equation 3.11 suggest that the time window is different for even and odd iteration, and for positive and negative incident angles. The plane-wave time windows $\tilde{\Theta}_{a,b}$ are designed to mute the Green's functions while preserving the focusing functions, with the exception of the direct wave of the down-going

focusing function, \tilde{f}_1^+ , which overlaps with the direct arrival of $\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, -t)$. Applying $\tilde{\Theta}_{a,b}$ to Equation 3.8 and Equation 3.9 gives:

$$\tilde{f}_1^-(\mathbf{x}_R, \mathbf{p}_A, t) = \tilde{\Theta}_a \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') \tilde{f}_1^+(\mathbf{x}_S, \mathbf{p}_A, t') dt' \right\}, \quad (3.12)$$

$$\tilde{M}_1^+(\mathbf{x}_R, \mathbf{p}_A, -t) = \tilde{\Theta}_b \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') \tilde{f}_1^-(\mathbf{x}_S, \mathbf{p}_A, -t') dt' \right\}. \quad (3.13)$$

Equation 3.12 and Equation 3.13 represent the plane-wave Marchenko equation, which are solved iteratively. The iterative scheme is initiated by setting plane-wave coda $\tilde{M}_1^+(\mathbf{x}_R, \mathbf{p}_A, t)$ to zero, and use $\tilde{G}_d(\mathbf{x}_S, \mathbf{p}_A, -t)$ as an initial estimation of $\tilde{f}_1^+(\mathbf{x}_S, \mathbf{p}_A, t)$.

Defining the plane-wave complementary time window $\tilde{\Psi}$ (Wapenaar et al., 2021):

$$\tilde{\Psi}_{a,b}(\mathbf{x}_R, \mathbf{p}_A, t) = 1 - \tilde{\Theta}_{a,b}(\mathbf{x}_R, \mathbf{p}_A, t). \quad (3.14)$$

By applying Equation 3.14 to Equation 3.8 and Equation 3.9 gives the plane-wave Green's functions (Wapenaar et al., 2021):

$$\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t) = \tilde{\Psi}_a \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') \tilde{f}_1^+(\mathbf{x}_S, \mathbf{p}_A, t') dt' \right\}, \quad (3.15)$$

$$\tilde{G}^{-,-}(\mathbf{x}_R, \mathbf{p}'_A, t) + \tilde{f}_{1,d}^+(\mathbf{x}_R, \mathbf{p}_A, -t) = \tilde{\Psi}_b \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_S \int_{-\infty}^{\infty} R(\mathbf{x}_R, \mathbf{x}_S, t-t') \tilde{f}_1^-(\mathbf{x}_S, \mathbf{p}_A, -t') dt' \right\}. \quad (3.16)$$

The computed Green's function $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ represents a source-redatumed reflection response, associated with a plane-wave virtual source at the focal depth \mathbb{S}_A , and observed by receivers at the acquisition surface \mathbb{S}_0 . In section 3.3, the plane-wave receiver-redatuming method will be explained, in addition to the application of the imaging condition.

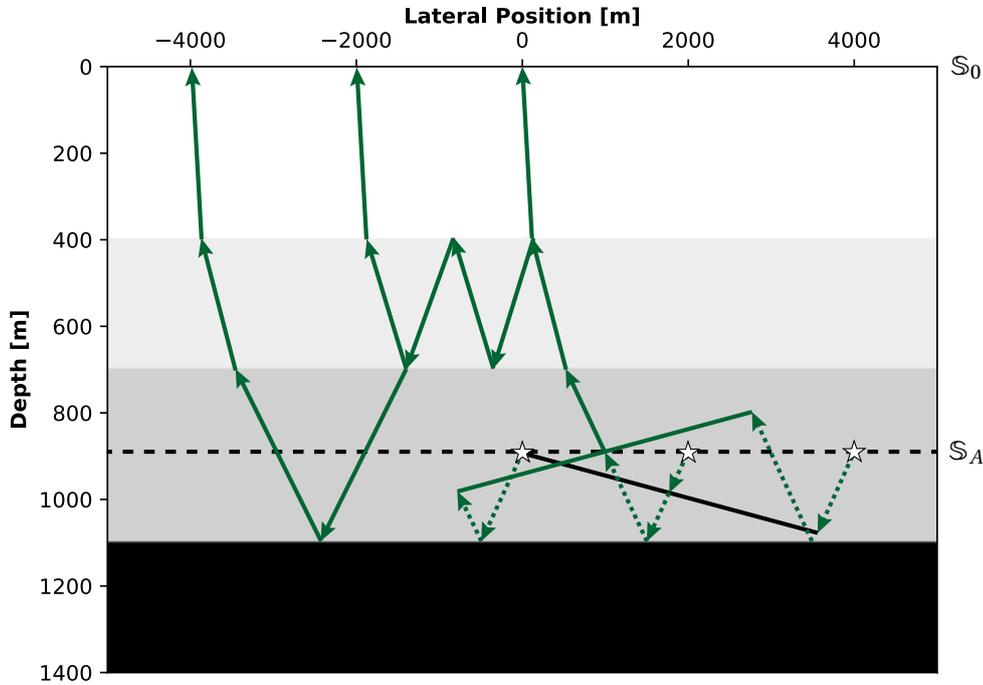


Figure 3.3: The down-going plane-wave Green's function $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ associated with plane-wave virtual source at the focal depth \mathbb{S}_A .

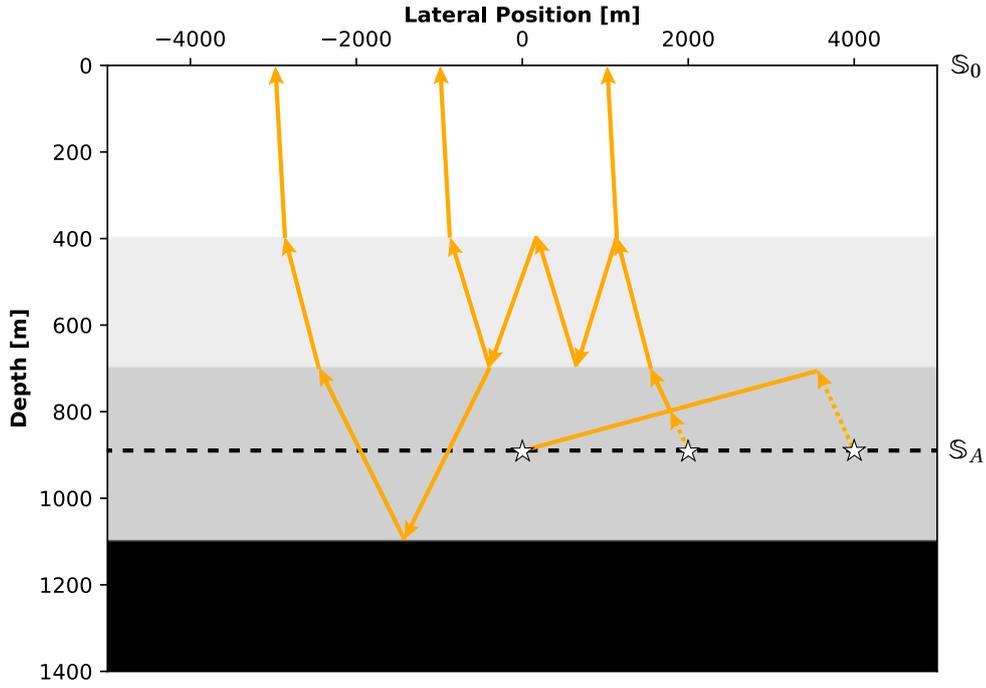


Figure 3.4: The up-going plane-wave Green's function $\tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t)$ associated with plane-wave virtual source at the focal depth \mathbb{S}_A .

3.3 Plane-Wave Receiver-Redatuming and Imaging

Similar to Equation 2.26, plane-wave receiver-redatuming is accomplished by applying the plane-wave double-focusing method, defined as (Meles et al., 2018; Wapenaar et al., 2021):

$$\tilde{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{p}_A, t) = \int_{\mathbb{S}_0} d\mathbf{x}_R \int_0^\infty f_1^+(\mathbf{x}_R, \mathbf{x}_B, t-t') \tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t') dt'. \quad (3.17)$$

Here, $\tilde{R}_{\text{tar}}(\mathbf{x}_B, \mathbf{p}_A, t)$ denotes the plane-wave reflection response of the physical medium below the focal depth \mathbb{S}_A . Note that $f_1^+(\mathbf{x}_R, \mathbf{x}_B, t)$ on the right-hand-side of Equation 3.17 is a point-source focusing function. Hence, applying double-focusing requires computing the point-source focusing functions for all \mathbf{x}_B locations along the focusing depth \mathbb{S}_A .

Extracting subsurface images using Equation 3.17 can be computationally expensive, especially if repeated for several plane-wave incident angles. Alternatively, subsurface images can be extracted more efficiently by combining receiver redatuming and imaging (Meles et al., 2018; Wapenaar et al., 2021) according to

$$\tilde{r}_{im}(\mathbf{x}_B, \mathbf{p}_A) = \left\{ \int_{\mathbb{S}_0} d\mathbf{x}_R \int_0^\infty G_d^{-,-}(\mathbf{x}_R, \mathbf{x}_B, -t') \tilde{G}^{-,+}(\mathbf{x}_R, \mathbf{p}_A, t-t') dt' \right\}_{t=\mathbf{p} \cdot \mathbf{x}_{H,B}}, \quad (3.18)$$

where $\tilde{r}_{im}(\mathbf{x}_B, \mathbf{p}_A)$ denotes the angle-dependent local reflection coefficient for all points \mathbf{x}_B along the focusing depth \mathbb{S}_A , $G_d^{-,-}(\mathbf{x}_R, \mathbf{x}_B, -t')$ denotes the time-reversed direct arrival of the point-source Green's function excited at \mathbf{x}_B along the focusing depth \mathbb{S}_A and received at point \mathbf{x}_R along the acquisition depth \mathbb{S}_0 , and $t = \mathbf{p} \cdot \mathbf{x}_{H,B}$ is the dipping plane-wave imaging condition. Images extracted using Equation 3.18 are equivalent to the images extracted from \tilde{R}_{tar} (Equation 3.17) at $t = \mathbf{p} \cdot \mathbf{x}_{H,B}$. Note that Equation 3.18 requires modelling $G_d^{-,-}(\mathbf{x}_R, \mathbf{x}_B, -t')$ for each point within the target zone. However, in section 3.4, practical solutions will be presented, which can expedite the imaging process. Also, note that for horizontal plane-waves, the imaging condition is defined as $t = 0$.

Comparing Equation 3.18 with Equation 2.27 reflects the computational advantage of the plane-wave Marchenko imaging method over the conventional Marchenko imaging methods. The

advantage arises from the fact that while the the plane-wave Marchenko method requires nz (times the number of incident angles) Marchenko solutions, the conventional Marchenko method requires $nz \times nx$ Marchenko solutions to construct the same image. Therefore, the plane-wave Marchenko provides better efficiency and reduced computational cost.

Images constructed by the plane-wave Marchenko method can show limited horizontal resolution due to poor illumination. However, the resolution can be improved by stacking images associated with multiple plane-wave incident angles. The number of required angles depends on the geometry of the subsurface events.

3.4 Plane-Wave Marchenko Imaging Workflow

Provided that a macro velocity model is available and the reflection data satisfy the Marchenko method conditions, as described in [section 2.5](#), imaging the subsurface using the plane-wave Marchenko method can be achieved by applying the workflow illustrated in [Figure 3.5](#).

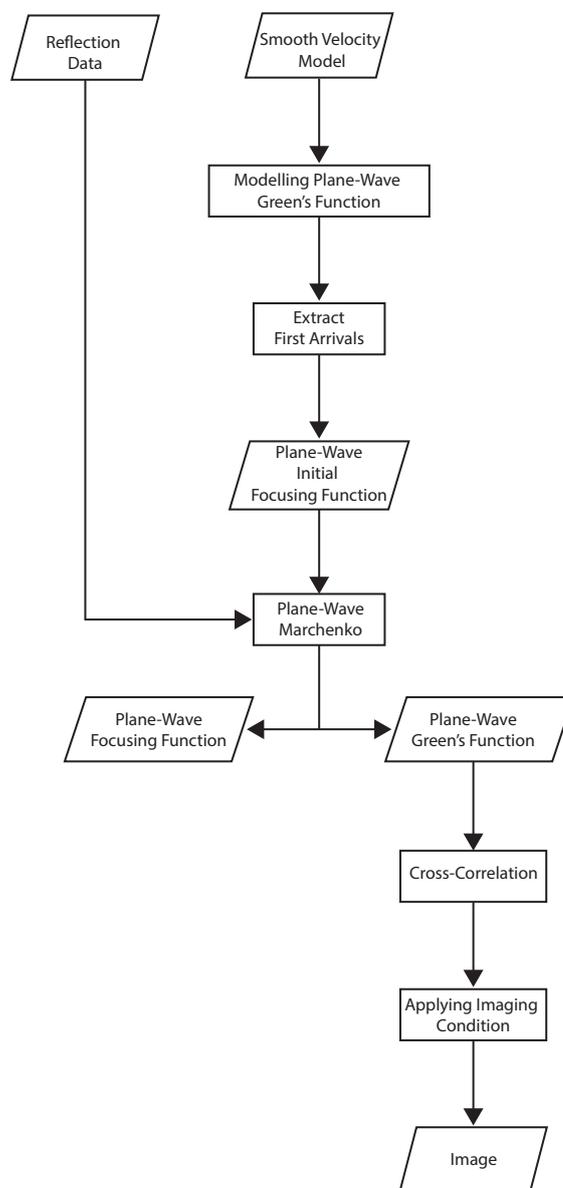


Figure 3.5: The adopted plane-wave Marchenko imaging workflow. The different approaches of performing the cross-correlation step are illustrated in [Figure 3.7](#).

The workflow starts with modelling the plane-wave Green's function by placing the plane-wave source at the imaging depth and the receivers at the acquisition plane. This modelling can be performed either by finite difference modelling or by an Eikonal solver. In general, the number of modelling receivers should match the total number of receivers present in the reflection data. Thus, for a moving spread acquisition, receivers that do not coincide with source locations should also be taken into account. According to Equation 3.1, modelling plane-waves with negative incident angles results in dips that are defined in negative time. Therefore, as finite difference modelling does not allow for modelling in negative time, the absolute value of t_{delay} is taken and then flipped with respect to the vertical axis, as shown in Figure 3.6 (a) and (b). Also, note that the dip of the modelled plane-wave Green's function is equal to the dip of the virtual plane-wave source, as explained in Appendix A. Then, the modelled plane wave response is muted such that

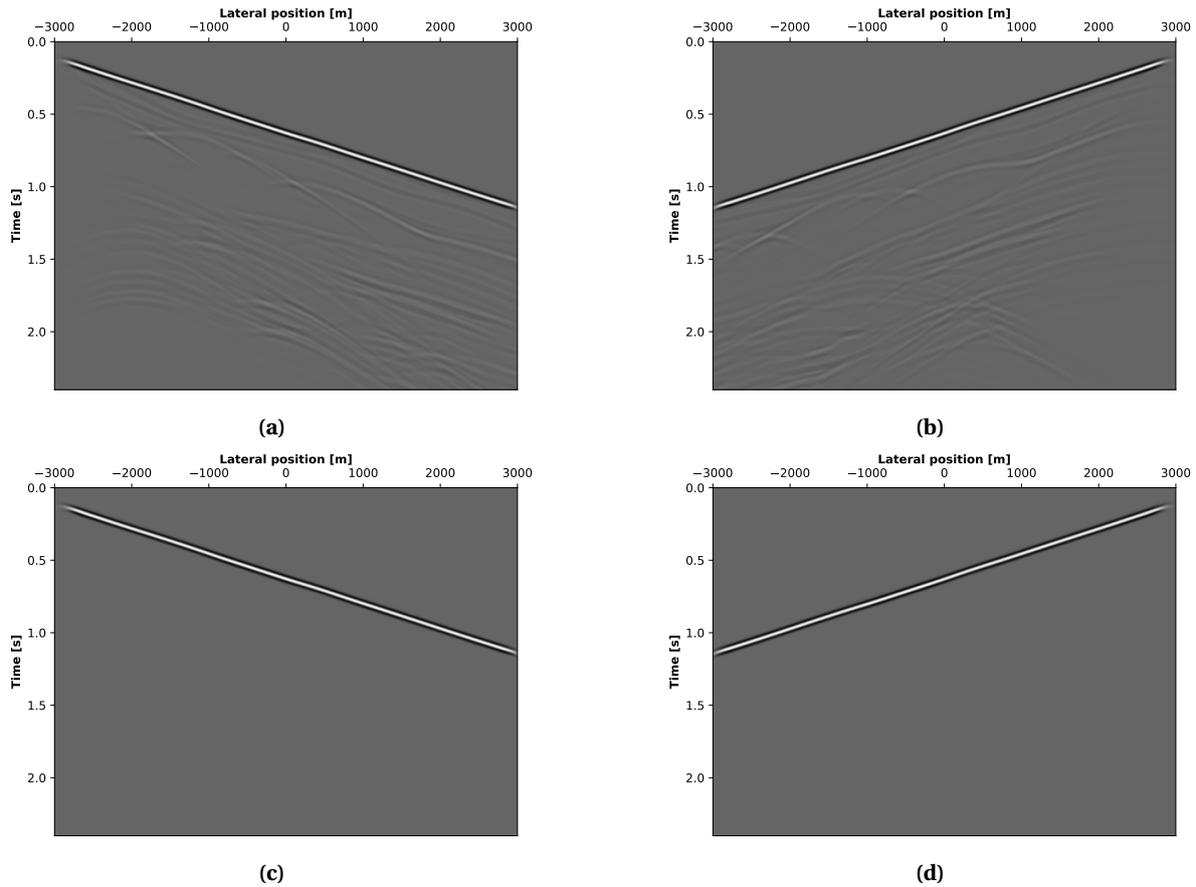


Figure 3.6: Examples for plane-wave Green's functions modelled via finite difference modelling for (a) angle= 15° , and (b) angle= -15° . (c), (d) are the muted versions of (a) and (b), respectively. The complex conjugates of (c) and (d) represent the initial focusing function used to initiate the Marchenko iterative scheme.

only the plane-wave first arrival is preserved, as shown in Figure 3.6 (c) and (d). Under the direct wave assumption (Equation 2.12), the complex conjugate of this first arrivals represents the initial focusing function $\tilde{f}_{1,d}^+$, which is used to initiate the plane-wave Marchenko iterative scheme. The retrieval of the initial plane-wave focusing functions is repeated for each imaging depth.

The next step is to retrieve the plane-wave Marchenko solutions, which are confined to the lateral position range where sources and receivers coincide, as dictated by the plane-wave Marchenko equation (Equation 3.8 and Equation 3.9). With reference to Equation 3.18, prior to applying the imaging condition, the $\tilde{G}^{-,+}$ functions need to be inversely extrapolated to their corresponding depths. This can be achieved by applying different approaches, as illustrated Figure 3.7. The first approach is to model the point-source Green's functions $G^{-,-}$ for each lateral position at the imaging depth via finite different modelling, and then extract the direct arrival, followed by

cross-correlating them with the $\tilde{G}^{-,+}$ function associated with the same imaging depth. This approach can be computationally expensive, especially for small lateral and vertical imaging intervals. Alternatively, the $\tilde{G}^{-,+}$ functions can be back-propagated in the time domain using finite difference extrapolation to their corresponding imaging depths. The finite difference extrapolation is performed by injecting the last samples of the plane-wave Green's functions $\tilde{G}^{-,+}$ first into the model, and hence the first event of $\tilde{G}^{-,+}$ will be recorded last. While this approach maintains the amplitude accuracy of the latter back-propagation approach, it can reduce the computational cost as with this approach performing the cross-correlation is not needed and is accounted for in the extrapolation process. Similar to the first approach, a third approach is to model the direct arrival of the point-source Green's functions with an Eikonal solver, followed by cross-correlation with the corresponding plane-wave down-going Green's functions $\tilde{G}^{-,+}$. This approach reduces the computational cost significantly. However, as the modeled Eikonal solver outputs include amplitude inaccuracies, the back-propagation results are also inaccurate amplitude-wise.

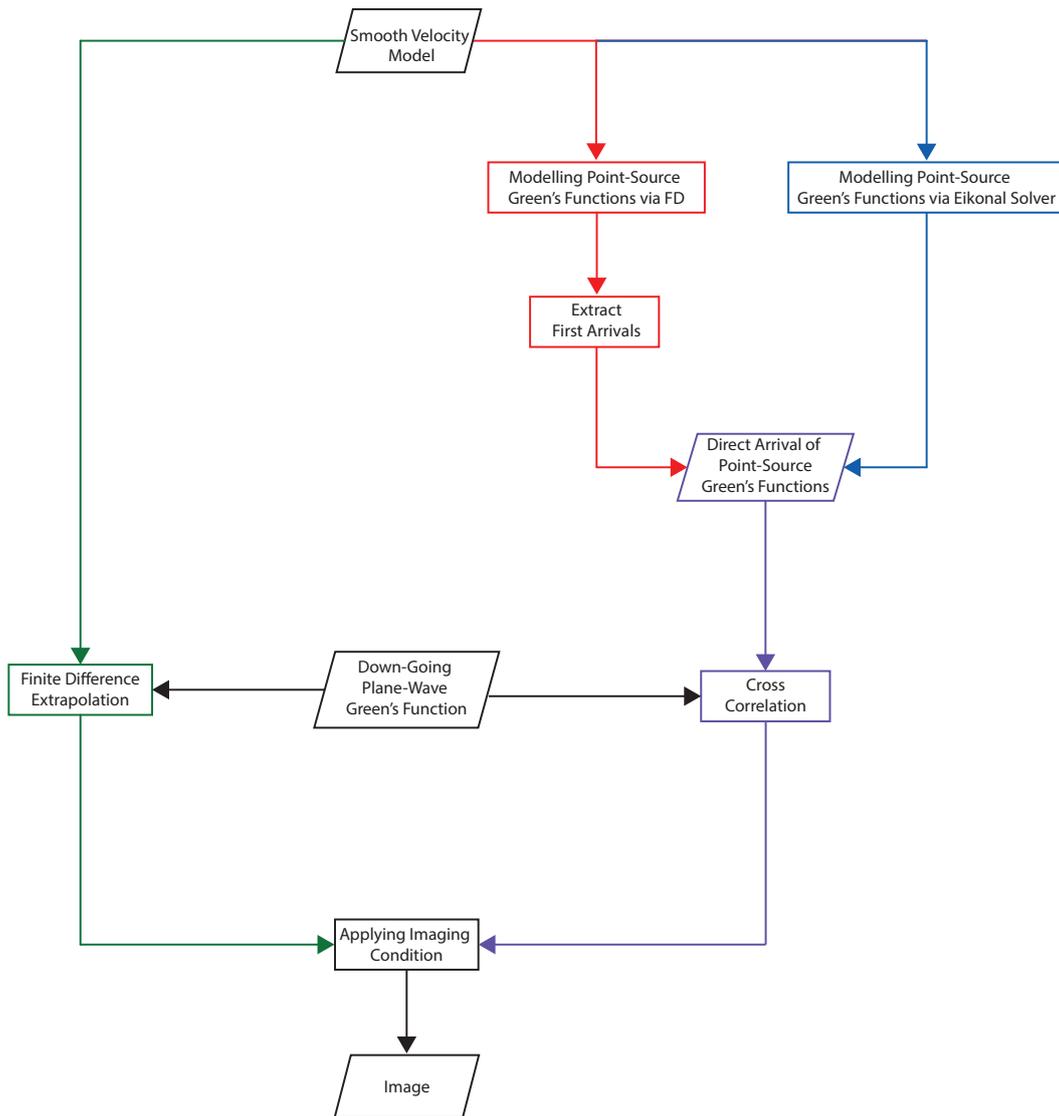


Figure 3.7: A workflow of the different approaches of performing the back-propagation of the down-going plane-wave Green's function $\tilde{G}^{-,+}$. The green lines indicate the back-propagation by finite difference extrapolation approach. The red lines indicate the back-propagation by cross-correlation with point-source Green's functions modelled by finite difference modelling. The blue lines indicate the back-propagation by cross-correlation with point-source Green's functions modelled with an Eikonal solver. The purple lines indicate the functions and processing step common to the latter two approaches. Note that for the first approach, the cross-correlation step is taken into account by the finite difference extrapolation process.

To further reduce the computational expense using back-propagation by finite difference extrapolation, the $\tilde{G}^{-,+}$ functions can be temporally truncated. The extent of the truncation depends on the geometry of the subsurface interfaces. The truncation is justified by presence of internal multiples from the medium below the focusing depth \mathbb{S}_A , which do not contribute to the imaging process, and hence, the truncation reduces the back-propagation computational cost. The caveat is that over-truncating the $\tilde{G}^{-,+}$ would result in incomplete images, and hence applying the truncation is not recommended, especially for field datasets. See [Appendix B](#) for the effect of different truncation extents on the extracted images.

After applying the inverse extrapolation, the plane wave imaging condition is applied. Referring to [Equation 3.18](#), the imaging condition is defined as:

$$t = \mathbf{p} \cdot \mathbf{x}_{H,B}, \quad (3.19)$$

which is applied to the result of the cross-correlation of $G_d^{-,-}$ and $\tilde{G}^{-,+}$. Based on the aforementioned description of how finite difference extrapolation is performed, the practical imaging condition for back-propagation by finite difference extrapolation is defined as:

$$t = t_0 + (\mathbf{p} \cdot \mathbf{x}_{H,B}) - \max(\mathbf{p} \cdot \mathbf{x}_{H,B}) - \epsilon, \quad (3.20)$$

where t_0 is duration of the plane-wave Green's function $\tilde{G}^{-,+}$, which is the equivalent of the zero-lag cross-correlation time, $\mathbf{p} \cdot \mathbf{x}_{H,B}$ is the linear time delay associated with dipping plane waves, $\max(\mathbf{p} \cdot \mathbf{x}_{H,B})$ is the maximum time delay associated with dipping plane-waves. Although, the plane-wave down-going Green's functions $\tilde{G}^{-,+}$ are zero-phased, the outputs of finite difference extrapolation are not, which is attributed to the adopted finite difference parametrization. Hence, the inclusion of ϵ , which is equal to half the wavelength of the wavelet, is necessary to extract the maximum energy and produce a zero-phased image. Note that for back-propagation by cross-correlation, ϵ is not needed as the cross-correlation outputs are zero-phased. [Figure 3.8](#) shows a visual representation of the terms expressed in [Equation 3.19](#) and [Equation 3.20](#). In addition, [Figure 3.8](#) shows that, in terms of event geometry, the result of finite difference extrapolation approach is a time-reversed version of the cross-correlation result, which is attributed to how back-propagation by finite difference extrapolation is performed. Despite the aforementioned differences, the two back-propagation approaches are equivalent, and their associated imaging conditions should extract the same events at the same imaging depths.

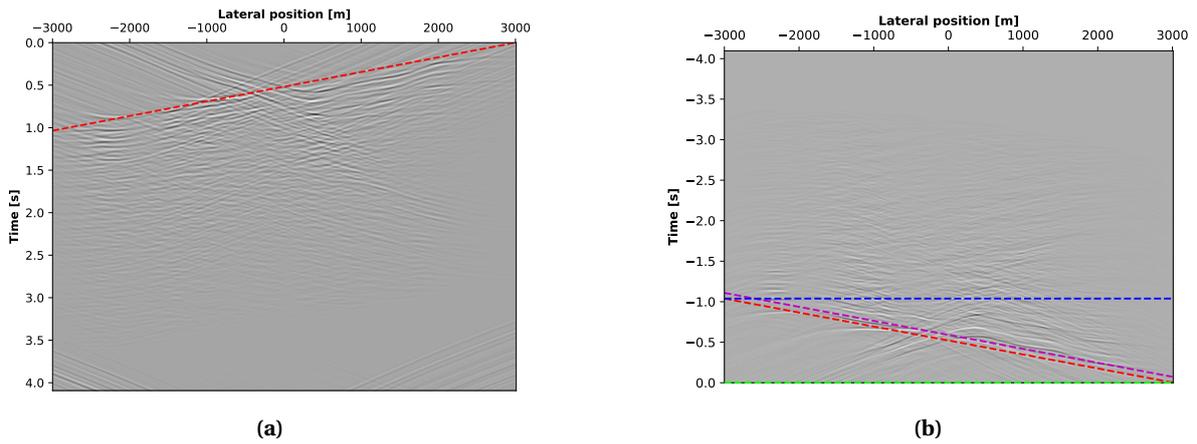


Figure 3.8: (a) The imaging condition associated with back-propagation by cross-correlation. (b) The imaging condition associated with back-propagation by finite difference extrapolation. The green line represents t_0 which is equal to the duration of the Green's function $\tilde{G}^{-,+}$ (coincides with $t=0$), used as an input to the back-propagation process. The blue line represents the maximum plane-wave time delay, $\max(\mathbf{p} \cdot \mathbf{x}_{H,B})$. The red line represents the linear plane-wave time delay, $\mathbf{p} \cdot \mathbf{x}_{H,B}$. The magenta line represents the shift of the plane-wave time delay by half the wavelength of the wavelet denoted by ϵ .

Theoretically, the observed wavelength depends on the incident angle of the plane wave, which is equal to the angle the plane wave forms with respect to a horizontal plane, as shown in [Figure 3.9](#).

The change in the observed wavelength λ_z as a function of incident angle α is given by:

$$\lambda_z = \frac{\lambda}{\cos(\alpha)} \quad (3.21)$$

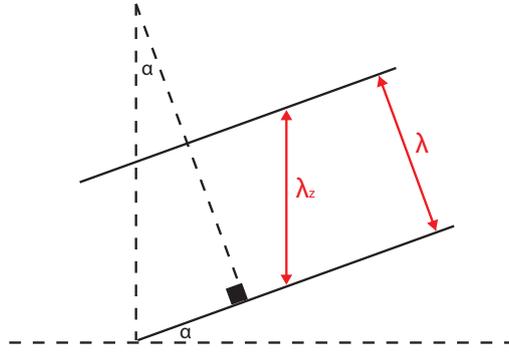


Figure 3.9: Schematic representation of the effect of the plane-wave incident angle α on the recorded wavelength λ_z for a horizontal receivers array.

For dipping plane-waves, using Equation 3.20 to apply the imaging condition means that for each depth, the image extraction is performed along a slanted line, as shown in Figure 3.10a, which results in a jittered image, as shown in Figure 3.10c. This jitter can be eliminated by applying a lateral position specific phase-shift in the frequency domain (i.e. a time shift in the time domain). The phase-shift plays the role of projecting the dipping events to a horizontal plane, and is applied by multiplying each trace in the frequency domain by $e^{j\omega\Delta t}$, where Δt is the lateral position specific time delay. Applying the phase-shift reduces Equation 3.19 to

$$t = 0. \quad (3.22)$$

Similarly, applying the phase-shift reduces Equation 3.20 to:

$$t = t_0 - \max(\mathbf{p} \cdot \mathbf{x}_{H,B}) - \epsilon. \quad (3.23)$$

After applying the phase shift, the image extraction is performed along a horizontal line, as shown in Figure 3.10b, resulting in smoother image as shown in Figure 3.10d.

The extracted images are contaminated with imaging artifacts, as indicated by the red arrows in Figure 3.10 (c) and (d). These artifacts can be attenuated using a dip filter if the dip of the artifacts is different from that of the interfaces. However, as filtering can damage the event's amplitude, it should be performed with caution, especially if the retrieved images are used for amplitude-sensitive applications, such as Amplitude-Versus-Angle (AVA). This workflow is repeated for all imaging depths within the target zone, and for all necessary illumination angles. The counterpart RTM-based plane-wave images can be created following the same workflow by limiting the Marchenko iterative scheme to one iteration.

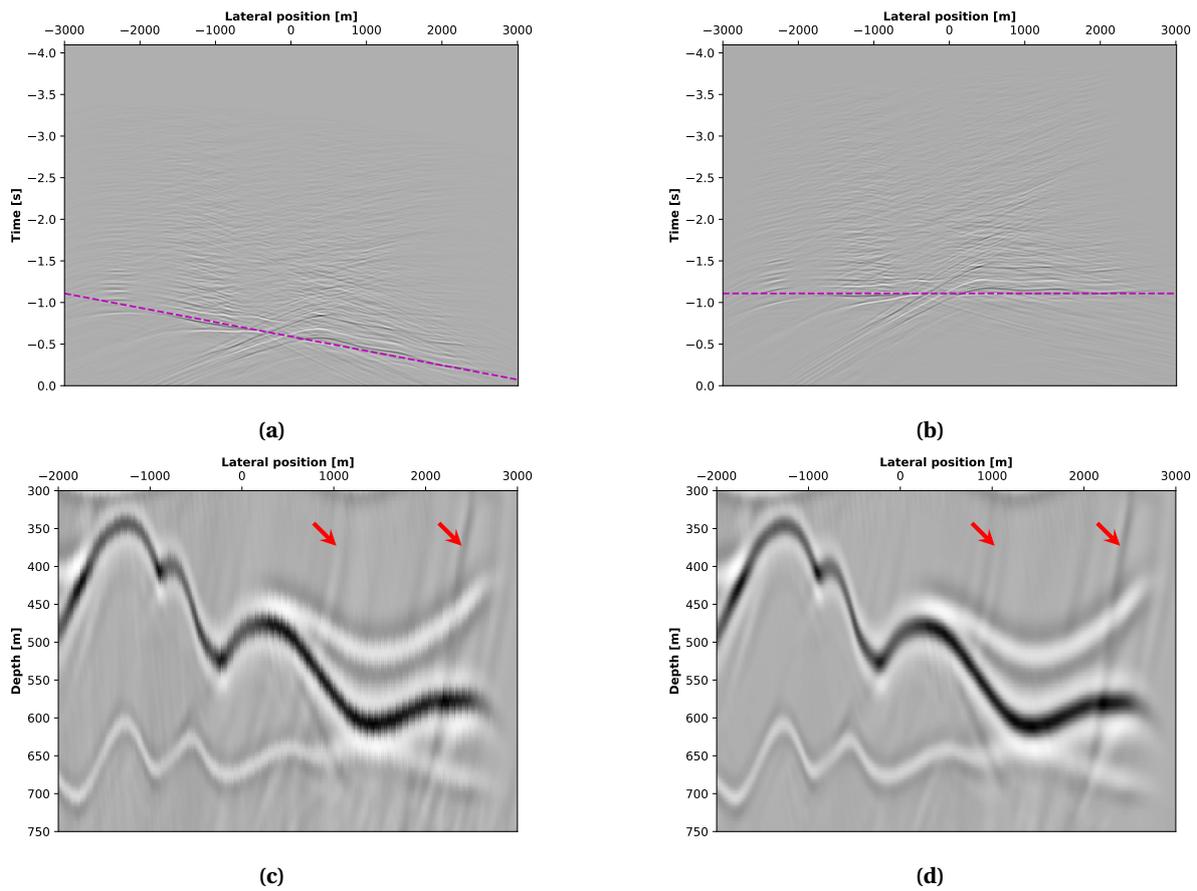


Figure 3.10: (a) The imaging condition expressed in Equation 3.20, which extracts the image along a slanted line, resulting in a jittered image (c). (b) The phase-shifted imaging condition, expressed in Equation 3.23, which extracts the image along a horizontal line, resulting in a smooth and jitter-free image (d). The red arrows indicate imaging artefacts, which can be attenuated using a dip filter.

4

APPLICATION I: SIMPLE MODEL

In this chapter, and [chapter 5](#), we reproduce the work presented in Figures 7 and 9 of [Meles et al. \(2018\)](#), respectively. The aim of this reproduction is to establish and test the plane-wave Marchenko imaging workflow described in [section 3.4](#). In these two chapters we follow the complexity progression presented by the original authors, starting with a simple model in this chapter, followed by a more complex model in [chapter 5](#).

4.1 Model Description

The model under consideration, shown in [Figure 4.1](#), consists of gently dipping reflectors and layers of variable thickness. The reflection data is modelled using a fixed spread with 601 co-located sources and receivers, ranging between -3000m and 3000m , and with a spatial increment of 10m . In addition, each shot is recorded by all receivers, with a sampling rate of 4ms . Furthermore, to model the reflection response, a source wavelet with a flat frequency spectrum between $f = 5\text{Hz}$ and $f = 80\text{Hz}$ was used.

Next, the plane wave imaging workflow, solely using horizontal plane-waves, was applied to generate an image from the reflection data. This was done by defining focal planes, with a depth increment of 5m , covering the entire vertical extent of the model. First, the smoothed model, shown in [Figure 4.2](#) was used to model the initial focusing functions with a Ricker wavelet characterized by a central frequency of 20Hz . These initial focusing functions were used to initiate the Marchenko iterative scheme. Upon retrieving the Marchenko scheme solutions, the smoothed model was used again to back-propagate the down-going Green's functions to their respective focal depths via finite difference extrapolation. Similarly, the counterpart RTM-based horizontal plane-wave image was created by limiting the Marchenko iterative scheme to a single iteration.

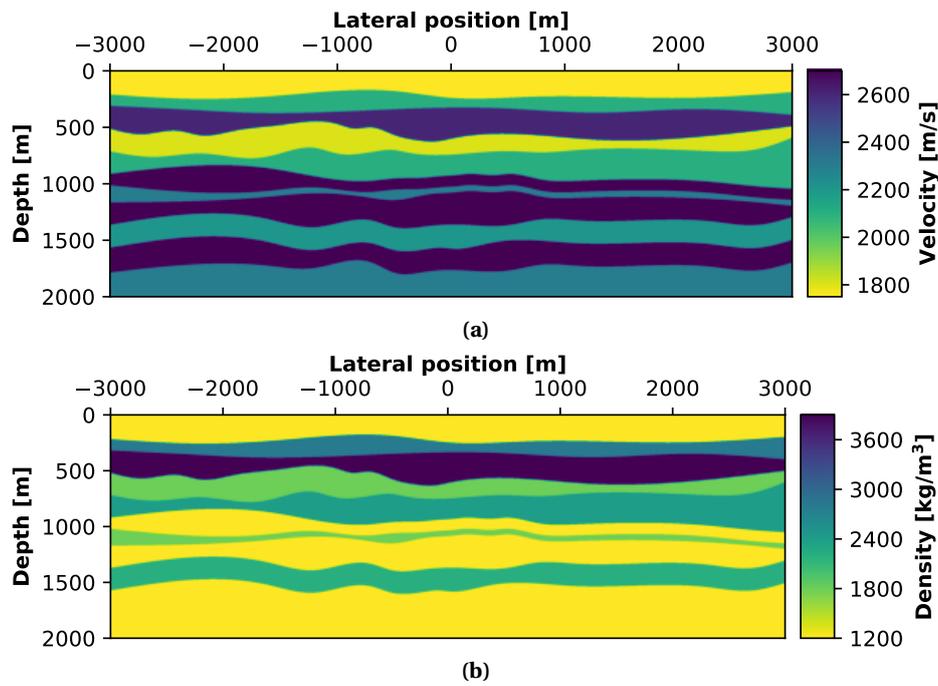


Figure 4.1: The velocity (a) and density (b) models used for computing the reflection response.

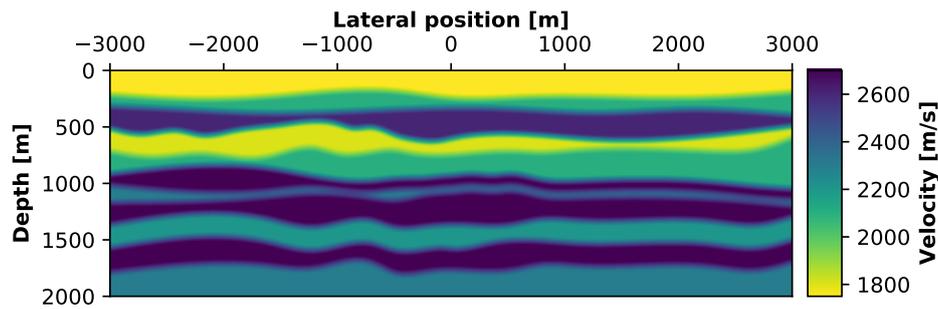


Figure 4.2: The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.

4.2 Imaging Results

Figure 4.3 presents the imaging results that were retrieved after applying the previously discussed workflow for RTM (Figure 4.3a) and after applying 20 Marchenko iterations (Figure 4.3b). The figure shows that the RTM-based image is contaminated with internal multiples, as indicated by the red arrows. While these multiples pose little problems in the shallow parts, they mask the primary events at larger depths ($> 1000\text{m}$). On the contrary, the Marchenko-based plane-wave image is free of internal multiples and the reflectors are properly imaged, except for the deep reflector highlighted by the blue arrow, which is discontinuous due to a lack of illumination. Dipping plane-waves were not considered for the current model, because of the excellent imaging performance of the horizontal plane-waves. However, in the next chapter the importance of adding dipping plane-wave images will be discussed in depth.

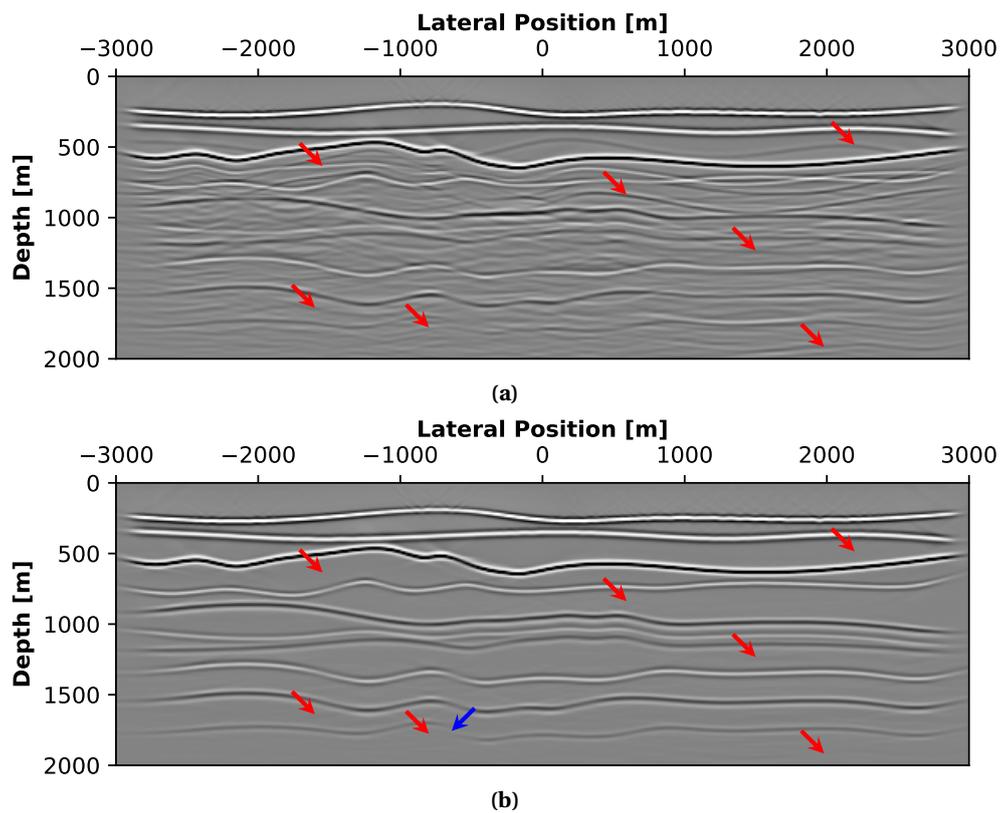


Figure 4.3: A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane wave image. The images are displayed using the same scaling factor. The red arrows indicate internal multiples that were suppressed using the plane-wave Marchenko imaging method. The blue arrow indicate a deep, dipping reflector that was poorly illuminated by horizontal plane-waves.

5

APPLICATION II: COMPLEX MODEL

5.1 Model Description

The model under consideration, shown in Figure 5.1, is characterized by a variety of layers, including relatively steeply dipping reflectors, thin layers, and diffractors. The reflection data were modelled using a fixed spread with 601 co-located sources and receivers, ranging between -3000m and 3000m , and with a spatial increment of 10m . In addition, a wavelet characterized by a flat spectrum of $0\text{-}5\text{-}80\text{-}100\text{Hz}$ was used for to model the reflection response. Furthermore, each shot was recorded by all receivers, with a sampling rate of 4ms .

The plane-wave imaging workflow was applied with a depth increment of 5m , covering the entire depth range of the model. The plane wave initial focusing functions were modelled using the smoothed model shown in Figure 5.2, using a Ricker wavelet with a central frequency of 20Hz . Additionally, the smoothed model was used to back-propagate the down-going plane-wave Green's functions to their corresponding depths by finite difference extrapolation. Using the same initial focusing functions, the RTM-based plane-wave images were created by using a single iteration of the Marchenko iterative scheme.

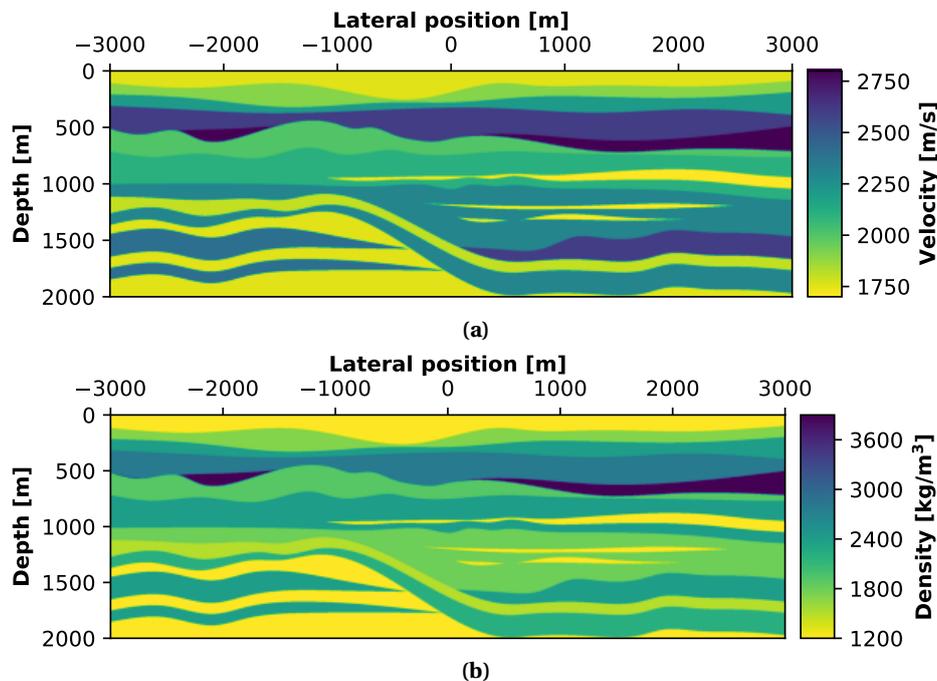


Figure 5.1: The velocity (a) and density (b) models used for computing the reflection response.

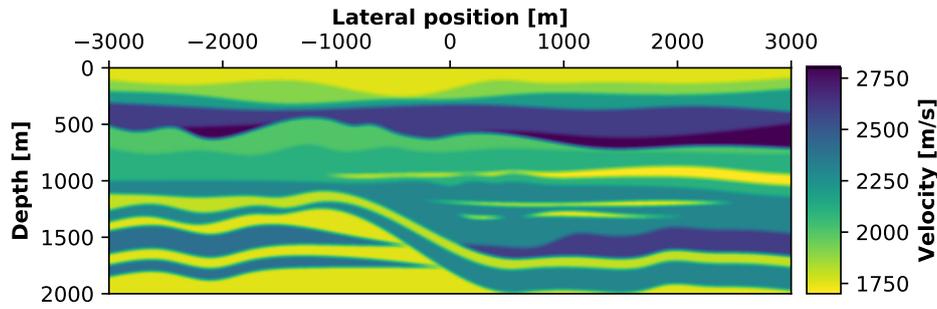


Figure 5.2: The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.

5.2 Imaging Results

A comparison between the RTM-based and the Marchenko-based horizontal plane-wave images is shown in [Figure 5.3](#). For both images, imaging artefacts were suppressed using a dip filter. This filter was designed such that only near-vertical artefacts were attenuated. Consequently, artefacts with similar dip to the reflectors remained in the images. The RTM-based horizontal plane-wave image is contaminated with internal multiples, as indicated by the red arrows. The multiples degrade the amplitude and continuity of the reflectors, especially below 1000m depth, making the interpretation of the RTM-based image difficult. In contrast, the Marchenko-based image shows a significant improvement over the RTM-based image. Internal multiples are greatly suppressed, improving the amplitude and continuity of the imaged reflectors. However, the multiple suppression performance is not perfect as residual internal multiples are present, mainly at the area mostly contaminated by the internal multiples below 1000m depth. Despite the presence of artefacts and residual internal multiples, reflectors are properly imaged and the image can be easily interpreted. In addition, discontinuities, highlighted by the red and green boxes in [Figure 5.3b](#), are present and are associated with deep dipping reflectors, hence the need for additional dipping plane-wave images arises.

To reduce the discontinuities observed in the Marchenko-based horizontal plane-wave image, we take a different approach to that taken by [Meles et al. \(2018\)](#). While the original authors used dipping plane-waves to specifically target the discontinuity zones, we generate dipping plane-wave images over the entire extent of the model, which were then stacked with the horizontal plane-wave image. The plane-wave Marchenko imaging workflow was repeated for six additional incident angles, ranging from -15° to -5° and from 5° to 15° with an increment of 5° and a depth sampling of 5m. The individual Marchenko-based dipping plane-wave images are shown in [Figure 5.4](#). These images were filtered using the same mild dip filter applied to the horizontal plane-wave image. By analyzing the individual plane-wave images, it can be concluded that imaging performance is consistent for the upper half of the model, above 1000m depth. However, below this depth level, the individual images differ. The images associated with positive plane-wave incident angles ([Figure 5.4b](#), [Figure 5.4d](#), and [Figure 5.4f](#)) manage to properly illuminate the event discontinuities highlighted by the green box in [Figure 5.3](#). Similarly, images associated with negative plane-wave incident angles ([Figure 5.4a](#), [Figure 5.4c](#), and [Figure 5.4e](#)) manage to properly illuminate the event discontinuities highlighted by the red box in [Figure 5.3](#). In addition, the model edge that is opposite to the plane-wave propagation direction is poorly illuminated. Furthermore, the orientation of the linear imaging artefacts is perpendicular to the propagation direction of the dipping plane-waves. Moreover, there is an inverse relationship between the plane-wave incident angle and the frequency of the observed imaging artefacts such that as the the incident angle increases, the frequency of the imaging artefacts decreases. Similar to the horizontal plan-wave image, residual internal multiples are present below 1000m depth.

The individual dipping plane-wave images were stacked to generate an image that accounts for the variable dip of the reflectors and does not suffer from event discontinuities. Stacking was

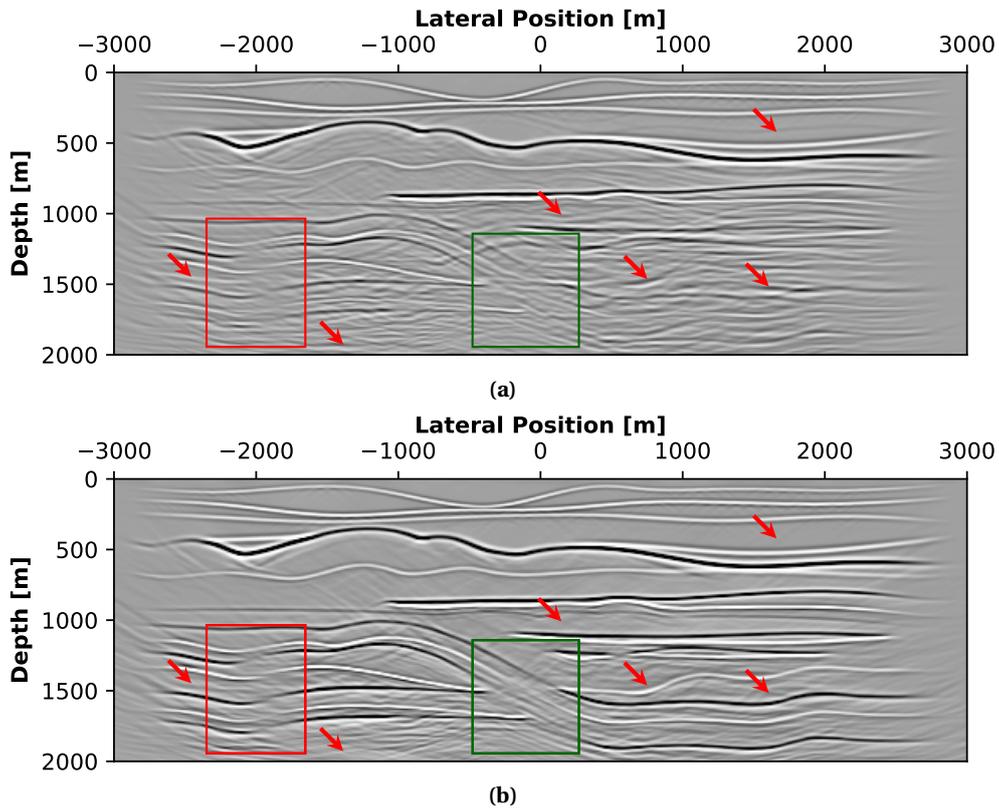


Figure 5.3: A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane-wave image. Both images are displayed using the same scaling factor. The red arrows indicate internal multiples that are mostly suppressed using the plane-wave Marchenko imaging method.

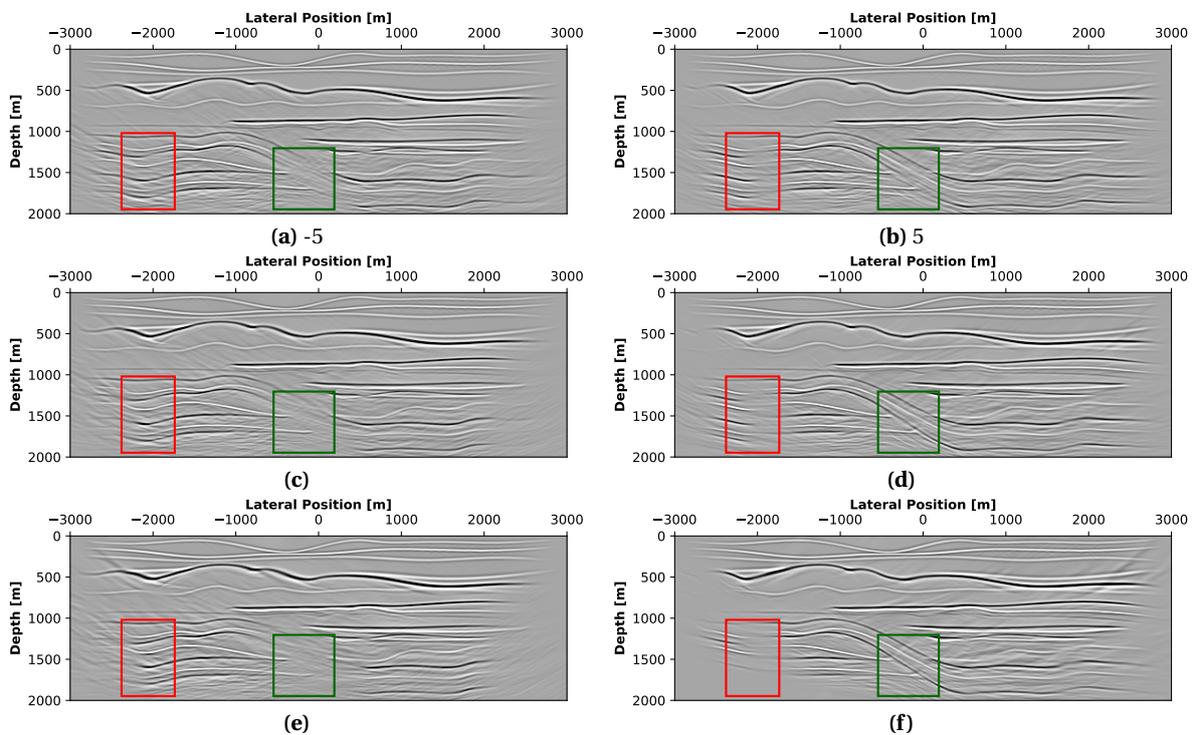


Figure 5.4: The individual Marchenko-based dipping plane-wave images associated with (a) -5° , (b) 5° , (c) -10° , (d) 10° , (e) -15° , and (f) 15° . All images are displayed using the same scaling factor. The red and green boxes indicate the location of the event discontinuities observed in the Marchenko-based horizontal plane-wave image (Figure 5.3).

performed by taking the average of all the dipping plane-wave images, in addition to the horizontal plane-wave image. A comparison between the horizontal plane-wave image and the dipping plane-wave images stack is shown in Figure 5.5. The stack image shows an improvement in terms of continuity in comparison with the horizontal plane-wave image at the location of the identified discontinuities. However, as the different dipping plane-waves illuminate different parts of model, the stack image shows a deterioration of the amplitude uniformity as a result of the stacking process. This deterioration can be seen, for example, at the location of the former discontinuities. The stacking process also reduces the amplitude of the imaging artefacts and residual internal multiples.

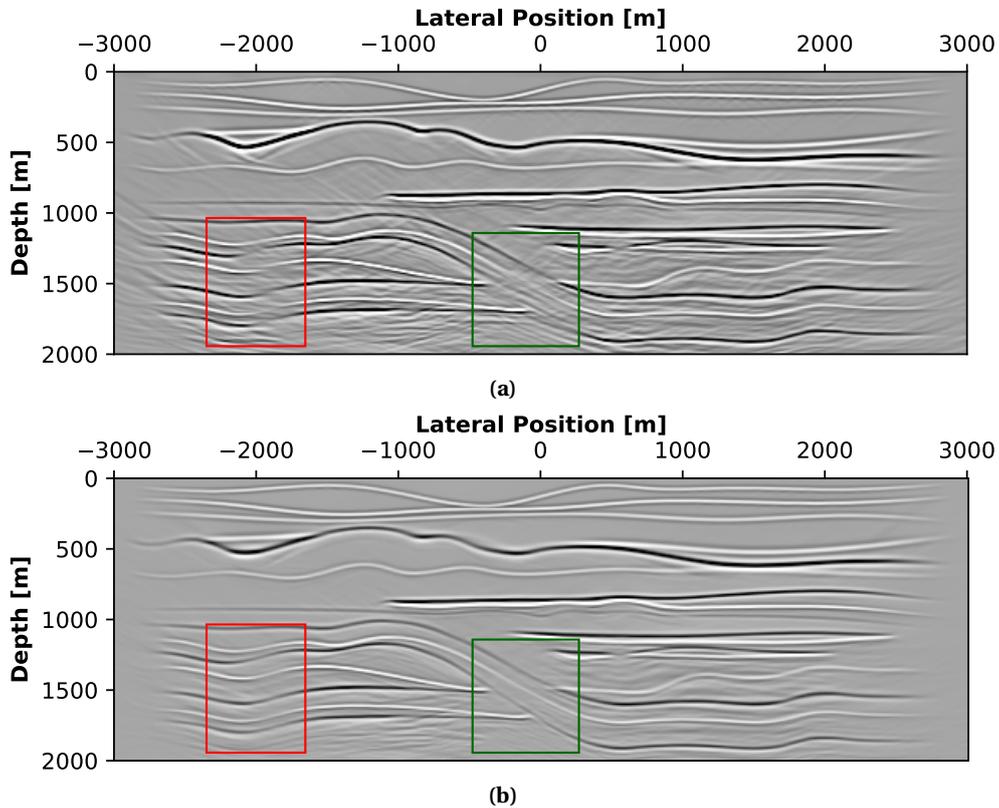


Figure 5.5: A comparison between (a) the Marchenko-based horizontal plane-wave image, and (b) the stack of the Marchenko-based dipping plane-waves images, with incident angles ranging from -15° to 15° , inclusive, with an increment of 5° . Both images are displayed using the same scaling factor.

6

APPLICATION III: SALT MODEL

6.1 Model Description

The model under consideration, shown in Figure 6.1, consists of multiple layers, which are characterized by high reflection coefficients and varying dip. These layer overlay deeper layers, that are characterized by low reflection coefficients. The importance of this model is that it represents a succession of multiple generating salt bodies, that usually prevent the interpretation of any underlying potential prospects. The reflection data were modelled using a fixed spread of 751 co-located sources and receivers, with a spatial sampling of 16m, and using a wavelet of a flat frequency spectrum with frequencies between 5Hz and 80Hz. In addition, the reflected wavefield was recorded by all receivers, with a sampling rate of 8ms.

The smoothed model, shown in Figure 6.2, was used to model the initial focusing functions, with a depth increment of 8m, covering the entire depth extent of the model. The initial focusing functions were modelled using a Ricker wavelet with a central frequency of 20Hz. Additionally, the smoothed model was used to back propagate the down-going plane-wave Green's functions via finite difference extrapolation to their corresponding depths.

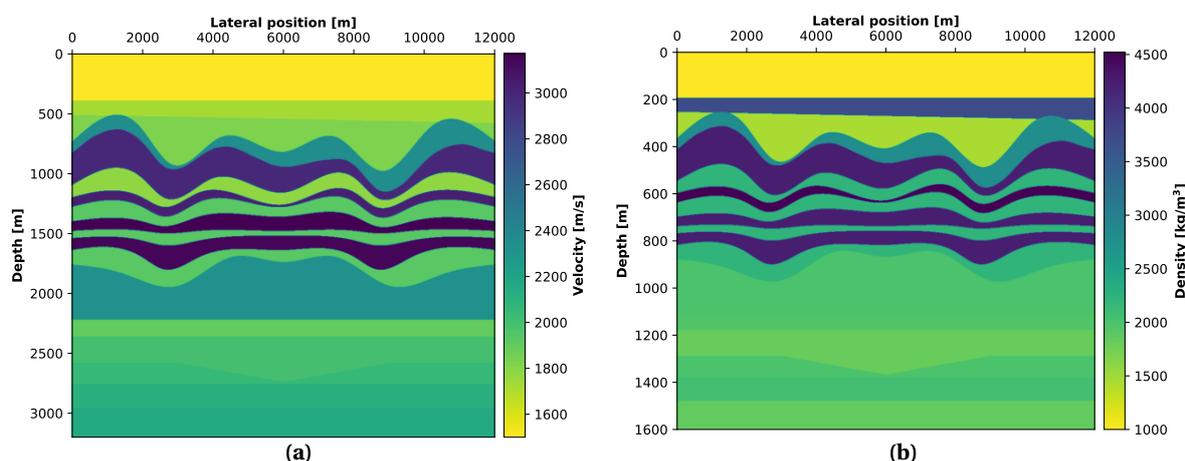


Figure 6.1: The velocity (a) and density (b) models used for computing the reflection response.

6.2 Imaging Results

Figure 6.3 shows a comparison between the RTM-based and the Marchenko-based horizontal plane-wave images. For these images, near-vertical imaging artefacts were attenuated using a dip filter. Other imaging artefacts were not filtered to avoid damaging the reflectors. In addition, to increase the amplitude of the deep layer interfaces associated with low reflection coefficient, a

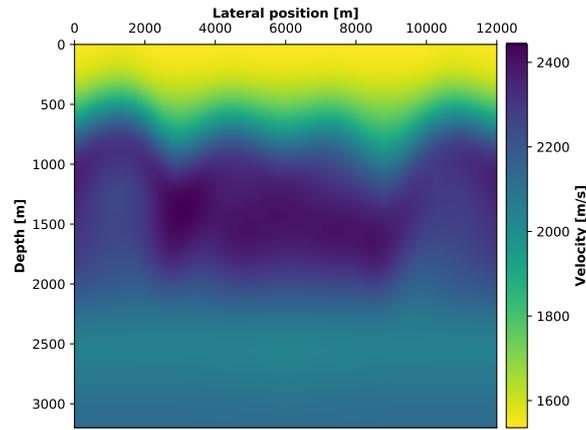


Figure 6.2: The smoothed velocity model used to model the initial focusing functions and to back-propagate the down-going plane-wave Green's functions to their corresponding depths.

time-dependent scaling factor of $e^{(1.4t)}$ was applied to both images. On one hand, the RTM-based horizontal plane-wave image shows a severe contamination by internal multiples. Reflectors below 1300m depth, including the high amplitude events, are completely masked by internal multiples and are impossible to interpret. On the other hand, the Marchenko-based plane-wave image shows a significant improvement in terms of events definition as events can be easily traced along the lateral extent of the image. However, residual internal multiples, indicated by the red arrows and characterized by similar geometry and orientation to the primary events, are present in the Marchenko-based image, making the differentiation between primary and multiple events ambiguous. The main advantage of the Marchenko-based image is the imaging of the low-amplitude deep reflectors, which are impossible to observe from the RTM-based image.

Figure 6.3 shows that horizontal plane-waves were sufficient to properly image the model without any illumination limitations. Therefore, one can safely assume that there is not a practical advantage of producing images for more angles illumination-wise. However, to investigate the behavior of dipping plane-wave images, Marchenko-based images associated with incident angles -10° , 10° , and 25° were created, as shown in Figure 6.4. The images associated with angles -10° and 10° show similar observations to the horizontal plane-wave image (Figure 6.3b) in terms of the imaging performance and the presence of residual internal multiples. However, the image associated with angle 25° does not image reflectors below 1300m depth. In addition, the imaging artefacts contaminate the image more than the artefacts associated with less steep angles and are characterized by a relatively low frequency. Furthermore, the lateral resolution is degraded for parts of the reflectors that have an opposite dip to that of the plane-wave, as indicated by the red arrows.

Figure 6.5 shows the stacking of different combinations of Marchenko-based angle-dependent images. Stacking near-horizontal images, namely angles -10° , 0° , 10° , produces an image that has the same lateral resolution as its constituents. However, including the image associated with angle 25° to the stacking process deteriorates the resolution of the stack image at the locations where the dip of the reflectors is opposite to that of angle 25° , as indicated by the red arrows. In addition, the deterioration of the lateral resolution is more severe when the image associated with angle 25° was stacked directly with the horizontal plane-wave image. Furthermore, from the latter two stack images, it can be observed that the low frequency imaging artefacts associated with angle 25° image negatively affect the reflectors definition at various locations.

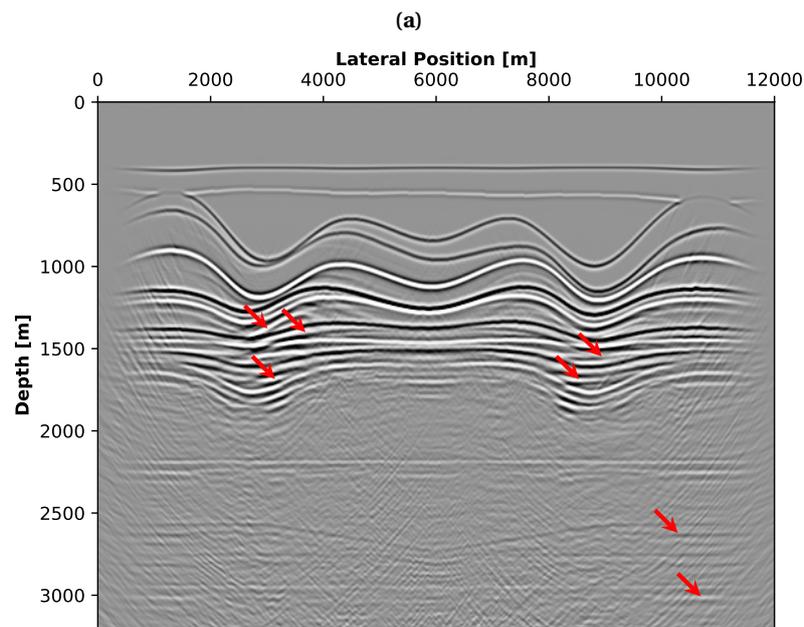
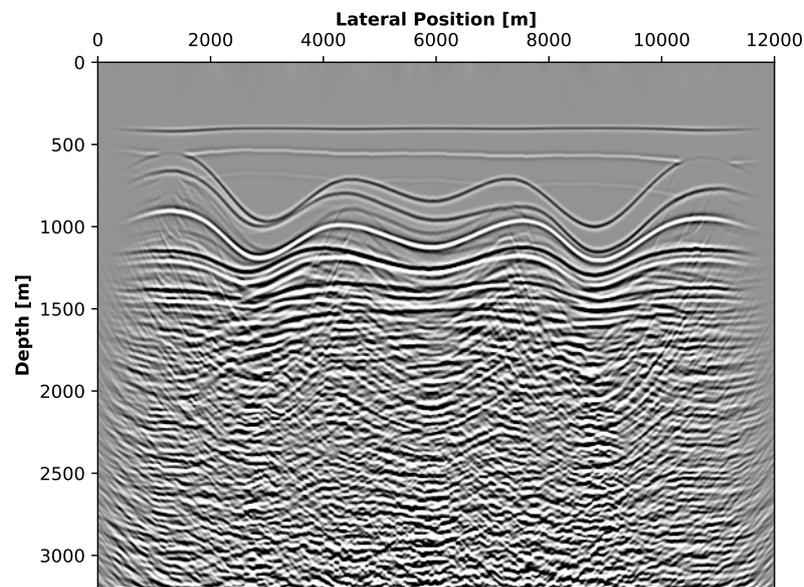


Figure 6.3: A comparison between (a) the RTM-based horizontal plane-wave image, and (b) the Marchenko-based horizontal plane-wave image. Both images are displayed using the same scaling factor. The red arrows indicate residual internal multiples that Marchenko plane-wave imaging method failed to suppress. Note the similarity in terms of character between the primary and multiple events.

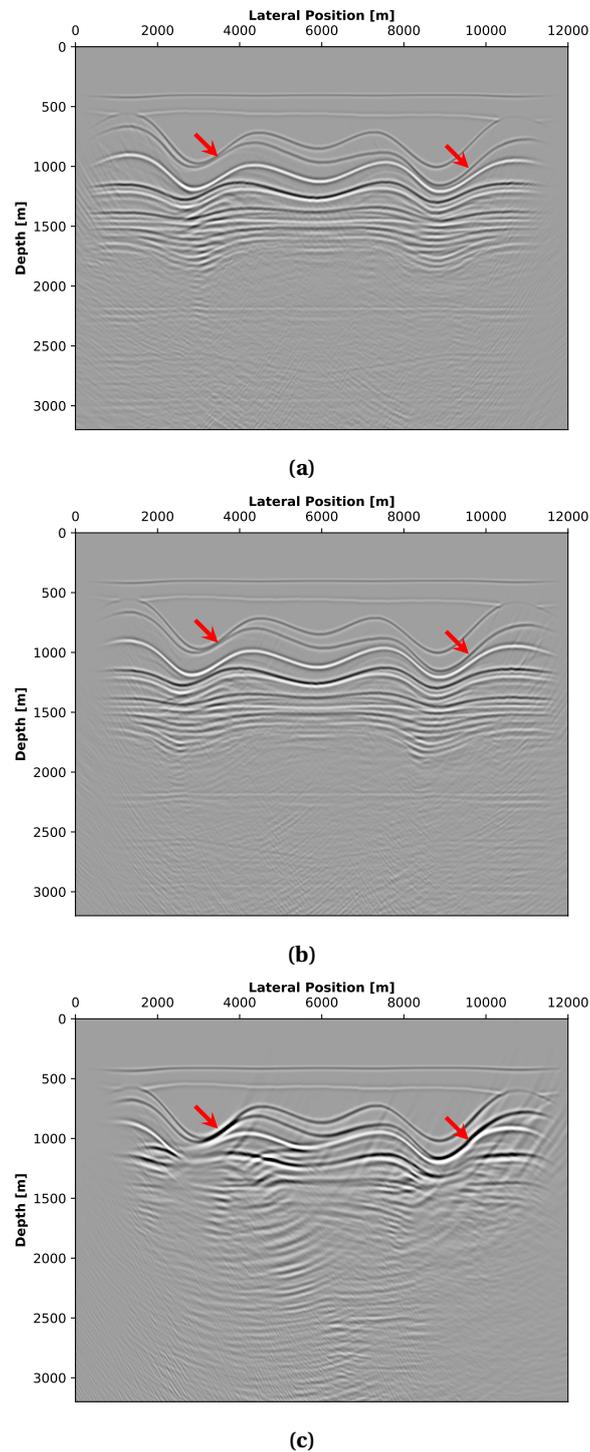


Figure 6.4: The Marchenko-based dipping plane-wave images associated with angles (a) -10° , (b) 10° , and (c) 25° . All images are displayed using the same scaling factor. The red arrows indicate the locations where the image associated with angle 25° shows degradation of the lateral resolution.

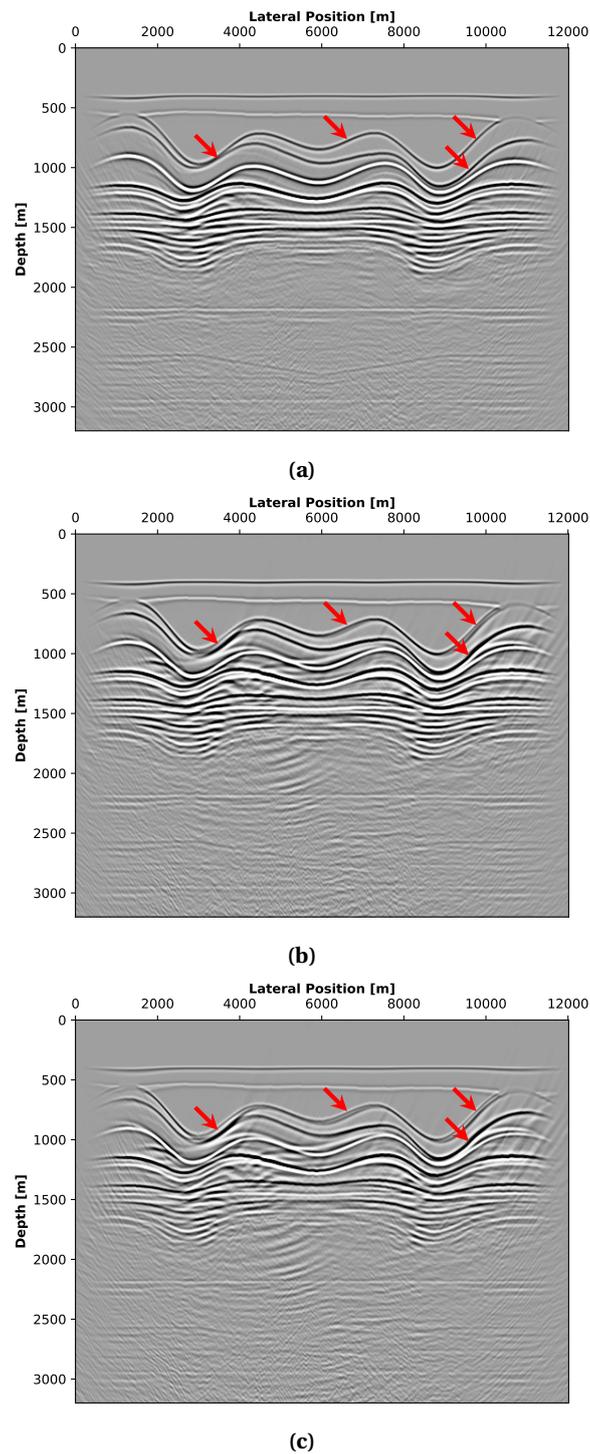


Figure 6.5: The stacking results of different Marchenko-based plane-wave images. (a) the stack of images associated with angles -10° , 0° , and 10° , (b) the stack of images associated with -10° , 0° , 10° , and 25° , and (c) the stack of images associated with angles 0° , and 25° . All images are displayed using the same scaling factor. The red arrows indicate the deterioration of the lateral resolution as a results of including the image associated with angle 25° .

7

APPLICATION IV: FIELD EXAMPLE

7.1 Dataset Description

The 2D seismic dataset under consideration was acquired in 1994 by SAGA Petroleum AS (currently Equinor ASA) over the Vøring basin, the North Sea. The acquisition parameters are summarized in [Table 7.1](#). The data was acquired using a moving spread, for which the active receivers move as the source moves, and only negative offsets were recorded. In addition to the reflection data, Equinor ASA provided a macro velocity model, which is shown in [Figure 7.1a](#).

Parameter	Value
Acquisition settings	Marine
Spread type	moving spread
Number of source positions	399
Source spacing	25m (after regularization)
First source position	5000m
Last source position	14950m
Number of receiver positions per source	180
Receiver spacing	25m
Minimum source-receiver offset	125m
Maximum source-receiver offset	4625m
Number of time samples	2001
Temporal sampling interval	0.004s
High-cut frequency	90Hz

Table 7.1: The acquisition parameters of the 2D SAGA dataset, acquired over the Vøring basin, the North sea.

7.2 Pre-processing Steps

The Vøring datasets suffers from a number of limitations for which corrections were needed in order to accommodate the Marchenko method requirements. The data were acquired with a receiver spacing of 12.5m for near offsets, and 25m for far offsets. The receiver spacing was regularized by summing every two near offset traces, resulting in a unified spacing of 25m. In addition, missing shots were interpolated prior to further processing. Furthermore, using a denser source-receiver spacing by means of interpolation has been tried, but it has failed to improve the performance of the Marchenko scheme ([Brackenhoff et al., 2019](#)).

The missing positive offsets were added to the shot gathers by applying source-receiver reciprocity. However, source-receiver reciprocity can only recover the positive offsets before the last source

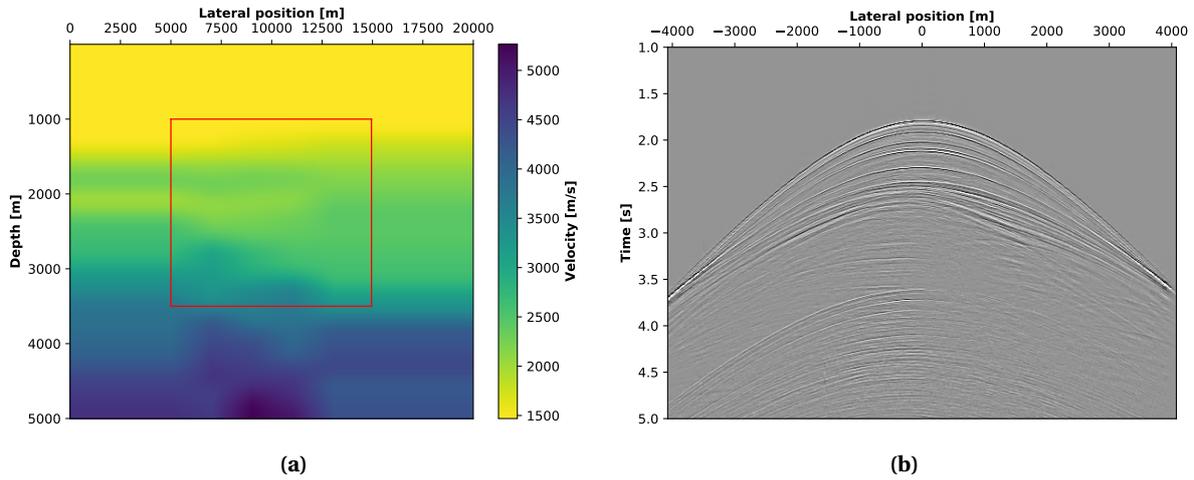


Figure 7.1: (a) The macro velocity model provided by Equinor ASA for the Vøring dataset. The red box indicates the imaged area of interest. (b) A pre-processed shot gather from the middle of the 2D line.

position, as represented by yellow region in [Figure 7.2](#). Note that to have a unified number of traces per shot, dummy traces were added where source-receiver reciprocity cannot be used, as represented by the blue region.

As is typical for marine seismic acquisition, the Vøring dataset is missing 125m of the near-offsets. However, having both positive (created) and negative (acquired) offsets allowed for the reconstruction of the missing near offsets by applying parabolic Radon interpolation ([Kabir and Verschuur, 1995](#)).

The seabed depth at the acquisition location is estimated to be around 1300m. Hence, surface multiples are well separated from the shallow reflection data ([Davydenko and Verschuur, 2018](#)). Although surface multiples do not affect the shallow depths, the Surface-Related Multiple Elimination (SRME) scheme was applied with adaptive subtraction to avoid complications at larger depths. In addition, the effective wavelet was estimated using a modified version of the Estimation of Primaries by Sparse Inversion (EPSI) method ([Ypma and Verschuur, 2013](#)), which was then deconvolved using SRME. Furthermore, predictive deconvolution was used to reduce the air-gun bubble effect.

[Brackenhoff et al. \(2019\)](#) described how the 3D spherical divergence for 2D data was corrected for by applying a time-dependent gain with a factor of \sqrt{t} . Unlike the spherical divergence scaling, the absorption scaling factor was a subject of thorough testing. As mentioned in [section 2.4](#), the Marchenko iterative scheme is energy-minimizing, and hence, a scaling factor leads to convergence when subsequent iterations show a decreasing energy. A rule of thumb is that when the relative energy of the last iteration is in the order of $1e^{-3}$ to $1e^{-4}$ relative to the first iteration, then the iterative scheme has converged and the Marchenko solutions are most likely stable ([Thorbecke, 2020](#)). However, as multiple scaling factors can lead to the convergence of the iterative scheme, another criterion needs to be considered. The second criterion is chosen to be the strength of the coda of the focusing functions. This criterion assures that the focusing functions are not dominated by their initial estimation, and the Marchenko scheme is sufficiently able to account for internal multiples. Several scaling factors were tested by running the Marchenko iterative scheme with 20 iterations for several depths within the depth range of interest. The scaling factor of $7.2e^t$ was chosen as it best satisfied both concerns, a converging and stable performance of the Marchenko iterative scheme, and a strong amplitude of the coda of the focusing functions. Refer to [Appendix C](#) for the full analysis that lead to this scaling factor. An example of a fully processed and scaled shot gather is shown in [Figure 7.1b](#).

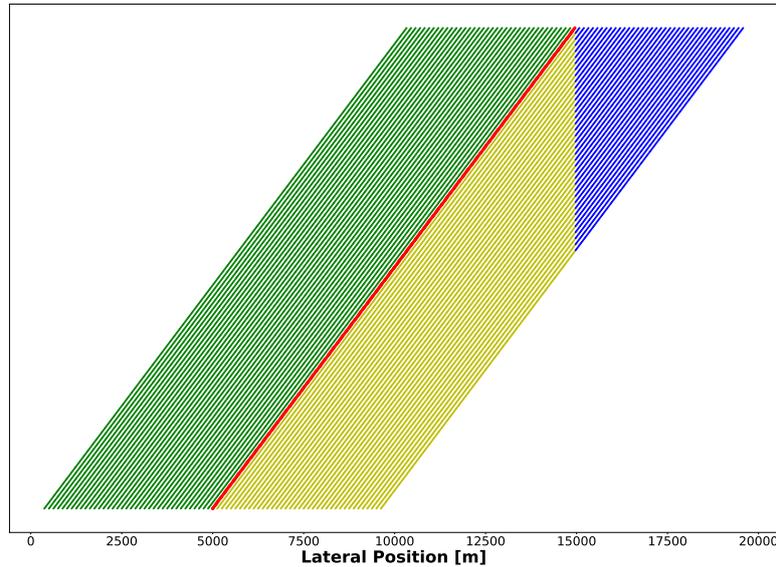


Figure 7.2: The source and receiver locations of the Vøring dataset used to apply the plane-wave Marchenko imaging method. The red stars represent the shot location. The green dots represent the acquisition receiver locations. The yellow dots represent the receiver locations added by source-receiver reciprocity. The blue dots represent the dummy traces added to unify the number of traces per shot location.

7.3 Imaging Results

Several authors reported the presence of internal multiples in the Vøring dataset and used different techniques to eliminate them, including [Verschuur and Berkhout \(2005\)](#) who used the common-focus-point (CFP) approach, and [Zhang \(2019\)](#) who used the Marchenko Multiple Elimination method (MME). Furthermore, by retrieving the homogeneous Green's function for the Vøring dataset, [Brackenhoff et al. \(2019\)](#) found that up- and down-going low energy multiples are present and they are strongly correlated with layer boundaries.

Based on the prior knowledge of the subsurface reflector geometry, the Marchenko plane-wave imaging workflow was repeated for five incident angles, ranging from -10° to 10° inclusive, with an increment of 5° . For each angle, the initial plane-wave focusing functions were modelled via finite difference extrapolation, with a depth increment of 5m and within the area of interest, marked by the red box in [Figure 7.1a](#). This area of interest ranges between 1000m and 3500m depth. Note that due to moving spread acquisition, the number of receivers is larger than the number of sources. Hence, the initial plane-wave focusing functions were modelled such that they take into account all receiver positions, between 375m and 19575m. However, as dictated by the Marchenko method requirements (refer to [section 2.5](#)), the Marchenko solutions were only retrievable where sources and receivers coincide, between 5000m and 14950m. Consequently, all the plane-wave images produced are limited to this range.

The supplied macro velocity model ([Figure 7.1a](#)) was used to migrate the angle-dependent plane-wave Marchenko results. The angle-dependent images suffer from imaging artefacts, which were partially attenuated using a dip filter. Upon analyzing the individual images, an angle-dependent upward shift in depth was noticed. This shift is not explainable and not accounted for by the implemented imaging workflow. Therefore, the individual images were manually shifted down in an angle-dependent manner such that all dipping plane-wave images have the same depth as the horizontal plane-wave image. The angle-dependent vertical shift correction was found to be:

$$t = t_0 - \max(\mathbf{p} \cdot \mathbf{x}_{H,B}) \cos(\text{angle}) - \epsilon, \quad (7.1)$$

where t_0 is the length of the down-going plane-wave Green's function used as an input to the back-propagation process, $\max(\mathbf{p} \cdot \mathbf{x}_{H,B})$ is the maximum dip of the plane-wave in time, $\cos(\text{angle})$ is the

cosine of the incident angle, and ϵ is half the wavelet width in time, included to extract the maximum event energy.

Then, the individual images were stacked to form the final image that accounts for the variable dip of the deep reflectors. Similarly, RTM-based plane-wave images were generated using the same incident angles range by limiting the Marchenko iterative scheme to a single iteration.

Figure 7.3 shows the comparison between RTM-based and the Marchenko-based plane-wave image stacks for the area of interest. The green and red boxes mark areas where the difference is most apparent, and are shown separately in Figure 7.4 and Figure 7.5. Figure 7.4 shows that internal multiples, indicated by the red ellipsoid and arrow, are partially suppressed by the plane-wave Marchenko method, this multiple suppression is manifested by a reduction in amplitude. Figure 7.5 shows a second advantage of the Marchenko-based image over the RTM-based image. The plane-wave Marchenko imaging methods successfully managed to suppress internal multiples that are cross cutting more steeply dipping events, as indicated by the red arrows, improving the continuity of the subsurface reflectors. In addition, the method managed to suppress destructively interfering multiples, which allowed for the recovery of primary events, as indicated by the green arrows.

Upon investigating the cause for the angle-dependent depth shift, it was observed that back-propagation by finite difference extrapolation led to the formation of turning waves, such that as the plane-wave Green's functions travel to their corresponding depths within the medium, their propagation direction changes and their dip steepens. Consequently, to test if the turning waves had any effect on the extracted images, back-propagation by cross-correlation was performed. The cross-correlation process involved modelling the direct arrivals of the point-source Green's functions for each source location and for each imaging depth via an Eikonal solver. With reference to Equation 3.18, the two back-propagation approaches allow for the extraction of the same subsurface events depth-wise. However, as the Eikonal solver is used to solely model the travel time of the direct arrivals, back-propagation by cross-correlation is not affected by turning waves. Prior to stacking, a similar angle-dependent upward shift was observed, and was corrected for by manually aligning the individual images.

Figure 7.6 shows a comparison between the Marchenko-based plane-wave image stacks produced by the two back-propagation approaches. A number of observations are obtained from this comparison. Firstly, the two images have different amplitude ranges, which is attributed to the amplitude scaling imposed by the Eikonal solver. Hence, a unified scaling factor cannot be used to display them coherently. Secondly, the two images have similar imaging performance up to 2100m depth. Below this depth, the image associated with back-propagation by cross-correlation shows higher amplitudes at the sides of the image, as indicated by the red boxes, which enhances the definition of the reflectors within these zones. At the middle, the cross-correlation image is lower in amplitude, especially at larger depths. In addition, residual free-surface multiples are visible in the cross-correlation image between the depths 3000m and 3500m, as indicated by the red arrows.

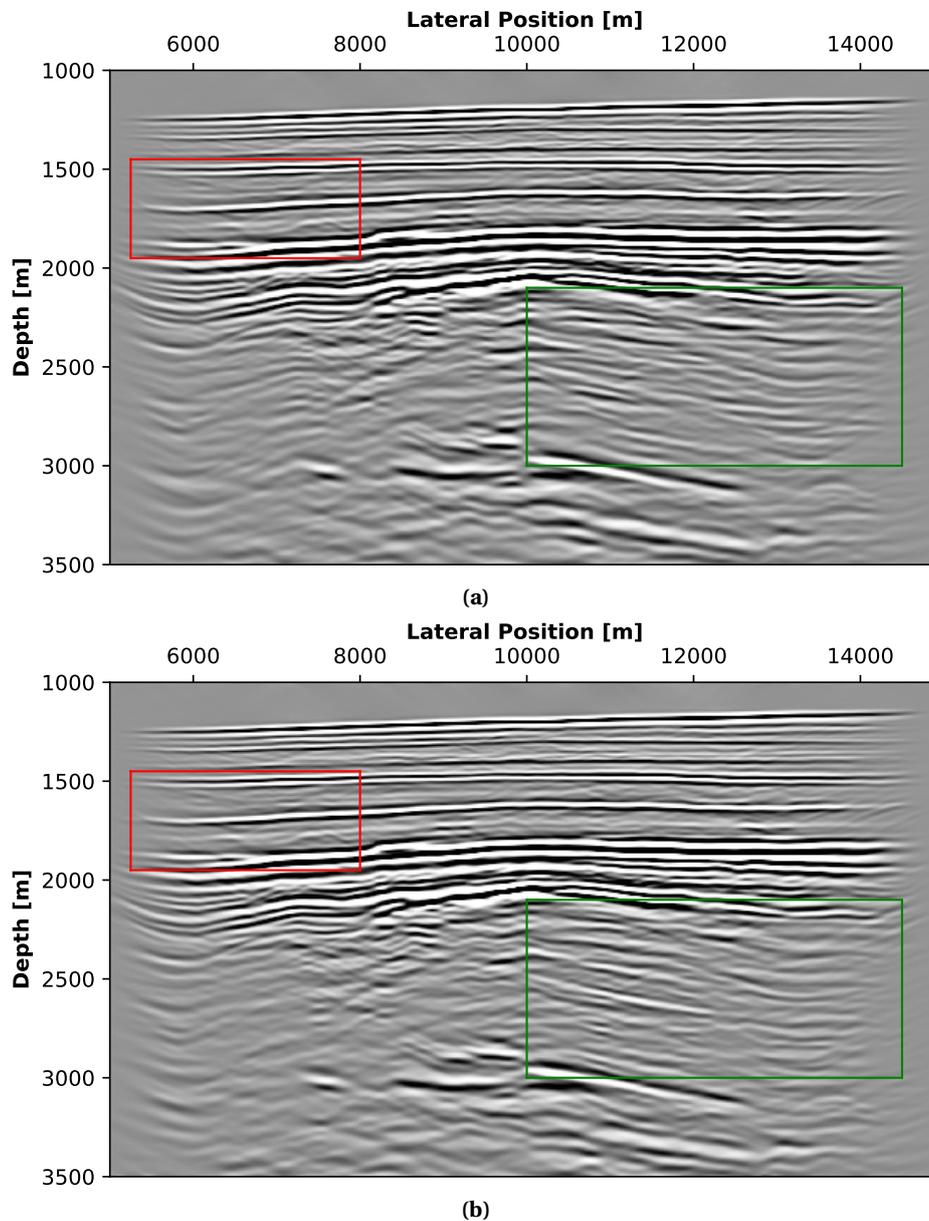


Figure 7.3: The imaging results of the SAGA A.S. dataset over the Vøring basin, the North Sea. (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. Both images are the results of summing plane wave images with incident angles ranging from -10° to 10° with an increment of 5° . Images are scaled using the same scaling factor. The red and green boxes highlight areas that are magnified in [Figure 7.4](#) and [Figure 7.5](#) for a more detailed comparison.

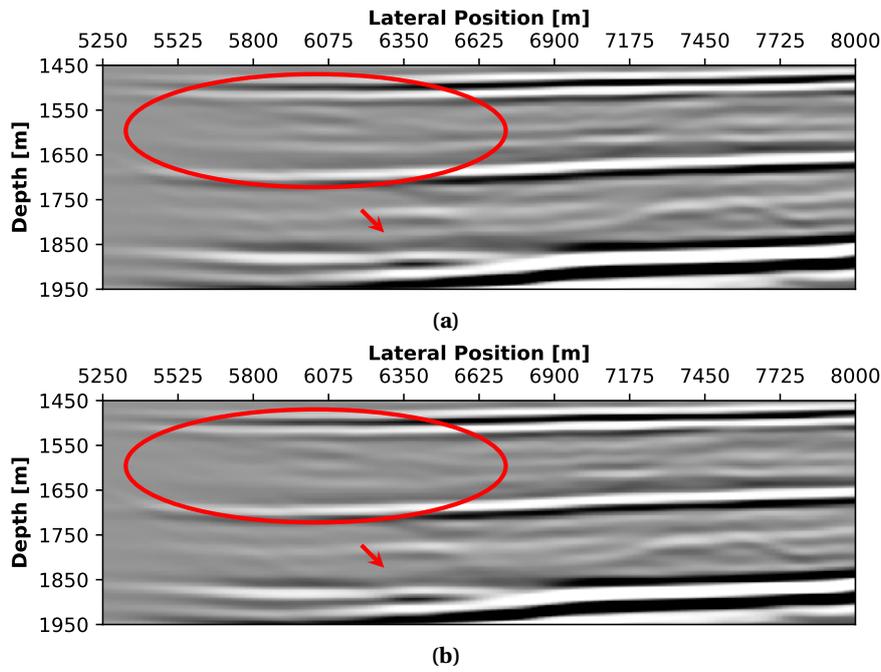


Figure 7.4: Magnification of the red box in [Figure 7.3](#). (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. The red ellipsoid and arrow indicate internal multiples that have been suppressed by the plane wave Marchenko imaging method. All images are scaled using the same scaling factor.

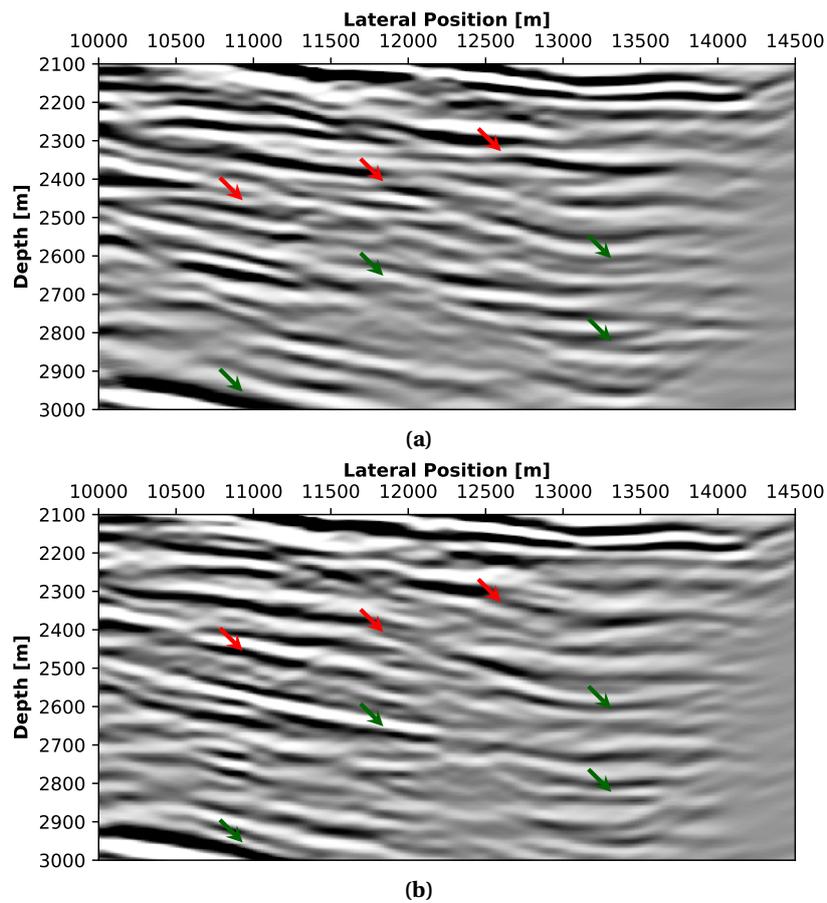


Figure 7.5: Magnification of the green box in [Figure 7.3](#). (a) the RTM-based plane wave image, (b) the Marchenko-based plane wave image. The red arrows indicate internal multiples that have been suppressed by the plane wave Marchenko imaging method. The green arrows indicate events that were recovered by suppressing destructively interfering internal multiples. All images are scaled using the same scaling factor.

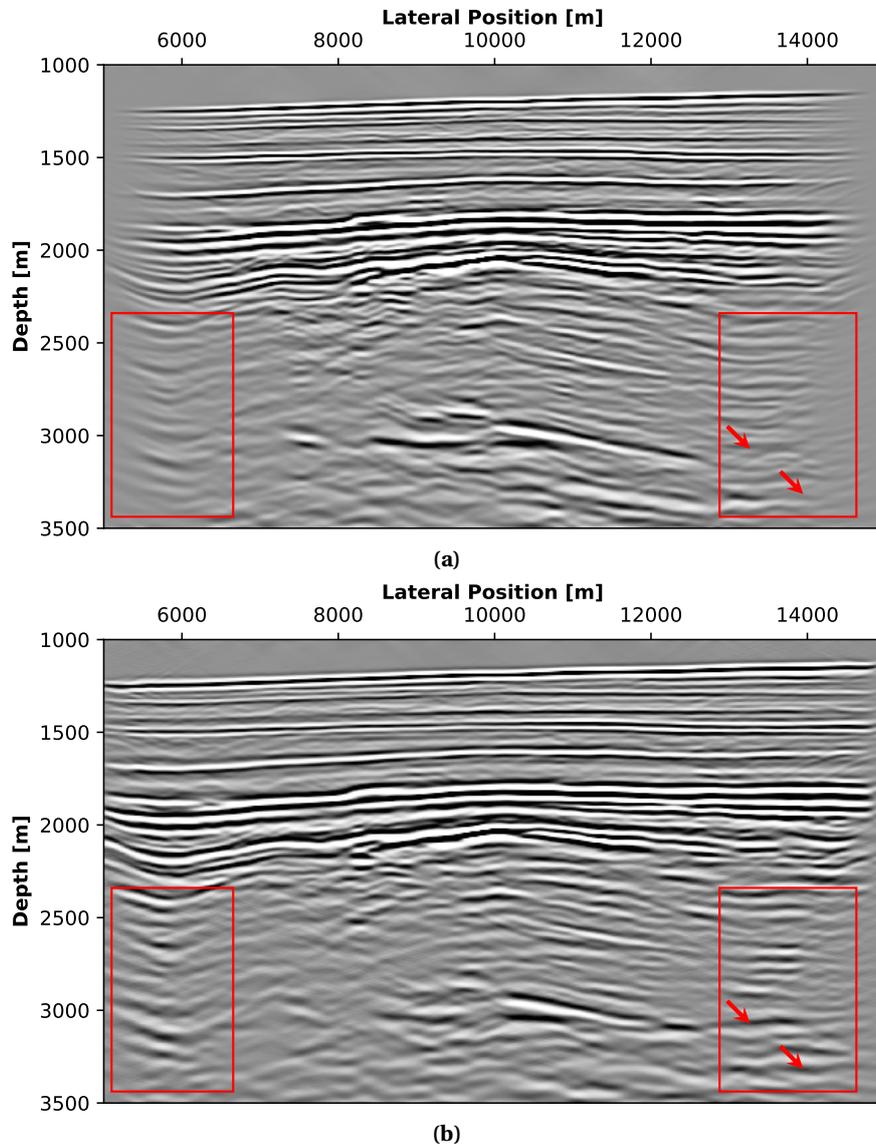


Figure 7.6: The image stacks associated with (a) back-propagation by finite difference extrapolation, and (b) back-propagation by cross-correlation. The red boxes indicate edges of the image for which the amplitude of the cross-correlation image is higher than that of the finite difference extrapolation image. The red arrows indicate the residual free-surface multiples present in the cross-correlation image.

8

DISCUSSION AND CONCLUSION

It was shown that the plane-wave Marchenko imaging method can be used to extract multiple-free images of the subsurface. Similar to the conventional Marchenko imaging method, the plane-wave Marchenko imaging method requires single-sided surface reflection data and a macro velocity model. The velocity model is used to model the direct arrival from each imaging depth to the surface, either by finite difference modelling or by an Eikonal solver. The plane-wave responses are modelled by exciting time delayed horizontally aligned point sources. The time delay allows for the formation of dipping plane wavefronts. The direct arrivals of the modelled plane-wave responses are used as an input to the Marchenko iterative scheme, allowing for the retrieval of the up- and down-going plane-wave Green's functions, which represent the wavefields associated with a plane-wave virtual source in the subsurface.

For plane-wave Marchenko solutions, imaging is performed by cross correlating the down-going plane-wave Green's functions with modelled point-source depth-specific up-going Green's functions prior to applying the imaging condition, as stated by [Equation 3.18](#). Ideally, the point source Green's functions are modelled using finite difference modelling. However, this can be a computationally expensive process as nx times nz Green's functions need to be modelled to produce a complete image. Alternatively, the down-going plane-wave Green's functions can be back-propagated to their corresponding depths by finite difference extrapolation. While this approach maintains the amplitude accuracy of the latter back-propagation approach, it can reduce the computational cost as with this approach performing the cross-correlation is not needed and is accounted for in the extrapolation process. The cost reduction is subjected to the availability of computational resources and requires future verification. A third back-propagation approach is cross-correlating the down-going plane-wave Green's functions with point-source up-going Green's functions modelled via an Eikonal solver. This approach reduces the computational expense significantly. However, this cost reduction is subject to computational resource availability and code design. Based on our work, the cost was reduced by a factor of four, in comparison to back-propagation by finite difference extrapolation. The disadvantage of this approach is inaccuracy of the Eikonal solver outputs. Consequently, this inaccuracy propagates to the retrieved images. Refer to [Figure 3.5](#) for the complete plane-wave imaging workflow.

As previous studies, such as the work of [Broggini et al. \(2014\)](#), have stated, Marchenko-based images obtained by Multi-Dimensional Deconvolution (MDD) reflect the true amplitude of the subsurface events more accurately than images obtained by cross correlation. In terms of the amplitude accuracy, the superiority of MDD over cross-correlation is attributed to the fact that MDD eliminates the transmission effects, while cross-correlation does not. As MDD is not applicable to plane-wave solutions, amplitude errors are always embedded within the plane-wave images. Furthermore, as MDD is not applicable to plane-wave solutions, additional errors in the retrieved Green's functions, such as errors associated with the usage of an Eikonal solver for modelling the initial focusing functions, cannot be eliminated, which further degrades the amplitude accuracy of the extracted images.

Angle-dependent plane-wave images can suffer from three possible limitations: illumination limitations, residual internal multiple energy, and imaging artefacts. The illumination limitation is attributed to the limited lateral extent of the receivers array. The physical explanation of this phenomenon is as follows: depending on the difference between the plane-wave incident angle and the dip of the subsurface reflectors, a down-going wavefield associated with a plane-wave virtual sources reflects from a subsurface reflector and may arrive to the surface at locations outside the receivers array, and hence it is not completely recorded at the surface. Consequently, back-propagating an incomplete down-going Green's functions leads to discontinuities at the imaged depth.

The design of the acquisition parameters plays an important role in the quality of the redatumed data, as sufficiently wide aperture is needed to properly reconstruct all events in the subsurface from the reflection data. Hence, for a limited aperture, the reflection data does not contain the entire reflected wavefield, which leads to suboptimal performance of the Marchenko scheme. Consequently, for such cases, the plane-wave images can contain residual internal multiple energy. This type of artefacts is not specific to the plane-wave Marchenko imaging method, and is inherited from the conventional Marchenko method. In addition, it occurs in any other internal multiple elimination method.

The presented results show that the produced images are contaminated by imaging artefacts. These artefacts are not limited to images associated with back-propagation by finite difference extrapolation, and are also observed for images associated with back-propagation by cross-correlation. The orientation of the imaging artefacts is perpendicular to the propagation direction of the plane-waves. In addition, the frequency of these artefacts scales by $\cos(\alpha)$ with α being the plane-wave incident angle, such that as incident angle increases, the frequency of the imaging artefacts decreases. Dip filters can be used to eliminate these artefacts, taking advantage of the difference in dip between the imaging artefacts and the subsurface events. However, filtering should be done with caution as it can damage the amplitude of the imaged events. The mechanism by which these artefacts are generated is subject to further investigation.

In theory, stacking Marchenko-based plane-wave images with an infinite number of incident angles is equal to an image produced by the conventional Marchenko method in terms of resolution and amplitude accuracy. This equality requires the conventional Marchenko imaging to be performed by cross correlation, rather than by MDD. However, for highly varying dip of the subsurface reflectors, it was shown that relatively steeply dipping plane-wave images are affected by lateral resolution degradation for parts of the reflectors that have opposite dip to that of the plane-wave virtual source. This degradation increases in severity as the incident angle increases, and is also present in the final image after stacking. As an alternative to stacking the individual angle-dependent plane-wave images directly, zones of poor lateral resolution can be improved by specifically imaging them using the conventional point-source Marchenko imaging method, and then stack the zone-specific images to the horizontal plane-wave image. Another approach is to truncate parts of the dipping plane-wave images that are not degraded resolution-wise and stack them to the horizontal plane-wave image. Therefore, either of the alternative stacking approaches should be able to meet the resolution of the conventional Marchenko imaging method. However, the amplitude uniformity of the conventional Marchenko images is more difficult to achieve. Consequently, it is clear from the presented results that the plane-wave Marchenko method is suitable for structural interpretations. However, more research is required to determine if the method is also suitable for amplitude sensitive applications.

Applying the plane-wave Marchenko imaging method to the Vøring dataset has proven to be successful. In comparison to the RTM-based image, the Marchenko-based stack image shows that energy related to internal multiples is attenuated, and the amplitude and continuity of the primary events are improved. The nature and locations of the suppressed multiples are comparable to results published by [Zhang \(2019\)](#) who used the Marchenko Multiple Elimination (MME) method to suppress multiples prior to imaging. While the two results are Marchenko based, the images presented in this thesis are worse, especially for relatively steep events. Due to the difference in amplitude correction applied to the reflection data, a direct comparison between the two results is

meaningless, even though some observations can still be made. Consequently, to properly compare the two methods, the pre-processing workflow, including the amplitude scaling of the reflection data, has to be the same. Another difference is that for our results, the restricted range of incident angles caused limited illumination, especially for deep reflectors. Hence, potentially by stacking more incident angles, our results can be improved to better match the MME-based results. An attempt was made to create images associated with incident angles $> 10^\circ$ using back-propagation both by finite difference extrapolation and cross-correlation with point-source responses modelled via an Eikonal solver. Unfortunately, the adopted imaging workflow could not produce coherent images for these large angles. Based on our preliminary investigation, we suspect that turning waves were formed when back-propagation by finite difference extrapolation was applied. However, the same imaging problems were observed when back-propagation by cross correlation with point-source Green's functions was applied. Therefore, a detailed investigation of the cause of the observed angle-dependent imaging problems is needed, with a focus on the pre-processing workflow applied to the reflection data. Furthermore, it is important to mention that to assess the accuracy of the Vøring results obtained by either method, an independently acquired dataset, such check-shots or VSP profiles, is needed.

Future work should compare the imaging performance of the conventional Marchenko imaging methods, the Marchenko-based plane-wave imaging method, and the Marchenko Multiple Elimination method when applied on the Vøring dataset. In addition, applying the Marchenko-based plane-wave imaging method on another field dataset could reveal if the observed imaging problems are dataset-specific, and could potentially develop the established workflow further to account for any additional problems that might be observed.

REFERENCES

- Bakulin, A. and Calvert, R. (2006). The virtual source method: Theory and case study. *Geophysics*, 71(4):SI139–SI150.
- Berkhout, A. (1982). *Imaging of acoustic energy by wave field extrapolation*. Elsevier Amsterdam.
- Berryhill, J. R. (1984). Wave-equation datuming before stack. In *SEG Technical Program Expanded Abstracts 1984*, pages 397–399. Society of Exploration Geophysicists.
- Born, M. and Wolf, E. (1965). *Principles of optics*. Pergamon Press, London.
- Brackenhoff, J., Thorbecke, J., and Wapenaar, K. (2019). Virtual Sources and Receivers in the Real Earth: Considerations for Practical Applications. *Journal of Geophysical Research: Solid Earth*, 124(11):11802–11821.
- Broggini, F., Wapenaar, K., van der Neut, J., and Snieder, R. (2014). Data-driven Green's function retrieval and application to imaging with multidimensional deconvolution. *Journal of Geophysical Research: Solid Earth*, 119(1):425–441.
- Davydenko, M. and Verschuur, D. J. (2017). Full-wavefield migration: using surface and internal multiples in imaging. *Geophysical Prospecting*, 65(1):7–21.
- Davydenko, M. and Verschuur, D. J. (2018). Including and using internal multiples in closed-loop imaging — Field data examples. *Geophysics*, 83(4):R297–R305.
- Fink, M. (1999). Time-Reversed Acoustics. *Scientific American*, 281(5):91–97.
- Kabir, M. M. and Verschuur, D. J. (1995). Restoration of missing offsets by parabolic Radon transform. *Geophysical Prospecting*, 43(3):347–368.
- Meles, G. A., Wapenaar, K., and Thorbecke, J. (2018). Virtual plane-wave imaging via Marchenko redatuming. *Geophysical Journal International*, 214(1):508–519.
- Minato, S., Matsuoka, T., and Tsuji, T. (2013). Singular-value decomposition analysis of source illumination in seismic interferometry by multidimensional deconvolution. *Geophysics*, 78(3):Q25–Q34.
- Rietveld, W. E., Berkhout, A. J., and Wapenaar, C. P. (1992). Optimum seismic illumination of hydrocarbon reservoirs. *Geophysics*, 57(10):1334–1345.
- Rose, J. H. (2001). “Single-sided” focusing of the time-dependent schrödinger equation. *Physical Review A*, 65(1):012707.
- Schuster, G. (2009). *Seismic interferometry*. Cambridge university press.
- Slob, E., Wapenaar, K., Broggin, F., and Snieder, R. (2014). Seismic reflector imaging using internal multiples with Marchenko-type equations. *Geophysics*, 79(2):S63–S76.
- Thorbecke, J. (2020). personal communication.
- Thorbecke, J., Slob, E., Brackenhoff, J., van der Neut, J., and Wapenaar, K. (2017). Implementation of the marchenko method. *Geophysics*, 82(6):WB29–WB45.

- van der Neut, J., Johnson, J. L., van Wijk, K., Singh, S., Slob, E., and Wapenaar, K. (2017). A Marchenko equation for acoustic inverse source problems. *The Journal of the Acoustical Society of America*, 141(6):4332–4346.
- van der Neut, J., Vasconcelos, I., and Wapenaar, K. (2015). On Green's function retrieval by iterative substitution of the coupled Marchenko equations. *Geophysical Journal International*, 203(2):792–813.
- Verschuur, D. J. and Berkhout, A. J. (2005). Removal of internal multiples with the common-focus-point (CFP) approach: Part 2 - Application strategies and data examples. *Geophysics*, 70(3):61–72.
- Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.
- Wapenaar, K., Brackenhoff, J., Dukalski, M., Meles, G., Reinicke, C., Slob, E., Staring, M., Thorbecke, J., van der Neut, J., and Zhang, L. (2021). Marchenko redatuming, imaging and multiple elimination, and their mutual relations. *Geophysics*, 86(5).
- Wapenaar, K., Broggini, F., and Snieder, R. (2012). Creating a virtual source inside a medium from reflection data: heuristic derivation and stationary-phase analysis. *Geophysical Journal International*, 190(2):1020–1024.
- Wapenaar, K. and Thorbecke, J. (2017). Review paper: Virtual sources and their responses, Part I: time-reversal acoustics and seismic interferometry. *Geophysical Prospecting*, 65(6):1411–1429.
- Wapenaar, K., Thorbecke, J., van der Neut, J., Broggini, F., Slob, E., and Snieder, R. (2014a). Green's function retrieval from reflection data, in absence of a receiver at the virtual source position. *The Journal of the Acoustical Society of America*, 135(5):2847–2861.
- Wapenaar, K., Thorbecke, J., Van der Neut, J., Broggini, F., Slob, E., and Snieder, R. (2014b). Marchenko imaging. *Geophysics*, 79(3):WA39–WA57.
- Wapenaar, K., Thorbecke, J., van der Neut, J., Slob, E., and Snieder, R. (2017). Review paper: Virtual sources and their responses, Part II: data-driven single-sided focusing. *Geophysical Prospecting*, 65(6):1430–1451.
- Wapenaar, K., Van Der Neut, J., Ruigrok, E., Draganov, D., Hunziker, J., Slob, E., Thorbecke, J., and Snieder, R. (2011). Seismic interferometry by crosscorrelation and by multidimensional deconvolution: A systematic comparison. *Geophysical Journal International*, 185(3):1335–1364.
- Ypma, F. H. and Verschuur, D. J. (2013). Estimating primaries by sparse inversion, a generalized approach. *Geophysical Prospecting*, 61(SUPPL.1):94–108.
- Zhang, L. (2019). *Acoustic multiple reflection elimination in the image domain and in the data domain*. PhD thesis, Delft University of Technology.

A

MARCHENKO-RELATED PLANE WAVEFIELD

For subsurface models characterized by laterally varying dip, dipping plane-waves may be needed to account for any image discontinuities. The schematic illustrations presented in [Figure A.1](#) aim to shed a light on the the dip of different Marchenko-related plane wavefields.

For an arbitrary positive angle, the direct arrival of the plane-wave Green's function \tilde{G}_d is modelled by placing a plane-wave source at the focal depth \mathbb{S}_A and receivers at the acquisition surface \mathbb{S}_0 , as shown in [Figure A.1a](#). The time-reversed version of \tilde{G}_d is considered the initial focusing function \tilde{f}_{1d}^+ , according to the plane-wave version of the direct arrival assumption expressed in [Equation 2.12](#). Due to the time-reversal, \tilde{f}_{1d}^+ has an opposite dip to the modelled \tilde{G}_d , as shown in [Figure A.1b](#). Naturally, the down-going focusing function \tilde{f}_1^+ has the same dip as its initial estimation \tilde{f}_{1d}^+ . By definition, the up-going focusing function \tilde{f}_1^- is the reflection response of down-going focusing function \tilde{f}_1^+ , and hence it has the opposite dip in space. However, in time, both functions have the same dip direction. The down-going plane-wave Green's function $\tilde{G}^{-,+}$, obtained from solving the Marchenko iterative scheme, and has the same dip as the focusing functions \tilde{f}_1^+ and \tilde{f}_1^- , and is illustrated in terms of receiver-redatuming as shown in [Figure A.1d](#) or in terms of source-redatuming as shown in [Figure A.1e](#). By comparing [Figure A.1a](#) to [Figure A.1d](#), it can be seen that the dip of the modelled direct arrival of the plane-wave Green's function \tilde{G}_d is the same as the dip of response of the plane-wave virtual source $\tilde{G}^{-,+}$, which contributes to the ease of use of the plane-wave Marchenko imaging method.

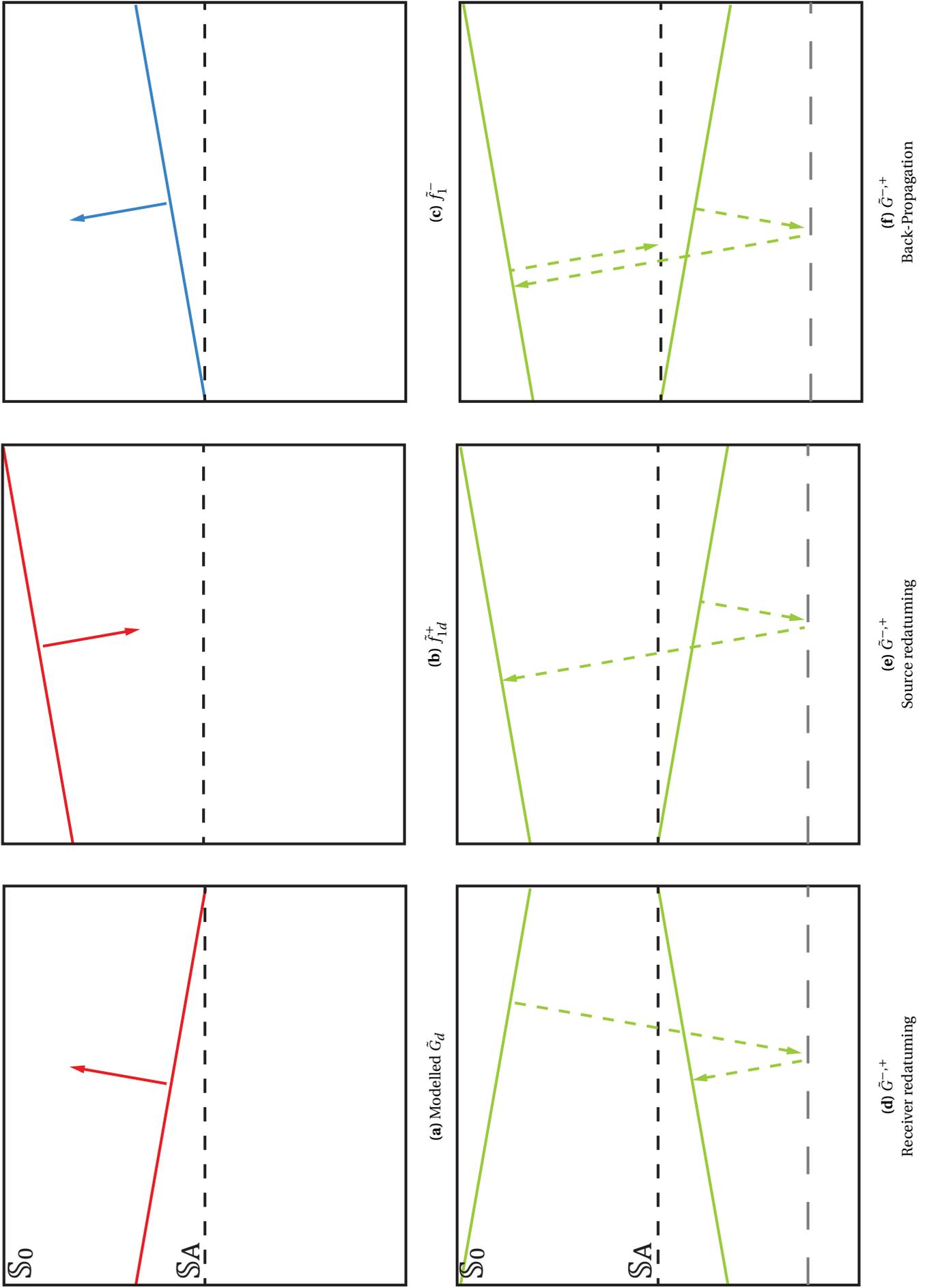


Figure A.1: A schematic illustration of the dip of different Marchenko-related plane wavefields.

B

$\tilde{\mathbf{G}}^{-,+}$ TRUNCATION PRIOR TO BACK-PROPAGATION

Although it reduces the computational cost, truncating the plane-wave Green's function $\tilde{\mathbf{G}}^{-,+}$ prior to back-propagation has a possible effect on the retrieved image. The effect is manifested by discontinuities in the imaged events, as shown in [Figure B.1](#). As the truncation starting time and orientation are chosen arbitrarily, applying the truncation is not recommended, especially for field datasets as the effect of the truncation cannot be identified. The model used for this test is fully described in [chapter 5](#).

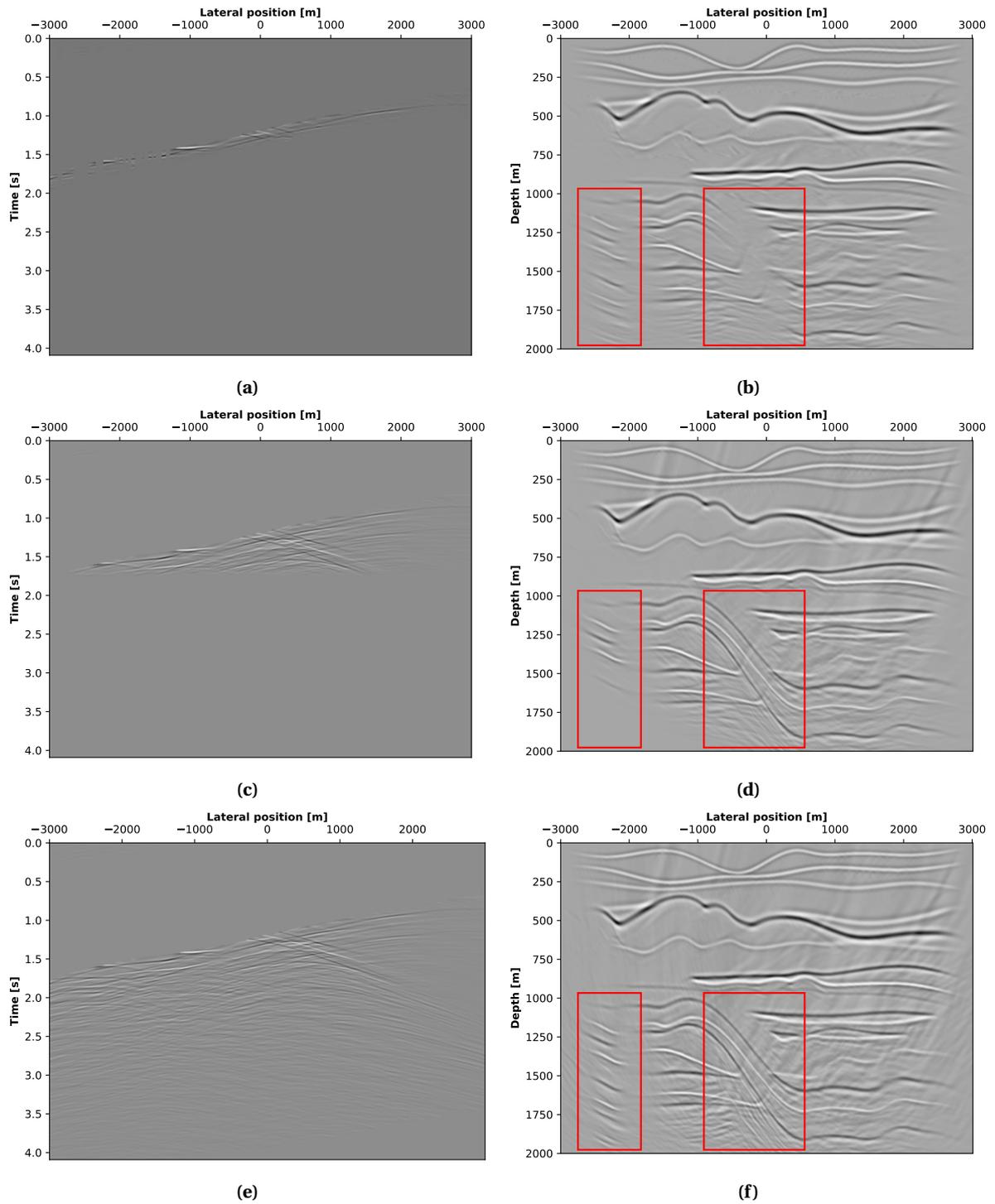


Figure B.1: The truncation of the down-going plane-wave Green's function $\tilde{G}^{-,+}$ associated with a plane-wave incident angle of 15° , using a truncation time that is (a) harsh and parallel to the first event, and (d) not parallel to the first event, and (e) not truncated. The images corresponding to (a), (c), and (e) are displayed in (b), (d), and (f), respectively. The red boxes indicate the part of the imaged events affected by the different truncation designs.

C

SCALING TESTS FOR THE VØRING DATASET

As the wavefield propagates through the medium, a frequency-dependent absorption occurs, which alters the amplitude of the recorded reflection data. Considering the impact of the accuracy of the reflection data on the convergence of the Marchenko iterative scheme, the absorption effect needs to be compensated for. Several scaling factors were tested by running the Marchenko scheme for 25 iterations, and computing the plane-wave focusing functions for two depths: 1500m and 3000m. To assess the performance of the iterative scheme, two criteria were considered: convergence and stability of the solutions, and the amplitude strength of the coda of the focusing functions. The convergence criterion is defined by a continuous decrease of energy for subsequent iterations relative to the first iteration. Furthermore, as a rule of thumb, the Marchenko solutions are considered stable when the relative energy is in the order of $1e^{-3}$ to $1e^{-4}$. The second criterion requires sufficiently high amplitude of the coda to assure that the focusing functions are not dominated by their initial estimation, and the Marchenko scheme is capable of accounting for internal multiples. [Table C.1](#) lists 4 parameter pairs that were selected out of 45 scaling experiments. [Figure C.1](#) shows the focusing functions associated with the listed scaling factors for depths 1500m and 3000m. This figure shows three classes of focusing functions: focusing functions dominated by the initial estimation, focusing functions with anomalous amplitude, focusing functions with proper focusing performance. The first class is represented by [Figure C.1 \(a\)](#) and [\(b\)](#), which were computed using a scaling factor of 12. For this test, the scaling factor led to poor focusing performance as the focusing functions are dominated by their initial estimation. The poor performance is also indicated by how fast the iterative scheme met the convergence and stability criterion. Consequently, this scaling factor, and others with poor focusing performance were rejected.

The second class is represented by [Figure C.1 \(c\)](#) and [\(d\)](#), which were computed using a scaling factor of $8.5e^t$. For this test, the used scaling factor focusing performance led to good focusing performance for $z = 1500\text{m}$. However, for $z = 3000\text{m}$, the iterative scheme did not converge, and hence amplitude anomalies are present in the focusing function, as indicated by the red boxes. As a result, this scaling factor, and others with anomalous coda amplitudes were rejected.

[Figure C.1 \(e\)](#), [\(f\)](#), [\(g\)](#), and [\(h\)](#), which were created using the scaling factors $9e^{0.9t}$ and $7.2e^t$ respectively, represent the third class of proper focusing performance. For $z = 1500\text{m}$, the focusing performance is very comparable for the two scaling factors. However, for $z = 3000\text{m}$, the coda associated with the scaling factor $7.2e^t$ has a higher amplitude in comparison to $9e^{0.9t}$. This test suggests that although multiple scaling factors can meet the aforementioned criteria, the amplitude of the coda differs. Consequently, the scaling factor $7.2e^t$ was chosen as it led to the highest amplitude of the coda, among all the tested scaling factors with proper focusing performance.

Scale	Depth	Relative energy in the order of $1e^{-3}$ after iteration number
12	1500	3
	3000	3
$8.5e^t$	1500	5
	3000	NA
$9e^{0.9t}$	1500	3
	3000	22
$7.2e^t$	1500	4
	3000	19

Table C.1: A list of a selection of the tested absorption scaling factors, including how fast the Marchenko iterative scheme reached the stability criterion, defined by a relative energy of $1e^{-3}$, after running the Marchenko scheme for 25 iterations, and for two depth: 1500m and 3000m.

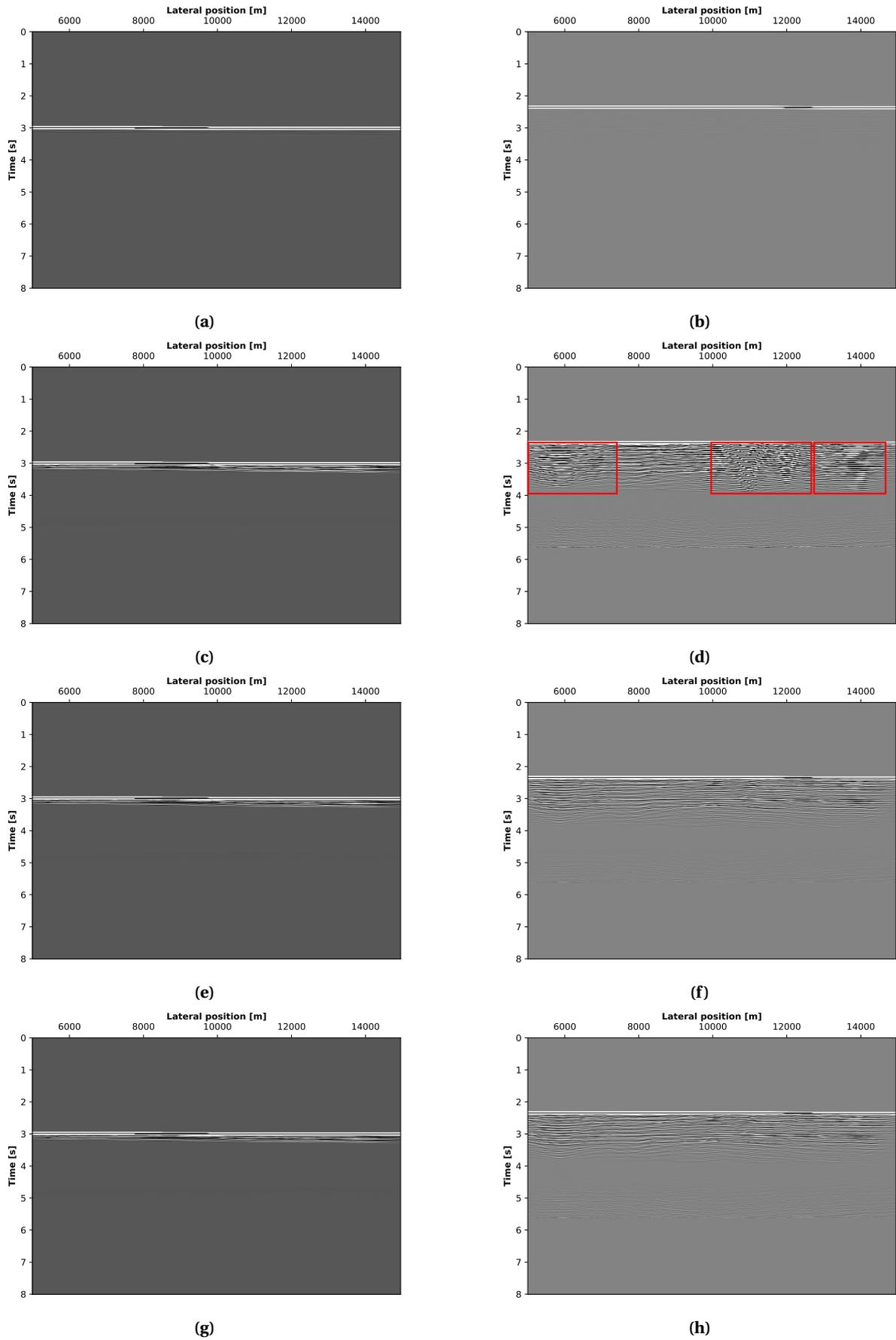


Figure C.1: The focusing functions associated with the scaling factors listed in [Table C.1](#). The left column contains the focusing functions for $z = 1500\text{m}$, whereas the right column contains the focusing functions for $z = 3000\text{m}$. Figures (a) and (b) were computed using a scaling factor of 12, figures (c) and (d) were computed using a scaling factor of $8.5e^t$, figures (e) and (f) were computed using a scaling factor of $9e^{0.9t}$, and figures (g) and (h) were computed using a scaling factor of $7.2e^t$. The red boxes in (d) indicate anomalous amplitudes.

D

PYTHON CODES

In this appendix, the main codes used to produce the results shown in this thesis are presented. The inputs to these codes, either finite difference or Marchenko related, are computed using the open-sourced package found in:

<https://github.com/JanThorbecke/OpenSource>.

In addition, the below codes require Python version 3.6 at least, and the python script supython.py, which can be found in:

<https://github.com/Jbrackenhoff/SUpython>.

D.1 Image Extraction for Back-Propagation by Finite Difference Extrapolation (Without Phase-Shift)

This code extracts subsurface images for the specified angle and depth ranges for plane-wave Green's functions back-propagated by finite difference extrapolation. It applies the imaging condition expressed in [Equation 3.20](#), and hence the obtained images are jittered.

```
1 #!/var/dim/home/almobarm/anaconda3/bin/python
2
3 # %% Importing Packages
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from supython import readsu, writesu
7
8 # %% Input parameters
9
10 # Imaging start depth
11 zmin = 1000
12 # Imaging end depth
13 zmax = 3500
14 # Depth increment
15 dz = 5
16 # Number of depth levels
17 nz = int(((zmax - zmin)/dz) + 1)
18
19 # Imaging start angle
20 angle_min = 25
21 # Imaging end angle
22 angle_max = 25
23 # Angle increment
24 dangle = 5
25 # Number of angles
26 nangle = int(((angle_max - angle_min)/dangle) + 1)
27
28 # Minimum source location (as seen in the reflection data headers)
```

```
29 src_min = 5000
30 # Maximum source location (as seen in the reflection data headers)
31 src_max = 14950
32 # sources spatial sampling
33 dxsrc = 25
34 # Number of sources
35 nsrc = int(((src_max - src_min) / dxsrc) + 1)
36
37 # Minimum Receiver location (as used for modelling the initial focusing
   functions)
38 rcv_min = 375
39 # Maximum Receiver location (as used for modelling the initial focusing
   functions)
40 rcv_max = 19575
41 # Receivers spatial sampling
42 dxrcv = 25
43 # Number of Receivers
44 nrcv = int(((rcv_max - rcv_min) / dxrcv) + 1)
45
46 # Image figure directory
47 image_fig_dir = '/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
   scaled_dipping_comparison_V2/'
48 # Image figure name
49 image_fig_name = 'RTMScaled'
50
51 # Image su file direcotry
52 image_su_dir = image_fig_dir
53 # Image su file name
54 image_su_name = image_fig_name
55
56 # %% Extracting the image
57
58 # Looping over the imaging anlges
59 for ia, a in enumerate(range(angle_min, angle_max+dangle, dangle)):
60     print(f"Extracting an image for angle={a}")
61
62     # Directory where the back-propagation results are stored
63     image_dir = f'/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
   imageRTMScaled_7.2_1_DippingLong/angle{a}/'
64     # Base name of the back-propagation results
65     image_base = 'image_z'
66     # Suffix of the back-propagation results (including the file extention)
67     image_suffix = '_rp.su'
68
69     # Directory where the G- functions (input to back-propagation) are stored
70     source_dir = f'/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
   marchenkoRTMScaled_7.2_1_DippingLong/Gmin/angle{a}/'
71     # Base name of the G- functions
72     source_base = "CutGmin_z"
73     # Suffix of the G- functions (including the file extention)
74     source_suffix = '.su'
75
76     # Calculating the plane-wave dip in time
77     # propagation velocity (assumed, and not related to the velocity model of
   interest)
78     velocity = 1500
79     # Ray parameter
80     p = np.sin(np.deg2rad(a))/velocity
81     # The x coordinates (as used for the initial focusing functions)
82     x_axis = np.arange(rcv_min, rcv_max+dxrcv, dxrcv)
83     # The plane-wave dip in time
84     pt = x_axis * p
85
86     # Flipping the plane-wave dip for negative angles
87     if a < 0:
```

```

88     pt = np.abs(pt[::-1])
89     pt_max = np.max(pt)
90 else:
91     pt = np.abs(pt)
92     pt_max = np.max(pt)
93
94 # Initiating imaging matrices
95 temp = np.zeros(nsrc)
96 image = np.zeros((nz, nsrc))
97
98 # Looping for each imaging depth
99 for iz, z in enumerate(range(zmin, zmax+dz, dz)):
100     file_name = f"{image_dir}{image_base}{z}{image_suffix}"
101
102     # Reading the back-propagation result
103     data, data_hdr = readsu(file_name)
104
105     # Extracting image file parameters
106     # Number of traces
107     ntr = int(data_hdr.ntr[0])
108     # Spatial sampling
109     dx = data_hdr.d2[0]
110     # Minimum x coordinate
111     x_min = data_hdr.f2[0]
112     # Maximum x coordinate
113     x_max = (ntr) * dx + x_min
114     # Temporal sampling
115     dt = data_hdr.dt[0] * 1e-6
116     # Number of time samples
117     nt = data_hdr.ns[0]
118
119     # Reading the back-propagated source function to calculate its length
120     # in time to find t=0
121     source_file_name = f"{source_dir}{source_base}{z}{source_suffix}"
122     source, source_hdr = readsu(source_file_name)
123     source_ns = source_hdr.ns[0]
124     source_dt = source_hdr.dt[0]*1e-6
125     source_t = (source_ns * source_dt)
126     source_tns = int(source_t / dt)
127
128     # Wavelet half wavelength in samples. Should be calculated from the
129     # initial focusing functions
130     wavelet_shift = 25
131
132     # Extracting the image from each source location
133     for ix in range(nsrc):
134         image[iz,ix] = data[source_tns + int(pt[int(src_min/dxrcv) + int(
135             dxsrc/dxrcv*ix)]/dt) - int((pt_max/dt)) - wavelet_shift, ix]
136
137     # Updating the image su file headers
138     data_hdr.ns = (data_hdr.ns * 0) + np.ones(np.shape(data_hdr.ns)) * (nz)
139     data_hdr.f1 = (data_hdr.f1 * 0) + np.ones(np.shape(data_hdr.f1)) * (zmin)
140     data_hdr.d1 = (data_hdr.d1 * 0) + np.ones(np.shape(data_hdr.d1)) * (dz)
141     # writing the Image su file into disk
142     writesu(f"{image_su_dir}{image_su_name}_angle{a}.su", image, data_hdr)
143
144     # Saving the image figure
145     plt.figure()
146     plt.imshow(image, cmap='Greys', extent=[x_min,x_max,zmax,zmin], aspect='auto')
147     plt.xlabel("x(m)", size='medium', fontweight="bold")
148     plt.ylabel("z(m)", size='medium', fontweight="bold")
149     plt.gca().xaxis.tick_top()
150     plt.gca().xaxis.set_label_position('top')

```

```

149 plt.tight_layout()
150 # plt.gca().invert_yaxis()
151 # plt.colorbar()
152 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.pdf", bbox_inches =
'tight')
153 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.png", bbox_inches =
'tight')
154 plt.close()
155
156 # plt.show()

```

D.2 Image Extraction for Back-Propagation by Finite Difference Extrapolation (With Phase-Shift)

This code extracts subsurface images for the specified angle and depth ranges for plane-wave Green's functions back-propagated by finite difference extrapolation. It applies the imaging condition expressed in [Equation 3.23](#), and hence the obtained images do not suffer from jitter.

```

1 #!/varldim/home/almobarm/anaconda3/bin/python
2 # %% Importing Packages
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from supython import readsu, writesu
6
7 # %% Input parameters
8
9 # Imaging start depth
10 zmin = 1000
11 # Imaging end depth
12 zmax = 3500
13 # Depth increment
14 dz = 5
15 # Number of depth levels
16 nz = int(((zmax - zmin)/dz) + 1)
17
18 # Imaging start angle
19 angle_min = 25
20 # angle_min = 5
21 # Imaging end angle
22 angle_max = 25
23 # angle_max = 10
24 # Angle increment
25 dangle = 5
26 # Number of angles
27 nangle = int(((angle_max - angle_min)/dangle) + 1)
28
29 # Minimum source location (as seen when surange is used)
30 src_min = 5000
31 # Maximum source location (as seen when surange is used)
32 src_max = 14950
33 # sources spatial sampling
34 dxsrc = 25
35 # Number of sources
36 nsrc = int(((src_max - src_min) / dxsrc) + 1)
37
38 # Minimum Receiver location (as used for modelling the initial focusing
  functions)
39 rcv_min = 375
40 # Maximum Receiver location (as used for modelling the initial focusing
  functions)
41 rcv_max = 19575
42 # Receivers spatial sampling

```

```
43 dxrcv = 25
44 # Number of Receivers
45 nrcv = int(((rcv_max - rcv_min) / dxrcv) + 1)
46
47 # Image figure directory
48 image_fig_dir = '/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
    scaled_dipping_comparison_V2/'
49 # Image figure name
50 image_fig_name = 'RTMScaled_timeshift'
51
52 # Image su file direcotry
53 image_su_dir = image_fig_dir
54 # Image su file name
55 image_su_name = image_fig_name
56
57 # %% Extracting the image
58
59 # Looping over the imaging anlgcs
60 for ia, a in enumerate(range(angle_min, angle_max+dangle, dangle)):
61     print(f"Extracting an image for angle={a}")
62
63     # Directory where the back-propagation results are stored
64     image_dir = f'/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
    imageRTMScaled_7.2_1_DippingLong/angle{a}/'
65     # Base name of the back-propagation results
66     image_base = 'image_z'
67     # Suffix of the back-propagation results (including the file extention)
68     image_suffix = '_rp.su'
69
70     # Directory where the G- functions (input to back-propagation) are stored
71     source_dir = f'/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
    marchenkoRTMScaled_7.2_1_DippingLong/Gmin/angle{a}/'
72     # Base name of the G- functions
73     source_base = "CutGmin_z"
74     # Suffix of the G- functions (including the file extention)
75     source_suffix = '.su'
76
77
78     # Calculating the plane-wave dip in time
79     # propagation velocity (assumed, and not related to the velocity model of
    interest)
80     velocity = 1500
81     # Ray parameter
82     p = np.sin(np.deg2rad(a))/velocity
83     # The x coordinates (as used for the initial focusing functions)
84     x_axis = np.arange(rcv_min, rcv_max+dxrcv, dxrcv)
85     # The plane-wave dip in time
86     pt = x_axis * p
87
88     # Flipping the plane-wave dip for negative angles
89     if a < 0:
90         pt = np.abs(pt[::-1])
91         # print(pt)
92         pt_max = np.max(pt)
93     else:
94         pt = np.abs(pt)
95         # print(pt)
96         pt_max = np.max(pt)
97
98     # Initiating imaging matrices
99     temp = np.zeros(nsrc)
100     image = np.zeros((nz, nsrc))
101
102     # Looping for each imaging depth
103     for iz, z in enumerate(range(zmin, zmax+dz, dz)):
```

```

104     file_name = f"{image_dir}{image_base}{z}{image_suffix}"
105
106     # Reading the result of back-propagation
107     data, data_hdr = readsu(file_name)
108
109     # Extracting image file parameters
110     # Number of traces
111     ntr = int(data_hdr.ntr[0])
112     # Spatial sampling
113     dx = data_hdr.d2[0]
114     # Minimum x coordinate
115     x_min = data_hdr.f2[0]
116     # Maximum x coordinate
117     x_max = (ntr) * dx + x_min
118     # Temporal sampling
119     dt = data_hdr.dt[0] * 1e-6
120     # Number of time samples
121     nt = data_hdr.ns[0]
122
123     # Calculating frequency parameters
124     # Frequency sampling
125     df = 1 / (nt * dt)
126     # Frequency axis
127     faxis = np.arange(0,(nt/2)+1) * df
128     # Number of frequencies
129     nf = len(faxis)
130
131     # Creating a matrix out of the frequency axis
132     fmatrix = np.transpose(np.array([faxis] * ntr))
133     # Creating a matrix out of the plane wave dip in time
134     ptmatrix = np.array([pt[int((src_min-rcv_min)/dxrcv):int(((src_max-rcv_min)+dxrcv)/dxrcv):int(dxsrc/dxrcv)]]*nf)
135
136     # Applying fft to back-propagation results
137     file_fft = np.fft.rfft(data, axis=0)
138
139     # Applying phase-shift (i.e. time shift in time domain)
140     file_rotated_fft = np.zeros(np.shape(file_fft), dtype=np.complex128)
141     file_rotated_fft = file_fft * np.exp(1j * 2*np.pi*fmatrix * ptmatrix)
142
143     # Applying inverse fft
144     file_rotated = np.fft.irfft(file_rotated_fft, axis=0)
145
146     # Reading the back-propagated source function to calculate its length
147     # in time to find t=0
148     source_file_name = f"{source_dir}{source_base}{z}{source_suffix}"
149     source, source_hdr = readsu(source_file_name)
150     source_ns = source_hdr.ns[0]
151     source_dt = source_hdr.dt[0]*1e-6
152     source_t = (source_ns * source_dt) - rec_delay
153     source_tns = int(source_t / dt)
154
155     # Wavelet half wavelength in samples. Should be calculated from the
156     # initial focusing functions
157     wavelet_shift = 25
158
159     # Extracting the image
160     image[iz,:] = file_rotated[source_tns - int((pt_max/dt)) -
161     wavelet_shift, :]
162
163     # Updating image su headers
164     data_hdr.ns = (data_hdr.ns * 0) + np.ones(np.shape(data_hdr.ns)) * (nz)
165     data_hdr.f1 = (data_hdr.f1 * 0) + np.ones(np.shape(data_hdr.f1)) * (zmin)
166     data_hdr.d1 = (data_hdr.d1 * 0) + np.ones(np.shape(data_hdr.d1)) * (dz)

```

```

165 # writing image su file into disk
166 writesu(f"{image_su_dir}{image_su_name}_angle{a}.su", image, data_hdr)
167
168 # Saving the image figure
169 plt.figure()
170 plt.imshow(image, cmap='Greys', extent=[x_min, x_max, zmax, zmin], aspect='auto')
171 plt.xlabel("x(m)", size='medium', fontweight="bold")
172 plt.ylabel("z(m)", size='medium', fontweight="bold")
173 plt.gca().xaxis.tick_top()
174 plt.gca().xaxis.set_label_position('top')
175 plt.tight_layout()
176 # plt.gca().invert_yaxis()
177 # plt.colorbar()
178 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.pdf", bbox_inches =
'tight')
179 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.png", bbox_inches =
'tight')
180 plt.close()
181
182 # plt.show()

```

D.3 Cross-Correlation with Point-Source Green's Functions

This code is used to cross-correlate the plane-wave Green's functions with the direct arrival of point-source Green's functions modelled via an Eikonal solver in a multi-dimensional manner. Refer to [Equation 3.18](#). The Eikonal solver parameters are chosen such that the point-source responses associated with a certain depth are stored in one file. Parallelization is implemented such that each CPU performs the cross-correlation for single depth.

```

1 #!/vardim/home/almobarm/anaconda3/bin/python3
2 # -- coding: utf-8 --
3
4 # %% Importing Packages
5 import numpy as np
6 from concurrent.futures import ProcessPoolExecutor
7 from concurrent.futures import wait
8 from supython import readsu, writesu
9 import time
10
11 # %% Defining input parameters
12
13 # Number of shots per raytime output (should be the same for each depth)
14 ray_nxsrc = 399
15
16 # %% Defining a function that would read the files and perform correlation and
    then write the data
17
18 def inv_extrap(ray_nxsrc, z):
19     # Directory where the G- functions are stored
20     Gmin_dir = '/vardim/home/almobarm/OpenSource/marchenko/saga_test3/
    marchenkoScaled_7.1_1_Horizontal/Gmin/'
21     # Base name of the G- functions
22     Gmin_base = 'Gmin_z'
23     # Suffix name of the G- functions (including the file extension)
24     Gmin_suffix = '.su'
25     # Complete path to the G- functions
26     Gmin_file = f'{Gmin_dir}{Gmin_base}{z}{Gmin_suffix}'
27
28     # Directory where the Eikonal solver results are stored
29     ray_dir = '/vardim/home/almobarm/OpenSource/marchenko/saga_test3/Ray/'
30     # Base name of the Eikonal solver results
31     ray_base = 'Ray_z'

```

```

32 # Suffix name of the Eikonal solver results
33 ray_suffix = '.su'
34 # Complete path to the Eikonal solver results
35 ray_file = f'{ray_dir}{ray_base}{z}{ray_suffix}'
36
37 # Reading G- file and extracting its headers
38 # Reading the G- file
39 Gmin, Gmin_hdr = readsu(Gmin_file)
40
41 # Extracting the G- headers
42 Gmin_nt = int(Gmin_hdr.ns[0])
43 Gmin_nx = int(Gmin_hdr.trwf[0])
44 dx = Gmin_hdr.d2[0]
45 Gmin_xmin = Gmin_hdr.f2[0]
46 Gmin_xmax = (Gmin_nx) * dx + Gmin_xmin
47 Gmin_dt = Gmin_hdr.dt[0] * 1e-6
48 Gmin_df = 1 / (Gmin_nt * Gmin_dt)
49 Gmin_faxis = np.arange(0, (Gmin_nt/2)+1) * Gmin_df
50 Gmin_nf = len(Gmin_faxis)
51 Gmin_dw = 2 * np.pi * Gmin_df
52 Gmin_waxis = np.array(Gmin_faxis * 2 * np.pi)
53 Gmin_nw = len(Gmin_waxis)
54
55 # Performing FFT on the G- file
56 Gmin_xf = np.fft.rfft(Gmin, axis=0)
57
58 # Reading raytime file and extracting its headers
59 # Reading the Eikonal solver result
60 ray, ray_hdr = readsu(ray_file)
61
62 # Extracting the headers from the Eikonal solver results
63 ray_nt = int(ray_hdr.ns[0])
64 ray_nx = int(ray_hdr.trwf[0])
65 dx = ray_hdr.d2[0]
66 ray_xmin = ray_hdr.f2[0]
67 ray_xmax = (ray_nx) * dx + ray_xmin
68 ray_dt = ray_hdr.dt[0] * 1e-6
69 ray_df = 1 / (ray_nt * ray_dt)
70 ray_faxis = np.arange(0, (ray_nt/2)+1) * ray_df
71 ray_nf = len(ray_faxis)
72 ray_dw = 2 * np.pi * ray_df
73 ray_waxis = np.array(ray_faxis * 2 * np.pi)
74 ray_nw = len(ray_waxis)
75
76 # reshaping the raytime file to a 3D matrix (time, number of receivers per
77 shot, number of shots)
78 ray_reshape = np.reshape(ray, (ray_nt, ray_nx, ray_nxsrc))
79
80 # Padding the raytime file so it matches the dimentions of the G- file
81 if np.shape(ray_reshape)[0] != np.shape(Gmin)[0]:
82     if np.shape(Gmin) > np.shape(ray_reshape):
83         temp = np.zeros((Gmin_nt, ray_nx, ray_nxsrc))
84         temp[0:ray_nt, :, :] = ray_reshape
85         ray_reshape = temp
86
87 # Performing FFT on the raytime file
88 ray_reshape_xf = np.fft.rfft(ray_reshape, axis=0)
89
90 # initiating the back-propagated G- file
91 Gmin_backprop_xf = np.zeros((Gmin_nf, Gmin_nx), dtype=np.complex128)
92 # Looping over each raytime shot (for a specific depth) and cross-correlate
93 it with the G- file
94 for i in range(ray_nxsrc):
95     Gmin_backprop_xf[:, i] = np.sum((np.conjugate(ray_reshape_xf[:, :, i]) *
96     Gmin_xf), axis=1)

```

```

94
95 # performing inverse FFT on the back-propagated G- file
96 Gmin_backprop_xt = np.fft.irfft(Gmin_backprop_xf, axis=0)
97
98 # Writing the back-propagated G- into disk
99 writesu(f"./angle0/back_extrapolation_{z}.su", Gmin_backprop_xt, Gmin_hdr)
100
101
102
103 # =====
104 # Including the main function is necessary for parallization
105 # =====
106 if __name__ == '__main__':
107     start = time.time()
108     # Making an object of the parallization engine. "max_workers" is the number
109     # of CPUs to be utilized
110     with ProcessPoolExecutor(max_workers=25) as executor:
111         # Executing the back-propagation function for each depth "z"
112         futures = [executor.submit(inv_extrap, ray_nxsrc, z) for z in range
113 (1000, 3500+5, 5)]
114         wait(futures, return_when="ALL_COMPLETED")
115     print(f"time taken is {time.time()-start}")

```

D.4 Image Extraction for Back-Propagation by Cross-Correlation (With Phase-Shift)

This code extracts subsurface images for the specified angle and depth ranges for plane-wave Green's functions back-propagated by cross correlation. It applies the phase-shifted imaging condition expressed in [Equation 3.22](#), and hence the obtained images do not suffer from jitter.

```

1 #!/var/dim/home/almobarm/anaconda3/bin/python
2 # %% Importing Packages
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from supython import readsu, writesu
6 from tqdm import tqdm, trange
7
8 # %% Input parameters
9
10 # Imaging start depth
11 zmin = 1000
12 # Imaging end depth
13 zmax = 3500
14 # Depth increment
15 dz = 5
16 # Number of depth levels
17 nz = int(((zmax - zmin)/dz) + 1)
18
19 # Imaging start angle
20 # angle_min = 25
21 angle_min = 0
22 # Imaging end angle
23 # angle_max = 25
24 angle_max = 10
25 # Angle increment
26 dangle = 5
27 # Number of angles
28 nangle = int(((angle_max - angle_min)/dangle) + 1)
29
30 # Minimum source location (as seen when surange is used)
31 src_min = 5000
32 # Maximum source location (as seen when surange is used)

```

```

33 src_max = 14950
34 # sources spatial sampling
35 dxsrc = 25
36 # Number of sources
37 nsrc = int(((src_max - src_min) / dxsrc) + 1)
38
39 # Minimum Receiver location (as seen when surange is used)
40 rcv_min = 375
41 # rcv_min = src_min
42 # Maximum Receiver location (as seen when surange is used)
43 rcv_max = 19575
44 # rcv_max = src_max
45 # Receivers spatial sampling
46 dxrcv = 25
47 # Number of Receivers
48 nrcv = int(((rcv_max - rcv_min) / dxrcv) + 1)
49
50 # Back-propagation source function time delay (see the rec_delay parameter in
    the back-propagation jobs)
51 rec_delay=0.0
52
53 # Image figure directory
54 image_fig_dir = f'//vardim/home/almobarm/OpenSource/marchenko/Back-Extrapolation
    /'
55 # Image figure name
56 image_fig_name = 'Test'
57
58 # Image su file direcotry
59 image_su_dir = image_fig_dir
60 # Image su file name
61 image_su_name = image_fig_name
62
63 # %% Extracting the image
64
65 # Looping over the imaging anlgcs
66 for ia, a in enumerate(trange(angle_min, angle_max+dangle, dangle)):
67     print(f"Creating an image for angle={a}")
68
69     # Directory where the back-propagation results are stored
70     image_dir = f'//vardim/home/almobarm/OpenSource/marchenko/Back-Extrapolation
        /angle{a}/'
71     # Base name of the back-propagation results
72     image_base = f'back_extrapolation_'
73     # Suffix of the back-propagation results (including the file extention)
74     image_suffix = '.su'
75
76     # Calculating the plane-wave dip in time
77     # propagation velocity (assumed, and not related to the velocity model of
        interest)
78     velocity = 1500
79     # Ray parameter
80     p = np.sin(np.deg2rad(a))/velocity
81     # The x coordinates (as used for the initial focusing functions)
82     x_axis = np.arange((src_min-rcv_min), (src_max-rcv_min)+dxsrc, dxsrc)
83     # The plane-wave dip in time
84     pt = x_axis * p
85     # Flipping the plane-wave dip for positive angles
86     if a > 0:
87         pt = pt[::-1]
88     else:
89         pt = np.abs(pt)
90
91     pt_max = np.max(pt)
92
93     # Initiating imaging matrices

```

```

94     temp = np.zeros(nsrc)
95     image = np.zeros((nz, nsrc))
96
97     # Looping for each imaging depth
98     for iz, z in enumerate(trange(zmin, zmax+dz, dz)):
99         file_name = f"{image_dir}{image_base}{z}{image_suffix}"
100
101         # Reading the result of back-propagation
102         data, data_hdr = readsu(file_name)
103
104         # Extracting image file parameters
105         # Number of traces
106         ntr = int(data_hdr.trwf[0])
107         # Spatial sampling
108         dx = data_hdr.d2[0]
109         # Minimum x coordinate
110         x_min = data_hdr.f2[0]
111         # Maximum x coordinate
112         x_max = (ntr) * dx + x_min
113         # Temporal sampling
114         dt = data_hdr.dt[0] * 1e-6
115         print(f"temporal sampling is {dt} s")
116         # Number of time samples
117         nt = data_hdr.ns[0]
118         # Frequency sampling
119         df = 1 / (nt * dt)
120         # Frequency axis
121         faxis = np.arange(0, (nt/2)+1) * df
122         # Number of frequencies
123         nf = len(faxis)
124
125         # Creating a matrix out of the frequency axis
126         fmatrix = np.transpose(np.array([faxis] * ntr))
127         # Creating a matrix out of the plane wave dip in time
128         ptmatrix = np.array([pt]*nf)
129
130         # Applying fft to back-propagation results
131         file_fft = np.fft.rfft(data, axis=0)
132
133         # Applying phase-shift (i.e. time shift in time domain)
134         file_rotated_fft = np.zeros(np.shape(file_fft), dtype=np.complex128)
135         file_rotated_fft = file_fft * np.exp(1j * 2*np.pi*fmatrix * ptmatrix)
136
137         # Applying inverse fft
138         file_rotated = np.fft.irfft(file_rotated_fft, axis=0)
139
140         # Extracting the image
141         image[iz,:] = file_rotated[0 , :]
142
143
144         # Updating su headers
145         data_hdr.ns = (data_hdr.ns * 0) + np.ones(np.shape(data_hdr.ns)) * (nz)
146         data_hdr.f1 = (data_hdr.f1 * 0) + np.ones(np.shape(data_hdr.f1)) * (zmin)
147         data_hdr.d1 = (data_hdr.d1 * 0) + np.ones(np.shape(data_hdr.d1)) * (dz)
148         # writing su file into disk
149         writesu(f"{image_su_dir}{image_su_name}_angle{a}.su", image, data_hdr)
150
151         # Saving the image figure
152         plt.figure()
153         plt.imshow(image, cmap='Greys', extent=[x_min, x_max, zmax, zmin], aspect='auto')
154         plt.xlabel("x(m)", size='medium', fontweight="bold")
155         plt.ylabel("z(m)", size='medium', fontweight="bold")
156         plt.gca().xaxis.tick_top()
157         plt.gca().xaxis.set_label_position('top')

```

```
158 plt.tight_layout()
159 # plt.gca().invert_yaxis()
160 # plt.colorbar()
161 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.pdf", bbox_inches =
'tight')
162 plt.savefig(f"{image_fig_dir}{image_fig_name}_angle{a}.png", bbox_inches =
'tight')
163 plt.show()
```