All-At-Once and Reduced Solvers for Visco-Acoustic Full Waveform Inversion

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der KIT-Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation

von

Christian Clemens Rheinbay

Tag der mündlichen Prüfung: 08. November 2023

1. Referent: Prof. Dr. Andreas Rieder
2. Referent: Prof. Dr. Christian Wieners
3. Referent: Prof. Dr. Roland Griesmaier

DANKSAGUNG

Sieben Jahre sind eine lange Zeit. In dieser Zeit durfte ich viel lernen und wurde von vielen Menschen untersützt. Einigen davon möchte ich an dieser Stelle danken.

Prof. Dr. Andreas Rieder danke ich für das gesetzte Vertrauen und die bis zum Schluß (sic!) notwendige Geduld. Die Zusammenarbeit hat mir Spaß gemacht und die Betreuung der Lehrveranstaltungen war eine schöne, wertvolle und oft auch lustige Erfahrung.

Bei Prof. Dr. Christian Wieners möchte ich mich für die Hilfe beim Debuggen von `M++` bedanken, insbesondere zu Beginn der Promotion, sowie für die Diskussionen zum Modell und der räumlichen Diskretisierung. Natürlich danke ich auch für die Übernahme der Zweitkorrektur und die Einladungen zu Grillfesten und Geburtstagen bei ihm zu Hause.

Prof. Dr. Roland Griesmaier danke ich für die Übernahme der Drittkorrektur und die aufmerksame Lektüre meiner Arbeit. Die vielen hilfreichen Hinweise haben die Endfassung meiner Arbeit lesbarer gemacht.

Bei Daniel Weiß bedanke ich mich für viele interessante Diskussionen und Denkanstöße. Besonders seine Motivation, die Lehre ständig zu verbessern, hat mich beeindruckt. Auch die Einladungen zum Grillen und das generelle Bemühen, das Zusammenleben in der Arbeitsgruppe positiv zu gestalten, verdienen Dank.

Insbesondere für die Hilfe zu Beginn meiner Promotionszeit bedanke ich mich bei Johannes Ernesti. Unzählige Diskussionen und Erklärungen rund um `C++`, `M++` und Mathematik haben mir den Einstieg in die Promotionszeit sehr erleichtert. Durch seine Hilfe konnte ich die ersten Versionen der FWI-Software erstellen und leichter Vorlesungen betreuen, die ich selbst nie gehört hatte.

Benny danke ich für viele Kaffeepausen (handgemahlen) mit Gesprächen über Musik, Dissertationsfrust, Wortwitze und das Leben im Allgemeinen und Speziellen. Gemeinsame Konzert- und Boulderhallenbesuche haben mein Leben auch außerhalb der Arbeit bereichert.

Ein weiterer besonderer Dank gilt Lukas für sehr viele (Vanilla) Cokes, viele Mittagessen im Panda, häufige FWI-Diskussionen und seltene gemeinsame Gymbesuche. Für ein lächerliches Entgelt von wenigen Schokobons hat er meine ersten, teilweise noch sehr rauhen Versionen gelesen und mir Korrekturen dazu gegeben. Ohne ihn hätte ich vielleicht die 10 Jahre voll gemacht.

Bei Christine bedanke ich mich für viele interessante Gespräche und Pflanzenpflege.

Daniele danke ich für seine Hilfsbereitschaft, unzählige Male hat er mir beim Kompilieren, Setzen von Umgebungsvariablen und Debuggen geholfen. Oft habe ich die Möglichkeit genutzt, ins nächste Büro zu schlappen und irgendwelche fachlichen Fragen zu disktuieren. Vielen Dank auch für das Korrekturlesen und Finden nicht geschlossener Klammern.

Bei Kevin bedanke ich mich fürs Korrekturlesen und die tägliche (Di-Do) Organisation des Mittagessens.

Auch dem restlichen (neben Daniele) `M++`-Team (Johnathan F., Johnathan W., Niklas) möchte ich danken. Durch viel Arbeit ist es in den letzten Jahren gelungen, die Bibliothek auf ein solide(ere)s Fundament zu stellen und somit die Arbeit mit der Bibliothek zu erleichtern.

Ich möchte mich auch bei allen anderen Kolleginnen und Kollegen bedanken. Eigentlich hätte jede und jeder einen eigenen Satz verdient, der Kürze halber beschränke ich mich aber auf eine einfache Nennung. Sie alle haben zu einer angenehmen, lustigen und geselligen Arbeitsatmosphäre beigetragen:
Christian K., Daniel Z., David, Johanna, Julian B., Julian K., Laura L., Laura S., Lydia, Mathias, Michael, Nathalie, Marcel, Patrick, Philip, Ramin, Simon, Sonja, Tobias.

Meiner wunderbaren Frau Juli danke ich für ihren dauerhaften Rückhalt, ihre Unterstützung (inklusive Nachtschicht) und ihre Liebe in den letzten Jahren. Gerade im letzten halben Jahr vor der Abgabe hat sie mich immer wieder aufgemuntert und wieder aufgepeppelt, wenn ich vor lauter Bäumen den Wald nicht mehr gesehen habe.

Ich danke meinem Vater für die jahrelange Unterstützung und die mathematische Früherziehung (mit Gummibärchen und Smarties). Obwohl ich früher nie Mathe studieren wollte, bin ich trotzdem letzten Endes in seine Fußstapfen getreten. Ich bin froh, dass ich „mein ε“ finden konnte und den Weg zu Ende gehen konnte.

Ich danke meiner Mutter dafür, dass sie mich in allem unterstützt hat. Egal was ich gemacht habe. Das gilt speziell für mein Studium. Auch wenn sie nicht viel von Mathematik versteht, weder damals und erst recht nicht heute.

CONTENTS

List of symbols iii

List of Algorithms v

1 Motivation and Introduction 1

2 Modeling and Existence theory 5

2.1 Forward modeling 5

2.1.1 Derivation of the model 7

2.1.2 Viscosity 7

2.2 Existence theory 9

2.3 Adjoint equation 13

2.4 Measurement operator 13

3 Discretization 15

3.1 Space Discretization 15

3.2 Time Discretization 17

3.3 The finite element library `M++` 18

3.4 Numerical Convergence 19

3.4.1 Source implementation 19

3.5 Absorbing boundary conditions 20

4 Inversion with the reduced formulation 23

4.1 Reduced formulation 23

4.1.1 Theory of the reduced operator 23

4.2 Discretization aspects for the Fréchet derivative and its adjoint 31

4.3 The `Red-CG-REGINN` algorithm 33

4.3.1 Connection to geophysics 36

4.4 `Red-CG-REGINN` Mono-parameter reconstruction 37

4.4.1 Transmission problem 37

4.4.2 Marmousi 49

4.4.3 Reconstructions with noise 55

4.4.4 Artifacts 59

4.5 Multi-parameter reconstruction in FWI 65

4.5.1 Multi Steepest Descent (mSD) - A subspace gradient descent ansatz 66
4.5.2 Application of `mSD-REGINN` in the reduced case 69
4.5.3 Subspace Preconditioned `Red-CG-REGINN` (`Red-SPCG-REGINN`) 73

5 Inversion with all-at-once formulation 75

5.1 All-at-once formulation 75
5.1.1 All-at-once operator 75
5.1.2 Theory of all-at-once 77
5.2 Discretization of the operators 80
5.3 Algorithms for the all-at-once formulation 83
5.3.1 Sequential inversion 83
5.3.2 `AAO-CG-REGINN` 86
5.3.3 `AAO-PmSD-REGINN` 87
5.4 Camembert Model 92
5.5 Marmousi 101
5.5.1 Application of `AAO-PmSD-REGINN` 101
5.5.2 Continuation 102
5.5.3 Marmousi with noise 103
5.5.4 On the choice of the penalty factor λ in (AAO) 106

6 Conclusion 109

6.1 Contribution 109
6.2 Outlook 110

A Green’s Functions 113

B Calculation Times 115

Bibliography 117

LIST OF SYMBOLS

Symbol	**Description**
$\overline{D}$	Closure of a set D
$\mathcal{L}(U,V)$	Space of linear bounded Operators $U \longrightarrow Y$
χ_D	Indicator function on a set D
$\mathrm{int}(D)$	Interior of a set D
$\mathcal{D}(A)$	Domain of the operator A
$L^p(\Omega, H)$	Standard L^p-space with respect to the Lebesgue measure on a domain Ω with values in a Hilbert Space H.
$W^{k,p}([0,T], H)$	Sobolev space of k times weakly differentiable functions on the interval $[0,T]$ with derivatives in $L^p([0,T], H)$ in a Hilbert space H
$C^k([0,T], H)$	Space of k times differentiable functions on the interval $[0,T]$ with values in a Hilbert space H
L	Number of damping mechanisms
$\boldsymbol{X}$	$L^2(\Omega, \mathbb{R}^{2+(1+L)})$
$\boldsymbol{\mathfrak{L}}$	Differential operator for the wave equation
$\boldsymbol{Y}$	$L^2([0,T], \boldsymbol{X})$, space of the wave fields
$\boldsymbol{P}$	$L^\infty(\Omega)^3$
$\boldsymbol{P}_{\mathrm{adm}}$	Space of admissible parameters, subset of $\boldsymbol{P}$
Ψ	Measurement operator
R	Number of receivers
J	Number of measurements
Ξ	Number of shots
$\tilde{\boldsymbol{S}}$	Discrete measurements gathered in assembled as $\mathbb{R}^{R\times J}$
$\boldsymbol{S}$	Linear spline representation of $\tilde{\boldsymbol{S}}$ with equivalent scalar product
$\boldsymbol{X}_h$	Discontinous Galerkin discretization space of $\boldsymbol{X}$
$\boldsymbol{Y}_h$	Discontinous Galerkin discretization space of $\boldsymbol{Y}$
$\mathcal{F}$	Parameter-to-solution map, used in reduced method
Φ	Parameter-to-seismogram map, used in reduced method
$\boldsymbol{M}$	Operator in containing the material parameters
$\boldsymbol{\mathfrak{L}}'_{\mu,\boldsymbol{u}}$	Linearized evaluation of $\boldsymbol{\mathfrak{L}}$ with respect to the parameter

Symbol	Description
$\boldsymbol{\mathfrak{L}}'^{*}_{\mu,\boldsymbol{u}}$	Adjoint of $\boldsymbol{\mathfrak{L}}'_{\mu,\boldsymbol{u}}$ (yields a parameter vector)
$\boldsymbol{P}_h$	Cell-wise constant discrete parameter space
r	Ricker wavelet
$\tilde{r}$	integrated Ricker wavelet
$\boldsymbol{F}$	Frequency stages in inversion
SNR	Signal-to-noise ratio
$\boldsymbol{X}_{-1}$	Extended version of $\boldsymbol{X}$
$\boldsymbol{Y}_{-1}$	Extended version of $\boldsymbol{Y}$
$\tilde{\boldsymbol{\mathfrak{L}}}$	Integrated version of $\boldsymbol{\mathfrak{L}}$
$\tilde{\Phi}$	All-at-once operator, used in the all-at-once method
$\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}$	Linearized evaluation of $\tilde{\boldsymbol{\mathfrak{L}}}$ with respect to the parameter
$\tilde{\boldsymbol{\mathfrak{L}}}'^{*}_{\mu,\boldsymbol{u}}$	Adjoint of $\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}$ (yields a parameter vector)
N_{inv}	Number of parameters in the inversion

LIST OF ALGORITHMS

Red-CG-REGINN	Reduced Conjugate Gradient REGINN
Red-mSD-REGINN	Reduced Multi Steepest Descent REGINN
Sequential Inversion	Sequential Inversion algorithm
AAO-CG-REGINN	All-At-Once Conjugate Gradient REGINN
AAO-PmSD-REGINN	All-At-Once Preconditioned Multi Steepest Descent REGINN

Reduced methods

Abbreviation	Used in Context	Geometries
Red-CG-REGINN	Mono- and multi-parameter with and without noise/inverse crime	T,M,C
Red-mSD-REGINN	multi-parameter inversions with inverse crime	T
Red-SPCG-REGINN	multi-parameter inversions with inverse crime	T

All-At-Once methods

Abbreviation	Used in Context	Geometries
–	Mono-parameter inversions with inverse crime	T
AAO-CG-REGINN	Mono-parameter inversions with inverse crime	T
AAO-PmSD-REGINN	Mono-parameter inversions with and without noise/inverse crime	T,M,C

Table 3: Name and in which contexts the algorithms are used. Abbreviations for the geometries are ’T’(Transmission Fig. 4.1),’M’(Marmousi Fig. 4.13) ’C’(Camembert Fig. 5.3)

CHAPTER ONE

MOTIVATION AND INTRODUCTION

Full waveform inversion (FWI) seeks to determine material properties such as (pressure/shear) wave velocity, mass density, and attenuation below the earth's surface from measurements of seismic waves on the surface. The aim is to identify a material distribution whose simulated solution of the wave equation can account for the full information contained in seismic recordings. The comparison of the full data content is what distinguishes FWI from techniques that invert for either phase (e.g travel time inversion) or amplitude (e.g Amplitude Versus Offset) information only. Historically, methods in seismic exploration rely on assumptions from geometric optics, that is, that waves travel on rays from one point to another. While this assumption made calculations numerically feasible in the past, it also results in a lower resolution in the reconstructions overall. By dropping this assumption and making full calculations of wave fields FWI does not have the constraint of geometric optics and allows for a increased resolution in exchange for more computational effort. With FWI we are able to gain information about the structure of earth. On a large scale we can gain insight about plate tectonics, and on a small scale we can find reservoirs of natural resources like oil or gas. An early pioneer of this field was Albert Tarantola. He formulated the reconstruction as a least-squares optimization problem to invert for the pressure wave velocity and suggested an iterative process to find a suitable model [Tar84]. Let μ be the sought-for material and $\boldsymbol{u}$ a space-time wave field. Then in geophysics this formulation is typically stated as a regularized constrained optimization problem of the form

$$\text{find } \mu \text{ such that } J(\mu) = C(\boldsymbol{u}) + \beta\mathcal{R}(\mu) \longrightarrow \min \ \text{ subject to } M(\mu)\boldsymbol{u} = 0,\ \beta \in \mathbb{R}_+\,,$$

with C being a quadratic functional and M a linear constraint on $\boldsymbol{u}$. The functional C contains the mismatch of the recorded and simulated data, and the constraint M ensures that the simulated data is a solution of the differential equation modeling the wave propagation. The term $\mathcal{R}(\mu)$ is called *regularization* and does not come directly from the physical problem but rather is a tool to stabilize the minimization. The influence of the regularization term is steered by β. While C being a suitable L^2 distance is the classic formulation by Tarantola and the one we will use in this thesis, many other choices are considered in the literature [EGvL+18, BOV09]. The problems in FWI are large scale in nature and as computational power continues to increase, more possibilities arise to exploit this process to its full potential. Because of this, FWI is an active field of research to this day. Overviews can be found in [VO09, OGP+13] or more recently in [VAB+17]. Textbooks by Tarantola [Tar05] and Fichtner [Fic11] offer introductions to a wide scope of topics in the field. Fichtner's book covers more seismological aspects whereas Tarantola's book is more focused on the inverse problem aspect from a probabilistic perspective. The problem of FWI has also been discussed in mathematical publications with regard to its ill-posedness in the sense of inverse problems (in a rigorous mathematical sense) and Fréchet-differentiability [KR16, KR19]. In the past decade, a new type of formulation for the underlying problem in FWI often called *Wavefield Reconstruction Inversion* (WRI) has become more popular. There, numerical solutions are allowed to violate the partial differential equations that model the physics ($M(\mu) \neq 0$) and the discrepancy is included as an additional penalty term in the discrepancy functional [vLH13, vLH15]. With the notation above, we then can cast it as a relaxed version of the optimization above as

$$\text{find } (\boldsymbol{u}, \mu) \text{ such that } \tilde{J}(\boldsymbol{u}, \mu) = C(\boldsymbol{u}) + \lambda^2 \|M(\mu)\boldsymbol{u}\|_2 + \beta \mathcal{R}(\boldsymbol{u}, \mu) \longrightarrow \min\,,\ \lambda \in \mathbb{R},$$

with λ determining how much emphasis is put on the constraint. Although it looks similar, we do not consider the term $\|M(\mu)\boldsymbol{u}\|_2$ as a regularization of the optimization, but rather a physics motivated reformulation of the problem. The WRI formulation expands the search space, and therefore can help to find better approximations of the material parameters in the end. Moreover, it can be shown that the associated forward operator is less nonlinear [Rie21, vLH13]. For this reason, this formulation is expected to behave better in some ways for example to have an increased radius of convergence, since the methods applied to FWI are local and known to be heavily dependent on a good initial guess. The first WRI methods were done in frequency domain due to its easier implementation. But in the last years algorithms related to WRI are successfully applied also in time domain [GAO21] and showed promising results. The underlying formulation of WRI can be interpreted as what is called *all-at-once* approach in the mathematical community [HA01], since we invert for both, the material and $\boldsymbol{u}$ at once. Some mathematical results for abstract all-at-once formulations regarding regularization properties of iterative methods, like Landweber or Tikhonov, have been shown in [Kal16]. The problem of the all-at-once formulation (in a setting slightly different from how physicists usually look at it) was also shown to be ill-posed and Fréchet-derivatives of the associated forward operator were calculated in [Rie21].

In this thesis we expand further on the viability of Inexact Newton methods used in the mathematical nonlinear inverse problem literature [Rie05] applied to FWI in the sense of Tarantola, which from now on we will call *reduced* formulation. Moreover, we develop and apply inexact Newton methods for the all-at-once formulation in time domain and explore its potential for some typical synthetic FWI experiments.

This thesis is structured as follows: In Chapter 2 a derivation of the physical model for wave propagation and a short introduction to the governing *partial differential equation* (PDE) is given. Also, existence and uniqueness of solutions of the PDE are discussed. In Chapter 3 we present some discretization aspects in time and space. In Chapter 4 we present the problem of FWI in a operator-equation based setting that is typical for mathematical inverse problems. Further, we derive the formulas we need for the inexact Newton method. Moreover, we apply problem-adapted inexact Newton methods to the reduced formulation and present inversion results. In Chapter 5 we present theory for the all-at-once approach in an operator-based formulation. We show some ansatzes of algorithms and finally apply a problem-adapted inexact Newton method to the all-at-once formulation and give results. We summarize the results and give an outlook on possible future research in Chapter 6.

Note that we understand this thesis as a thesis in the context of mathematical inverse problems and therefore use the operator-based formulations in Chapter 4 and Chapter 5. However, we do also refer to *the underlying problems as (regularized) optimization problems for two reasons: First, every geophysical publication uses optimization formulations. Second, the behavior of optimization problems can often be somewhat easier to conceptualize compared to ill-posed equations.*

CHAPTER

TWO

MODELING AND EXISTENCE THEORY

2.1 Forward modeling

The starting point of any physics-related inverse problem is choosing a model describing the physics – in our case wave propagation. Wave propagation in solids is governed by the material law the wave permeates, that is, the relation of *stress* $\boldsymbol{\sigma}$ (force per area) and *strain* ε (change of deformation). These quantities are usually expressed as second order tensors with 3 and 6 components for 2 and 3 dimensions, respectively. In seismics it is typically assumed that the waves do not permanently deform the materials they permeate (often called *elasticity*). Another assumption is *linearity* of the material law, which is justified by the fact that the amplitudes of the waves (how much a particle moves from its position of equilibrium), with respect to the wavelength are small. *Attenuation* makes the response of the material to strain time-dependent and a convolution in the material law necessary. The most general form of a material law with the properties above is

$$\boldsymbol{\sigma} = \partial_t \Psi * \varepsilon \,, \tag{2.1}$$

where Ψ is the *relaxation function*, which vanishes for $t < 0$. The relationship via convolution in time means the reaction of the material is not completely spontaneous, but the material's stress response also depends on the time history of the strain and Ψ. The material law on its own does not give rise to an equation of motion, which is needed to model waves. Therefore, we complement (2.1) with the balance of momentum (or Newton's second law)

$$\varrho \partial_t \boldsymbol{v} = \operatorname{div} \boldsymbol{\sigma} \,, \tag{2.2}$$

with the velocity $\boldsymbol{v} = \partial_t \boldsymbol{u}$ and $\boldsymbol{u}$ being the displacement vector, that is, the difference between the position of a particle at equilibrium (when no force is applied) and its current position. The divergence in (2.2) is applied row wise. If we assume the displacements to be very small compared to the whole body (infinitesimal strain assumption), we get

$$\varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top) \,.$$

In this thesis we only consider longitudinal waves – called pressure waves – which makes $\boldsymbol{\sigma}$ diagonal, because the diagonals contain the stresses in normal directions. Further, we only consider isotropic behavior, that is, the material behaves the same, regardless of incidence of the incoming waves making

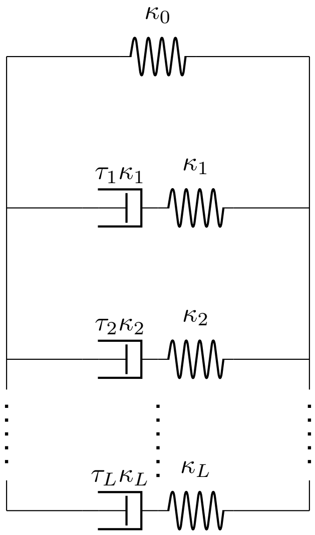


Figure 2.1: Dash pot and spring model for the GSLS

the material law symmetric. With these assumptions, the material law in (2.1) reduces to a scalar valued dependency. For $\kappa := \partial_t \Psi$ this yields

$$\boldsymbol{\sigma} = \kappa * \varepsilon(\boldsymbol{u}) \Rightarrow \partial_t \boldsymbol{\sigma} = \kappa * \varepsilon(\partial_t \boldsymbol{u}) = \kappa * \varepsilon(\boldsymbol{v})\,.$$

The model is further reduced in components, since for longitudinal waves it is sufficient to only consider the hydrostatic pressure p. We assume $\boldsymbol{\sigma}_{ii} = p/\dim$ for $i = 1, 2$ and $\dim = 2$ in the two-dimensional case, or $i = 1, 2, 3$ and $\dim = 3$ for the three-dimensional case, then

$$p := \operatorname{trace}(\boldsymbol{\sigma}) = \kappa * \operatorname{trace}(\varepsilon(\boldsymbol{v})) \Leftrightarrow p = \kappa * \operatorname{div}(\boldsymbol{v})\,.$$

Combining this with (2.2) yields the linear system

$$\begin{aligned} \varrho \partial_t \boldsymbol{v} &= \nabla p\,, \\ p &= \kappa * \operatorname{div}(\boldsymbol{v})\,. \end{aligned} \tag{2.3}$$

In this formulation Ψ is typically unknown. The time-dependent behavior of materials can be quantified by applying sudden stress to it and measure the evolution of the strain in the material and vice versa. In general, the relaxation function is not known. Therefore, modeling attenuation means modeling Ψ. Typically, the function Ψ is modeled as a combination of mechanical elements like springs and dash pots. This ansatz yields an analytic representation of Ψ, which eliminates the need of convolutions at the cost of additional equations (with additional variables that are called memory variables) in the linear system. In this thesis we choose the *Generalized Standard Linear Solid* (GSLS) to model attenuation. How springs and dash pots are arranged in GSLS is depicted in Fig. 2.1. A detailed account of all aspects discussed above (anisotropy, attenuation) as well as other ways to combine the mechanical elements can be found in Carcione's book [Car14].

2.1.1 Derivation of the model

Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain. Then, the visco-acoustic wave equation using the GSLS to model attenuation can be written as the following first order system

$$\begin{aligned}
\varrho(\boldsymbol{x})\partial_t \boldsymbol{v}(t,\boldsymbol{x}) &= \nabla \sum_{l=0}^{L} p_l(t,\boldsymbol{x}) + \boldsymbol{f}_{\boldsymbol{v}}(t,\boldsymbol{x}) && \text{in } [0,\infty)\times\Omega\,,\\
\frac{1}{\kappa_0(\boldsymbol{x})}\partial_t p_0(t,\boldsymbol{x}) &= \operatorname{div} \boldsymbol{v}(t,\boldsymbol{x}) + f_{p_0}(t,\boldsymbol{x}) && \text{in } [0,\infty)\times\Omega\,,\\
\frac{1}{\kappa_l(\boldsymbol{x})}\partial_t p_l(t,\boldsymbol{x}) &= \operatorname{div} \boldsymbol{v}(t,\boldsymbol{x}) - \frac{1}{\tau_l\kappa_l(\boldsymbol{x})} p_l(t,\boldsymbol{x}) + f_{p_l}(t,\boldsymbol{x}) \text{ for } l=1,\dots,L && \text{in } [0,\infty)\times\Omega\,,
\end{aligned} \tag{2.4}$$

where $\boldsymbol{v}\colon [0,\infty)\times\Omega \longrightarrow \mathbb{R}^2$ is the velocity field and $p_0,\dots,p_L\colon [0,\infty)\times\Omega \longrightarrow \mathbb{R}$ are the scalar pressure components and $\boldsymbol{f}_{\boldsymbol{v}}\colon [0,\infty)\times\Omega \longrightarrow \mathbb{R}^2$ and $f_{p_l}\colon [0,\infty)\times\Omega \longrightarrow \mathbb{R}$ are forcing terms for $l=0,\dots,L$. Further, $\varrho, \kappa_0,\dots,\kappa_L$ are the density and the bulk moduli and $\tau_1,\dots,\tau_L$ are the stress relaxation times of the dash pots. Writing these equations in that way was introduced in [Zel19]. If we choose $L=0$, the system reduces to the linear acoustic wave equation without attenuation.

2.1.2 Viscosity

We briefly discuss which effect attenuation has on the physics and modeling of the waves. Attenuation effects are linked to loss (or rather conversion) of energy of the wave due to friction. A typical physical quantity to characterize this behavior is called the quality factor Q ($Q^{-1} = 1/Q$ is often called *dissipation factor*). It can be defined by

$$Q(\omega) = \frac{2\pi\, W(\omega)}{\Delta W(\omega)},$$

where ΔW is the energy loss of the wave with frequency ω per period and W is the energy in that period. From this formula we can see that higher values of Q signify less energy loss and vice versa. Although the definition of Q allows frequency dependence, in seismics it is usually assumed – and this assumption fits to seismic measurements – that Q is constant over all relevant frequencies [LAK76]. Hence, over the same traveling distance the content of higher frequencies is damped stronger than lower frequencies. This is due to shorter wavelengths which go through more periods to travel the same distance. This effect can be observed in real data, meaning that the frequency content of measurements with a large distance to the source contain mainly low frequencies. To make this more tangible, assume $Q \gg 1$. Then, the amplitude decay of a monochromatic plane wave of frequency ω with velocity c as a function of propagation distance is described by $a(x) = \mathrm{e}^{-\omega x/(2cQ)}$ [Ige16]. The decay of the amplitude is faster for higher values of ω. Decreasing Q leads to a faster decay as well. This attenuating effect on the amplitudes also implies dispersion, that is, phase velocity is also dependent on frequency [Fut62]. As mentioned in the previous section, attenuation can be implemented into the wave equation with different phenomenological models [Car14]. Each model of attenuation comes with a formula to calculate Q in the assumed model. Setting

$$\kappa_1 = \dots = \kappa_L = \kappa_0 \tau_{\mathrm{p}}\,, \tag{2.5}$$

then the formula to calculate $Q = Q_{\mathrm{GSLS}}$ in GSLS is

$$Q(\omega) = \frac{1 + \alpha_1(\omega)\,\tau_{\mathrm{p}}}{\alpha_2(\omega)\,\tau_{\mathrm{p}}} = \frac{\frac{1}{\tau_{\mathrm{p}}} + \alpha_1(\omega)}{\alpha_2(\omega)}\,, \tag{2.6}$$

with

$$\alpha_1(\omega) = \sum_{l=1}^{L} \frac{\omega^2\tau_l^2}{1+\omega^2\tau_l^2}\,, \qquad \alpha_2(\omega) = \sum_{l=1}^{L} \frac{\omega\tau_l}{1+\omega^2\tau_l^2}\,,$$

see [Boh02]. The representation in (2.6) shows a somewhat reciprocal relationship between the quantities Q and τ_{p}. Note that τ_{p} is a parameter which is introduced due to computational convenience in [BRS95] whereas Q is a quantity with physical meaning. The dependence of the phase velocity on the frequency in GSLS is expressed by the dispersion relation

$$v_{\mathrm{p}}(\omega) = \sqrt{\frac{\kappa_0\,(1+\alpha_1(\omega)\tau_{\mathrm{p}})}{\varrho}}\,. \tag{2.7}$$

Looking at α_1 we can see that it is monotone in ω, which results in anormal dispersion, that is, higher frequencies travel faster. Like mentioned above, in seismics we assume a constant quality Q_0 that depends on the subsurface we want to model. To achieve this behavior in GSLS over some frequency band $[\omega_1, \omega_2]$, the functional

$$J_{Q_0}(\tau_l, \tau_{\mathrm{p}}) = \int_{\omega_1}^{\omega_2} \left(Q^{-1}(\omega, \tau_l, \tau_{\mathrm{p}}) - Q_0^{-1}\right)^2 \,\mathrm{d}\omega\,,$$

is minimized for τ_{p} with τ_l's fixed a priori [BRS95] or with $\tau_{\mathrm{p}}, \tau_l$ as minimization parameters [Boh02]. Including more τ_l's allows for a better constant approximation. However, one wants to keep τ_l as low as possible, because an increase in the number of equations in (2.4) leads to a higher number of unknowns in numerical calculations. This results in longer computing times and greater memory requirements leads to more computing time and higher memory requirements. Typical choices for values of L are between 1 and 5.

Now, let $\omega_0 \in [\omega_1, \omega_2]$ be fixed and $\alpha_1 = \alpha_1(\omega_0)$. Plugging the assumption on the κ_l's (2.5) and the definition of v_{p} from (2.7) into the forward equation (2.4), we get

$$\begin{aligned}
\varrho(\boldsymbol{x})\partial_t \boldsymbol{v}(t,\boldsymbol{x}) &= \nabla \sum_{l=0}^{L} p_l(t,\boldsymbol{x}) + \boldsymbol{f}_v(t,\boldsymbol{x}) && \text{in } [0,\infty)\times\Omega\,,\\
\frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2}\partial_t p_0(t,\boldsymbol{x}) &= \operatorname{div}\boldsymbol{v}(t,\boldsymbol{x}) + f_{p_0}(t,\boldsymbol{x}) && \text{in } [0,\infty)\times\Omega\,,\\
\frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2\tau_{\mathrm{p}}(\boldsymbol{x})}\partial_t p_l(t,\boldsymbol{x}) &= \operatorname{div}\boldsymbol{v}(t,\boldsymbol{x}) - \frac{1}{\tau_l}\frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2\tau_{\mathrm{p}}(\boldsymbol{x})}p_l(t,\boldsymbol{x})\\
&\qquad + f_{p_l}(t,\boldsymbol{x}),\ \text{for } l=1,\dots L,\ \text{in } [0,\infty)\times\Omega\,.
\end{aligned} \tag{2.8}$$

From now on, we denote the vector containing the particle velocity $\boldsymbol{v}$ and the pressure vector $\boldsymbol{p} = (p_0,\dots,p_L)^\top$ as $\boldsymbol{u}(t,\boldsymbol{x}) = (\boldsymbol{v}(t,\boldsymbol{x}), \boldsymbol{p}(t,\boldsymbol{x}))^\top \in \mathbb{R}^{2+(1+L)}$. Define the operators

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}(t,\boldsymbol{x}) &= \left(\varrho(\boldsymbol{x})\boldsymbol{v}(t,\boldsymbol{x}), \frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2}p_0(t,\boldsymbol{x}),\dots,\frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2\tau_{\mathrm{p}}(\boldsymbol{x})}p_L(t,\boldsymbol{x})\right)\,,\\
\boldsymbol{D}(\boldsymbol{x})\boldsymbol{u}(t,\boldsymbol{x}) &= (\boldsymbol{0}, 0, \tau_1^{-1}p_1(t,\boldsymbol{x}),\dots,\tau_L^{-1}p_L(t,\boldsymbol{x}))\\
\boldsymbol{A}\boldsymbol{u}(t,\boldsymbol{x}) &= -(\nabla(p_0(t,\boldsymbol{x})+p_1(t,\boldsymbol{x})+\dots,+p_L(t,\boldsymbol{x})), \operatorname{div}\boldsymbol{v}(t,\boldsymbol{x}),\dots,\operatorname{div}\boldsymbol{v}(t,\boldsymbol{x}))\,.
\end{aligned} \tag{2.9}$$

If we further write $\boldsymbol{f}(t) = (\boldsymbol{f}_{\boldsymbol{v}}(t), f_{p_0}(t), f_{p_1}(t),\dots,f_{p_L}(t))$, we can rewrite (2.8) as

$$\mathfrak{L}\boldsymbol{u}(t,\boldsymbol{x}) = \boldsymbol{M}(\boldsymbol{x})\partial_t\boldsymbol{u}(t,\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{u}(t,\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{D}(\boldsymbol{x})\boldsymbol{u}(t,\boldsymbol{x}) = \boldsymbol{f}(t,\boldsymbol{x}) \qquad \text{in } [0,\infty)\times\Omega\,. \tag{2.10}$$

Combined with initial value $\boldsymbol{u}(0) = \boldsymbol{u}(0,x) = \boldsymbol{u}_0$ and spatial boundary conditions, this can be written as an initial value problem in time. For the end time $T > 0$, find $\boldsymbol{u}$, such that

$$\mathfrak{L}\boldsymbol{u}(t) = \boldsymbol{f}(t)\,, \qquad t \in [0,T], \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{2.11}$$

This equation is purely symbolic since we have no proper definition of spaces where these formal operators act on. In the next chapter we will put this in a more formal frame.

Remark 1. *Note that in (2.8) the forcing term f_p should be divided by $\frac{1+\alpha_1\tau_{\mathrm{p}}(\boldsymbol{x})}{\varrho(\boldsymbol{x})v_{\mathrm{p}}(\boldsymbol{x})^2}$, as the source is added to the second equation in (2.3). In all reconstructions we assume that the material at the source position is known. Therefore, we can incorporate that factor into $\boldsymbol{f}$ without having to consider it later in the formulas for the inversion.*

2.2 Existence theory

In this section we want to give a short existence and uniqueness result for solutions of equation (2.8). Additionally, we want to repeat the results used for the proof from semigroup theory. Detailed proofs for existence and uniqueness of solutions in a more general setting for the visco-elastic equations can be found in [Zel19, KR19], where the results for the visco-acoustic case are a simple corollary. Again, let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain with disjoint decomposition of the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. We define

$$\begin{aligned}
&\boldsymbol{X} := L^2(\Omega, \mathbb{R}^{2+(1+L)}) \qquad \boldsymbol{Y} := L^2([0,T], \boldsymbol{X})\,,\\
&\mathcal{D}(\boldsymbol{A}) := \{\boldsymbol{u} = (\boldsymbol{v}, p_0, p_1, \ldots, p_L) \in \boldsymbol{X} \mid \boldsymbol{A}\boldsymbol{u} \in \boldsymbol{X}, \mathbf{n}\cdot\boldsymbol{v}|_{\partial\Omega_N} = 0, (p_0 + \cdots + p_L)|_{\partial\Omega_D} = 0\},
\end{aligned}$$

where we use the usual notation of the L^p-spaces, $p \in [0,\infty]$ related to the Lebesgue measure. Equipped with the scalar products

$$\begin{aligned}
&(\cdot,\cdot)_{0,\Omega} : \boldsymbol{X} \times \boldsymbol{X} \longrightarrow \mathbb{R},\ (\boldsymbol{u}_1, \boldsymbol{u}_2) \longmapsto \int_\Omega \boldsymbol{u}_1(\boldsymbol{x}) \cdot \boldsymbol{u}_2(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\,,\\
&(\cdot,\cdot)_{0,[0,T]\times\Omega} : \boldsymbol{Y} \times \boldsymbol{Y} \longrightarrow \mathbb{R},\ (\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2) \longmapsto \int_{[0,T]} \int_\Omega \tilde{\boldsymbol{u}}_1(t,\boldsymbol{x}) \cdot \tilde{\boldsymbol{u}}_2(t,\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}t\,,
\end{aligned}$$

the spaces $\boldsymbol{X}$ and $\boldsymbol{Y}$ are Hilbert spaces. By $\|\cdot\|^2_{0,\Omega} = (\cdot,\cdot)_{0,\Omega}$ and $\|\cdot\|^2_{0,[0,T]\times\Omega} = (\cdot,\cdot)_{0,[0,T]\times\Omega}$ we denote the norms induced by the scalar products. For $\boldsymbol{u} = (\boldsymbol{u}_{\boldsymbol{v}}, \boldsymbol{u}_{p_0}, \boldsymbol{u}_{p_1}, \ldots, \boldsymbol{u}_{p_L}), \boldsymbol{y} = (\boldsymbol{y}_{\boldsymbol{v}}, \boldsymbol{y}_{p_0}, \boldsymbol{y}_{p_1}, \ldots, \boldsymbol{y}_{p_L}) \in \mathcal{D}(\boldsymbol{A})$ we get with the divergence theorem

$$\begin{aligned}
(\boldsymbol{A}\boldsymbol{u}, \boldsymbol{y})_{0,\Omega} &= \sum_{l=0}^{L} \big(\operatorname{div} \boldsymbol{u}_{\boldsymbol{v}}, \boldsymbol{y}_{p_l}\big)_{0,\Omega} + \sum_{l=0}^{L} (\nabla \boldsymbol{u}_{p_l}, \boldsymbol{y}_{\boldsymbol{v}})_{0,\Omega}\\
&= -\sum_{l=0}^{L} \big(\boldsymbol{u}_{\boldsymbol{v}}, \nabla \boldsymbol{y}_{p_l}\big)_{0,\Omega} - \sum_{l=0}^{L} (\boldsymbol{u}_{p_l}, \operatorname{div} \boldsymbol{y}_{\boldsymbol{v}})_{0,\Omega}\\
&= (\boldsymbol{u}, -\boldsymbol{A}\boldsymbol{y})_{0,\Omega}\ .
\end{aligned}$$

Therefore, the operator $\boldsymbol{A}$ is skew symmetric on the chosen domain. The multiplication operators $\boldsymbol{M}, \boldsymbol{D}$ depend on the values of scalar parameter functions $\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}} \colon \Omega \longrightarrow \mathbb{R}$. For our applications it is physically meaningful to assume that functions are bounded from below and from above, that is, there are bounds

$0 < \varrho_{\min} < \varrho_{\max},\ 0 < v_{\mathrm{p},\min} < v_{\mathrm{p},\max},\ 0 < \tau_{p,\min} < \tau_{p,\max}$, which gives us a set of physically admissible parameter functions

$$\boldsymbol{P}_{\mathrm{adm}} := \{\mu \in L^\infty(\Omega, \mathbb{R}^3) \colon \mu(\boldsymbol{x}) \in [\varrho_{\min}, \varrho_{\max}] \times [v_{\mathrm{p},\min}, v_{\mathrm{p},\max}] \times [\tau_{\mathrm{p},\min}, \tau_{\mathrm{p},\max}]\}\,. \tag{2.12}$$

From now on we denote the mappings that map a set of admissible parameters $\mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})^\top \in \boldsymbol{P}_{\mathrm{adm}}$ to the corresponding operator from (2.9) and (2.10) by $\boldsymbol{M}(\mu), \boldsymbol{D}(\mu)$, and $\boldsymbol{\mathfrak{L}}(\mu)$. Since $\boldsymbol{M}$ and $\boldsymbol{D}$ are multiplication operators with non-negative factors, for $\mu \in \boldsymbol{P}_{\mathrm{adm}}$ and $\boldsymbol{u} \neq \boldsymbol{0}$ we get

$$C_1 \|\boldsymbol{u}\|_{0,\Omega}^2 \geq (\boldsymbol{M}(\mu)\boldsymbol{u}, \boldsymbol{u})_{0,\Omega} > C_2 \|\boldsymbol{u}\|_{0,\Omega}^2,\ (\boldsymbol{D}(\mu)\boldsymbol{u}, \boldsymbol{u})_{0,\Omega} \geq 0 \tag{2.13}$$

with $C_1, C_2 > 0$ depending on the bounds in $\boldsymbol{P}_{\mathrm{adm}}$.We present a concise part of the existence theory for linear evolution equations and apply it to (2.11).

Let H be a Hilbert space, and B be a (typically unbounded) linear operator in H with domain $\mathcal{D}(B)$. Let $T > 0$ and $f(t) \in H$, for $t \in [0, T]$, then we seek a function $u(t) \in \mathcal{D}(B)$, for $t \in [0, T]$ that fulfills the initial value problem

$$\partial_t u(t) = -Bu(t) + f(t)\,,\ t \in [0, T],\ u(0) = u_0\,, \tag{2.14}$$

with $u_0 \in \mathcal{D}(B)$. We present some conditions on B and f that ensure the existence of a unique solution of this evolution equation, utilizing semigroup theory. Thus, the notion of semigroups is introduced.

Definition 1. *Let W be a Banach space. A map $S(\cdot) \colon \mathbb{R}_{\geq 0} \longrightarrow \mathcal{L}(W)$ is called a* strongly continuous operator semigroup *if it satisfies*

1. *$S(0) = I$ and $S(t + s) = S(t)S(s)$ for all $t, s \in \mathbb{R}_{\geq 0}$,*
2. *for each $w \in W$ $S(\cdot)w \colon \mathbb{R}_{\geq 0} \longrightarrow W,\ t \longmapsto S(t)w$ is continuous.*

The generator *B of $S(\cdot)$ is given by*

$$Bw = \lim_{h \longrightarrow 0+} (S(h)w - w)/h\,,\ \text{for } \mathcal{D}(B) = \left\{ w \in W |\lim_{h \longrightarrow 0+} (S(h)w - w)/h \text{ exists in } W \right\}.$$

With this concept the existence of a possibly non-differentiable function as a solution is usually shown first.

Definition 2. *Let B be the generator of the strongly continuous operator semigroup $S(\cdot)$, $u_0 \in W$, $f \in L^1([0, T], W)$, then we call $u \in C([0, T], W)$ defined as*

$$u(t) = S(t)u_0 + \int_0^t S(t - s)f(s)\,\mathrm{d}s$$

a mild solution *of* (2.14).

Now, we combine several standard theorems into a single theorem which yields the necessary existence results required for this thesis.

Theorem 1. *Let H be a Hilbert space, and let B be a linear operator in H with domain $\mathcal{D}(B)$ satisfying the following conditions:*

1. *$(y, By)_H \geq 0$ for every $y \in \mathcal{D}(B)$.*

2. *B is* maximal monotone *that is $I + B$ from $\mathcal{D}(B) \longrightarrow H$ is surjective.*

Then, for $u_0 \in \mathcal{D}(B)$ the initial value problem (2.14) *has a*

(a) *unique mild solution $u \in C\left([0,T], H\right)$ for $f \in L^1([0,T], H)$.*

(b) *unique* classical solution $u \in C^1\left([0,T], H\right) \cap C([0,T], \mathcal{D}(B))$ *for $f \in C\left([0,T], H\right) \cap W^{1,1}\left([0,T], H\right)$.*

Proof. For the existence of a mild solution one has to show that B is the generator of a strongly continuous operator semigroup. Assumptions 1. and 2. are sufficient for this. The combination of properties of maximal monotone operators [Bre11, Proposition 7.1] and the Lumer-Phillips theorem [RR04, 12.22] imply that B generates strongly continuous operator semigroup. (it implies even more, that is $\|S(t)\| \leq 1, \forall t \in [0,T]$). Therefore, if we assume $f \in L^1([0,T], H)$ this yields a mild solution in the sense of Definition 2. Assuming additionally that $f \in C\left([0,T], H\right) \cap W^{1,1}\left([0,T], H\right)$, the mild solution can be shown to be differentiable and thus is a classical solution (see [RR04, Theorems 12.16]). □

We want to apply the theorem above to our setting to obtain a classical solution.

Theorem 2. *Let $\boldsymbol{f} \in C\left([0,T], \boldsymbol{X}\right) \cap W^{1,1}\left([0,T], \boldsymbol{X}\right)$ and $\mu \in \boldsymbol{P}_{\mathrm{adm}}$. Then, the initial value problem of the visco-acoustic wave equation*

$$\boldsymbol{\mathfrak{L}}(\mu)\boldsymbol{u}(t) = \boldsymbol{M}(\mu)\partial_t \boldsymbol{u}(t) + (\boldsymbol{A} + \boldsymbol{M}(\mu)\boldsymbol{D}(\mu))\boldsymbol{u}(t) = \boldsymbol{f}(t), \;\; t \in [0,T], \;\; \boldsymbol{u}(0) = 0 \tag{2.15}$$

has a unique solution $\boldsymbol{u} \in C^1\left([0,T], \boldsymbol{X}\right) \cap C\left([0,T], \mathcal{D}(\boldsymbol{A})\right)$.

Proof. For a more compact notation we suppress the dependence of the operators $\boldsymbol{M}, \boldsymbol{D}$ on μ. We also use the parameterization from (2.4) because it is less cluttered. Then, $\boldsymbol{M}$ has the form

$$\boldsymbol{M}\boldsymbol{u} = \left(\varrho \boldsymbol{v}, \kappa_0^{-1} p_0, \kappa_1^{-1} p_1, \ldots, \kappa_L^{-1} p_L\right),$$

with $\kappa_0 = v_{\mathrm{p}}^2 \varrho (1 + \alpha_1 \tau_{\mathrm{p}})^{-1}$, $\kappa_l = \tau_{\mathrm{p}} \kappa_0$, $l = 1, \ldots, L$. We want to apply Theorem 1, for this reason we transform the equation (2.15) to

$$\partial_t \boldsymbol{u}(t) = -\boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D})\boldsymbol{u}(t) + \boldsymbol{M}^{-1}\boldsymbol{f}(t),$$

with $f = \boldsymbol{M}^{-1}\boldsymbol{f}$ and $B = \boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D})$ we are in the right setting. Also, since $\boldsymbol{M}, \boldsymbol{D}$ are invertible bounded linear operators, it holds that $\mathcal{D}(\boldsymbol{A}) = \mathcal{D}(\boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D}))$. We also equip $\boldsymbol{X}$ with the energy scalar product $(\boldsymbol{u}, \boldsymbol{v})_{\boldsymbol{M},\Omega} := (\boldsymbol{M}\boldsymbol{u}, \boldsymbol{v})_{0,\Omega}$ which induces an equivalent norm due to (2.13).

$$\left(\boldsymbol{x}, \boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D})\boldsymbol{x}\right)_{\boldsymbol{M},\Omega} = \left(\boldsymbol{x}, \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x}\right)_{\boldsymbol{M},\Omega} + (\boldsymbol{x}, \boldsymbol{D}\boldsymbol{x})_{\boldsymbol{M},\Omega} \geq (\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})_{0,\Omega} = 0.$$

In the last step we used the skew-symmetry of $\boldsymbol{A}$. We need to show that $\boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D}) + \boldsymbol{I}$ is onto. Therefore, for every $\boldsymbol{g} = (\boldsymbol{g}_v, g_0, \ldots, g_L) \in \boldsymbol{X}$ we need to find $(\boldsymbol{v}, p_0, \ldots, p_L) \in \mathcal{D}(\boldsymbol{A})$ such that

$$\boldsymbol{v} - \frac{1}{\varrho}\nabla \sum_{l=0}^{L} p_l = \boldsymbol{g}_v\,, \tag{2.16}$$

$$p_0 - \kappa_0 \operatorname{div} \boldsymbol{v} = g_0\,, \tag{2.17}$$

$$\left(1 + \frac{1}{\tau_l}\right) p_l - \kappa_l \operatorname{div} \boldsymbol{v} = g_l, \qquad l = 1, \ldots, L. \tag{2.18}$$

The trick of the proof is to find an auxiliary variational problem whose solution in turn gives us such an element in $\mathcal{D}(\boldsymbol{A})$. In order to define this problem we add the pressures

$$p = \sum_{l=0}^{L} p_l = g_0 + \kappa_0 \operatorname{div} \boldsymbol{v} + \sum_{l=1}^{L} \frac{\tau_l}{\tau_l + 1}(g_l + \kappa_l \operatorname{div} \boldsymbol{v}) = \tilde{g} + \tilde{\kappa} \operatorname{div} \boldsymbol{v}\,,$$

with $\tilde{\kappa} := \kappa_0 + \sum_{l=1}^{L} \frac{\tau_l}{\tau_l+1}\kappa_l, \tilde{g} := g_0 + \sum_{l=1}^{L} \frac{\tau_l}{\tau_l+1} g_l$. Multiply this equation with a test function $\psi \in Z = \{\psi \in H^1(\Omega)\colon \psi|_{\partial\Omega_D} = 0\}$ and assume $(\boldsymbol{v}, \boldsymbol{0})^\top \in \mathcal{D}(\boldsymbol{A})$. Integrating over Ω and using partial integration we get

$$\begin{aligned}
\left(\tilde{\kappa}^{-1} p, \psi\right)_{0,\Omega} &= (\operatorname{div} \boldsymbol{v}, \psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega} \\
&= -(\boldsymbol{v}, \nabla\psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega} + \int_{\partial\Omega} (\psi \boldsymbol{v}) \cdot \boldsymbol{n} \,\mathrm{d}s \\
&\overset{\psi\in Z}{=} -(\boldsymbol{v}, \nabla\psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega} + \int_{\partial\Omega_N} (\psi \boldsymbol{v}) \cdot \boldsymbol{n} \,\mathrm{d}s \qquad (2.19)\\
&= -(\boldsymbol{v}, \nabla\psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega}\,.
\end{aligned}$$

Plugging (2.16) into the equation above yields

$$\begin{aligned}
\left(\tilde{\kappa}^{-1} p, \psi\right)_{0,\Omega} &= -\left(\varrho^{-1}\nabla p + \boldsymbol{g}_{\boldsymbol{v}}, \nabla\psi\right)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega} \\
\Leftrightarrow \left(\varrho^{-1}\nabla p, \nabla\psi\right)_{0,\Omega} + \left(\tilde{\kappa}^{-1} p, \psi\right)_{0,\Omega} &= -(\boldsymbol{g}_{\boldsymbol{v}}, \nabla\psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega}\,. \qquad (2.20)
\end{aligned}$$

In the derivation of the equation above we assumed $(\boldsymbol{v}, \boldsymbol{0})^\top \in \mathcal{D}(\boldsymbol{A})$. Now, if we have $\hat{p} \in Z$ that fulfills (2.20), we can “walk the steps backwards” and construct a corresponding $\hat{\boldsymbol{v}}$ that, in fact, lies in the domain of the differential operator. To achieve, this we define the operators

$$\begin{aligned}
&a\colon Z \times Z \longrightarrow \mathbb{R},\ (p, \psi) \longmapsto \left(\varrho^{-1}\nabla p, \nabla\psi\right)_{0,\Omega} + \left(\tilde{\kappa}^{-1} p, \psi\right)_{0,\Omega}\,, \\
&l\colon Z \longrightarrow \mathbb{R},\ \psi \longmapsto -(\boldsymbol{g}_v, \nabla\psi)_{0,\Omega} + \left(\tilde{\kappa}^{-1}\tilde{g}, \psi\right)_{0,\Omega}\,,
\end{aligned}$$

and the corresponding variational problem

$$\text{find } p \in Z \text{ such that } a(p, \psi) = l(\psi),\ \forall\psi \in Z\,.$$

The functional on the right-hand side l is obviously bounded. Using Poincaré's inequality and the boundedness and positivity of the parameters, we can show that the bilinear form a is uniformly elliptic. Therefore, we can apply the Theorem of Lax-Milgram [RR04, Theorem 9.14] which gives us a unique $\hat{p} \in Z$ with $a(\hat{p}, \psi) = l(\psi)\ \forall\psi \in Z$. Define $\hat{\boldsymbol{v}} = \boldsymbol{g}_v + \varrho^{-1}\nabla\hat{p}$, then rearranging (2.20) yields

$$-(\hat{\boldsymbol{v}}, \nabla\psi)_{0,\Omega} = \left(\tilde{\kappa}^{-1}\tilde{}(\hat{p} - \tilde{g}), \psi\right)_{0,\Omega} \forall\psi \in Z. \qquad (2.21)$$

Note that (2.19) is the variational formulation of (2.16).Therefore, if we compare (2.19) and (2.21), we can see that $\hat{\boldsymbol{v}} \cdot \boldsymbol{n} = 0$ on $\partial\Omega_N$ holds in a variational sense. Further, this equivalence shows $\operatorname{div} \hat{\boldsymbol{v}} = \tilde{\kappa}^{-1}(\hat{p} - \tilde{g}) \in L^2(\Omega)$. Therefore, we have shown $(\hat{\boldsymbol{v}}, \boldsymbol{0})^\top \in \mathcal{D}(\boldsymbol{A})$. Plugging this $\hat{\boldsymbol{v}}$ into (2.17) and (2.18) gives us the individual pressure functions $\hat{p}_0, \dots, \hat{p}_L$. Together, we showed for every $\boldsymbol{g} \in \boldsymbol{X}$ there is a $\hat{\boldsymbol{u}} = (\hat{\boldsymbol{v}}, \hat{p}_0, \hat{p}_1, \dots, \hat{p}_L) \in \mathcal{D}(\boldsymbol{A})$ such that $(\boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{M}\boldsymbol{D}) + \boldsymbol{I})\hat{\boldsymbol{u}} = \boldsymbol{g}$. This concludes the proof that the operator is onto. Now that we have shown the prerequisites for Theorem 1, we get that the evolution equation has a solution $u \in C^1([0,T], \boldsymbol{X}) \cap C([0,T], \mathcal{D}(\boldsymbol{A}))$. □

Remark 2. *Note that in this work we are not interested in assuming low regularity. In fact, we will always assume that our right-hand sides are in C_c^∞ and shifted such that they vanish at $t = 0$ and the initial value $\boldsymbol{u}_0$ is zero. Therefore Theorem 3.3 in [KR19] gives us smooth solutions in time.*

2.3 Adjoint equation

Later on in Chapter 4 the adjoint equation to (2.8) is needed. Here, we calculate its representation with respect to the scalar product $(\cdot,\cdot)_{0,[0,T]\times\Omega}$. Let $\mathfrak{L}$ be the differential operator defined in (2.10). For this we seek suitable functions $\boldsymbol{u}, \boldsymbol{y} \in C^1([0,T], \mathcal{D}(\boldsymbol{A}))$ and an operator $\mathfrak{L}^\star$ such that it holds

$$(\mathfrak{L}\boldsymbol{u}, \boldsymbol{y})_{0,[0,T]\times\Omega} = (\boldsymbol{u}, \mathfrak{L}^\star \boldsymbol{y})_{0,[0,T]\times\Omega}\,.$$

Using the skew-symmetry of $\boldsymbol{A}$, symmetry of $\boldsymbol{M}, \boldsymbol{D}$, and integration by parts, we get

$$\begin{aligned}(\mathfrak{L}\boldsymbol{u}, \boldsymbol{y})_{0,[0,T]\times\Omega} &= \int_0^T (\boldsymbol{M}\partial_t \boldsymbol{u}(t), \boldsymbol{y}(t))_{0,\Omega} + (\boldsymbol{M}\boldsymbol{D}\boldsymbol{u}(t), \boldsymbol{y}(t))_{0,\Omega} + (\boldsymbol{A}\boldsymbol{u}(t), \boldsymbol{y}(t))_{0,\Omega} \,\mathrm{d}t \\ &= \int_0^T (\boldsymbol{u}(t), -\boldsymbol{M}\partial_t \boldsymbol{y}(t))_{0,\Omega} + (\boldsymbol{u}, \boldsymbol{M}\boldsymbol{D}\boldsymbol{y}(t))_{0,\Omega} + (\boldsymbol{u}(t), -\boldsymbol{A}\boldsymbol{y}(t))_{0,\Omega} \,\mathrm{d}t \\ &\quad + (\boldsymbol{M}\boldsymbol{u}(T), \boldsymbol{y}(T))_{0,\Omega} - (\boldsymbol{M}\boldsymbol{u}(0), \boldsymbol{y}(0))_{0,\Omega} \\ &= (\boldsymbol{u}, (-\boldsymbol{M}\partial_t - \boldsymbol{A} + \boldsymbol{M}\boldsymbol{D})\boldsymbol{y})_{0,[0,T]\times\Omega} \\ &\quad + (\boldsymbol{M}\boldsymbol{u}(T), \boldsymbol{y}(T))_{0,\Omega} - (\boldsymbol{M}\boldsymbol{u}(0), \boldsymbol{y}(0))_{0,\Omega}\,.\end{aligned}$$

For functions satisfying $(\boldsymbol{M}\boldsymbol{u}(T), \boldsymbol{y}(T))_{0,\Omega} = (\boldsymbol{M}\boldsymbol{u}(0), \boldsymbol{y}(0))_{0,\Omega}$ a notion of adjointness can be seen. We restrict ourselves to pairs of functions with the properties $\boldsymbol{u}(0) = \boldsymbol{y}(T) = \boldsymbol{0}$. For these functions we can define an adjoint

$$\mathfrak{L}^\star = -\boldsymbol{M}\partial_t - \boldsymbol{A} + \boldsymbol{M}\boldsymbol{D}\,. \tag{2.22}$$

Note that we use $\star$ as superscript to mark the adjoint in regard to the differential equation and use $*$ at most other occasions. Solving the adjoint equation requires finding the solution to the terminal value problem

$$-\boldsymbol{M}\partial_t \boldsymbol{z}(t) - \boldsymbol{A}\boldsymbol{z}(t) + \boldsymbol{M}\boldsymbol{D}\boldsymbol{z}(t) = \boldsymbol{g}(t) \quad \boldsymbol{z}(T) = 0.$$

We can cast it into an initial value problem with the transformation $\tilde{t}(t) = T - t$. The chain rule gives us $\partial_t \boldsymbol{z}(\tilde{t}(t)) = \partial_{\tilde{t}(t)} \boldsymbol{z}(\tilde{t}(t)) \cdot \partial_t \tilde{t}(t) = -\partial_{\tilde{t}(t)} \boldsymbol{z}(\tilde{t}(t))$. Dropping the dependency on t we get

$$\boldsymbol{M}\partial_{\tilde{t}} \boldsymbol{z}(\tilde{t}) - \boldsymbol{A}\boldsymbol{z}(\tilde{t}) + \boldsymbol{M}\boldsymbol{D}\boldsymbol{z}(\tilde{t}) = \boldsymbol{g}(\tilde{t}) \quad \boldsymbol{z}(0) = 0. \tag{2.23}$$

This shows that we can solve an initial value problem instead. For this initial value problem the theory from section Section 2.2 is applied easily. Later, we will not distinguish between t and $\tilde{t}$, the time will always be called t. Looking at (2.23) we can see that the attenuation operator $\boldsymbol{D}$ has the same sign as in the forward equation (2.8). Hence, both equations exhibit damping properties.

2.4 Measurement operator

In real world applications of FWI, data is gathered with instruments that are called receivers. While both measurements of components of $\boldsymbol{v}$ and $p = \sum_{l=0}^{L} p_l$ are possible, we restrict ourselves to p. We assume to have $R \in \mathbb{N}$ receivers at positions $\boldsymbol{r}_0, \dots, \boldsymbol{r}_{R-1} \in \Omega$, which measure at discrete times $\gamma_0, \dots, \gamma_{J-1}$, with $\gamma_j = \gamma_0 + j\,\Delta t$ for $j = 0, \dots, J-1$ and $\Delta t \in \mathbb{R}_+$. Define $\widetilde{\boldsymbol{S}} := \mathbb{R}^{R \times J}$ and the measurement operator

$$\Psi \colon \boldsymbol{Y} \longrightarrow \widetilde{\boldsymbol{S}},\ \boldsymbol{u} = (\boldsymbol{v}, \boldsymbol{p})^\top \longmapsto \tilde{\boldsymbol{s}} = \begin{pmatrix} (\boldsymbol{u}, \phi_{\boldsymbol{r}_0,\gamma_0})_{0,[0,T]\times\Omega} & \cdots & (\boldsymbol{u}, \phi_{\boldsymbol{r}_0,\gamma_{J-1}})_{0,[0,T]\times\Omega} \\ \vdots & \ddots & \vdots \\ (\boldsymbol{u}, \phi_{\boldsymbol{r}_{R-1},\gamma_0})_{0,[0,T]\times\Omega} & \cdots & (\boldsymbol{u}, \phi_{\boldsymbol{r}_{R-1},\gamma_{J-1}})_{0,[0,T]\times\Omega} \end{pmatrix},$$

where $\phi_{\boldsymbol{r}_r,\gamma_j} \colon [0,T] \times \Omega \longrightarrow \mathbb{R}^{2+(1+L)}$ is a function with compact support that models the measurement process at the location of the r-th receiver at the j-th time (think of a smooth approximation to the delta distribution). On this space of seismograms a scalar product can be defined by

$$(\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2)_{\widetilde{\boldsymbol{S}}} = \sum_{r=0}^{R-1} \tilde{\boldsymbol{s}}_{1,r} W \tilde{\boldsymbol{s}}_{2,r}^\top\,, \text{ with } W = \Delta t\, \mathrm{diag}(0.5, 1, \dots, 1, 0.5) \in \mathbb{R}^{J\times J}\,,$$

where $\tilde{\boldsymbol{s}}_{i,r}$ denotes the r-th row of $\tilde{\boldsymbol{s}}_i$ for $i \in \{1,2\}$. Later, we need the adjoint of the measurement operator. Note that in Chapter 2 we denoted the relaxation function by Ψ. Since the relaxation function was only of interest in Chapter 2 we can reuse the symbol without creating confusion later on. Consider $\tilde{\boldsymbol{s}} \in \widetilde{\boldsymbol{S}}$ and $\boldsymbol{u} \in \boldsymbol{Y}$, then

$$\begin{aligned} (\tilde{\boldsymbol{s}}, \Psi\boldsymbol{u})_{\widetilde{\boldsymbol{S}}} &= \sum_{r=0}^{R-1} \tilde{\boldsymbol{s}}_r W (\Psi\boldsymbol{u})_r^\top \\ &= \sum_{r=0}^{R-1}\sum_{j=0}^{J-1} W_{jj} \tilde{\boldsymbol{s}}_{j,r} \int_0^T \int_\Omega \boldsymbol{u}(t,\boldsymbol{x}) \cdot \phi_{\boldsymbol{r}_r,\gamma_j}(t,\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}t \\ &= \int_0^T \int_\Omega \sum_{r=0}^{R-1}\sum_{j=0}^{J-1} W_{jj} \tilde{\boldsymbol{s}}_{j,r} \phi_{\boldsymbol{r}_r,\gamma_j}(t,\boldsymbol{x}) \cdot \boldsymbol{u}(t,\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}t\,. \end{aligned}$$

Hence, the adjoint is defined by

$$\Psi^* \colon \widetilde{\boldsymbol{S}} \longrightarrow \boldsymbol{Y},\ \tilde{\boldsymbol{s}} \longmapsto \sum_{r=0}^{R-1}\sum_{j=0}^{J-1} W_{jj} \tilde{\boldsymbol{s}}_{j,r} \phi_{\boldsymbol{r}_r,\gamma_j}(t,\boldsymbol{x})\,.$$

The evaluation of the adjoint measurement operator on a seismogram is the wave field that has the support of the measurement kernels. Its values are multiplied by the values of the seismogram. Often it is convenient to interpret seismograms as continuous functions in time as opposed to a matrix. Hence, we define the interpolation space

$$\boldsymbol{S} := \{\boldsymbol{s} \in C([0,T], \mathbb{R}^R) \colon (\boldsymbol{s}(t))_r \text{ is a linear spline with nodes } \gamma_0, \dots, \gamma_{J-1} \text{ for all } r \in R\}. \tag{2.24}$$

Then, the interpolation operator $\iota \colon \widetilde{\boldsymbol{S}} \longrightarrow \boldsymbol{S}$ is bijective and an isometry from $\widetilde{\boldsymbol{S}}$ to $L^2([0,T], \mathbb{R}^R)$ since the weights W yield exact integration for linear functions. Therefore, we define the scalar product on $\boldsymbol{S}$ via

$$(\boldsymbol{s}_1, \boldsymbol{s}_2)_{\boldsymbol{S}} := \left(\iota^{-1}\boldsymbol{s}_1, \iota^{-1}\boldsymbol{s}_2\right)_{\widetilde{\boldsymbol{S}}}\,.$$

Usually, in practical applications, data gathers for inversion is collected in multiple phases. Human-made sources of pressure waves, such as explosions, are excited at different locations and times and are individually measured. Each instance of these excitations are is referred to as a *shot*. The number of shots is $\Xi \in \mathbb{N}$, and therefore our data typically lies in $\boldsymbol{S}^\Xi$.

CHAPTER THREE

DISCRETIZATION

Now, we shift our focus to some discretization aspects. Since discretization itself is not the main topic of this thesis we will provide a concise overview and give references in which the topics are explained in more depth.

3.1 Space Discretization

In space we choose a *discontinuous Galerkin* (dG) discretization with full upwind stabilization which relies on ideas from [HW08, LeV02], and was presented for waves without attenuation in [HPS$^+$15] (For a general introduction to finite element methods we refer to [EG04]). The full upwind stabilization for dG was first introduced for conservation laws of the form $\partial_t \boldsymbol{u} + \operatorname{div}(T\boldsymbol{u}) = 0$ with a suitable linear operator T and relies on traveling wave solutions of Riemann problems (initial value problems with discontinuous initial data along a hyperplane). Due to the operator $\boldsymbol{D}$ from (2.9) our problem is not in the divergence form. However, neglecting $\boldsymbol{D}$, the ansatz yields a suitable stabilization for the visco-acoustic (as special case of the visco-elastic case) wave equation as shown in [Zie20]. For the discretization in space the domain Ω is decomposed into open convex polyhedral sets $K \subset \Omega$, called *cells*, that is, $\overline{\Omega} = \bigcup_{K \in \mathcal{K}} \overline{K}$ ($\mathcal{K}$ is the set of all cells). The discontinuous finite element space of polynomials of maximal order $k \geq 0$ is defined by

$$\boldsymbol{X}_h := \left\{ \boldsymbol{x}_h \in \mathrm{L}^2(\Omega, \mathbb{R}^{\dim + (1+L)}) \colon \boldsymbol{x}_h|_K \text{ component-wise polynomial of max. order } k \ \forall\, K \in \mathcal{K} \right\}. \tag{3.1}$$

As ansatz and test functions, we use nodal Lagrangian ansatz functions which are denoted by $\boldsymbol{\varphi}_{i,K} \in \boldsymbol{X}_h$ for the node with index i in cell K. All space integrals are evaluated with a Gaussian quadrature and therefore products of ansatz and test functions are integrated exactly. Let $\mathcal{F}_K$ be the set of faces belonging to one cell K, and set $\mathcal{F} = \bigcup_K \mathcal{F}_K$ for all faces in the domain. For inner faces $f \in \mathcal{F} \cap \Omega$, let K_f be the neighboring cell such that $f = \partial K \cap \partial K_f$. On boundary faces $f \in \mathcal{F} \cap \partial\Omega$ we set $K_f = K$. Let $\boldsymbol{n}_{K,f}$ be the outer unit normal vector on the face f of K. For $\boldsymbol{x}_h \in \boldsymbol{X}_h$ the restriction to K is denoted by $\boldsymbol{x}_{h,K} = \boldsymbol{x}_h|_K$. On inner faces $f \in \mathcal{F} \cap \Omega$, we define the jump on the face of a cell by $[\boldsymbol{x}_h]_{K,f} = \boldsymbol{x}_{h,K_f} - \boldsymbol{x}_{h,K}$. On boundary faces $f \in \mathcal{F} \cap \partial\Omega_N$, we set $\boldsymbol{n}_{K,f} \cdot [\boldsymbol{v}_h]_{K,f} = 0$ and $[p_h]_{K,f} = -2p_h$, and on $f \in \mathcal{F} \cap \partial\Omega_D$ we set $[\boldsymbol{v}_h]_{K,f} = -2\boldsymbol{v}_h$ and $[p_h]_{K,f} = 0$. The operators without spatial derivative are discretized by

$$\begin{aligned}
(\boldsymbol{M}_h \boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} &= (\boldsymbol{M}\boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} \quad \forall \boldsymbol{x}_h, \boldsymbol{y}_h \in \boldsymbol{X}_h\,,\\
(\boldsymbol{D}_h \boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} &= (\boldsymbol{D}\boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} \quad \forall \boldsymbol{x}_h, \boldsymbol{y}_h \in \boldsymbol{X}_h\,,\\
(\boldsymbol{G}_h \boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} &= (\boldsymbol{I}\boldsymbol{x}_h, \boldsymbol{y}_h)_{0,\Omega} \quad \forall \boldsymbol{x}_h, \boldsymbol{y}_h \in \boldsymbol{X}_h\,.
\end{aligned}$$

For $\boldsymbol{x} \in \Omega$, define the impedance

$$Z(\boldsymbol{x}) := \sqrt{(1 + L\tau_{\mathrm{p}}(\boldsymbol{x}))/(1+\alpha\tau_{\mathrm{p}})(\boldsymbol{x})}\, \varrho(\boldsymbol{x}) v_{\mathrm{p}}(\boldsymbol{x}),\ Z_K := Z|_K, Z_{K_f} := Z|_{K_f},$$

and define functions on the faces of cells $\alpha_1 = 1/(Z_K + Z_{K_f})$, $\alpha_2 = Z_K Z_{K_f}/(Z_K + Z_{K_f})$, $\alpha_3 = Z_{K_f}/(Z_K + Z_{K_f})$, $\alpha_4 = Z_K/(Z_K + Z_{K_f})$. The upwind flux scheme (for the calculation of the flux we, again, refer to the references above) then gives us the upwind stabilized differential operator $\boldsymbol{A}_h = \sum_{K\in\mathcal{K}} \boldsymbol{A}_{K,h}$. Let $(\boldsymbol{v}_h, \boldsymbol{p}_h), (\boldsymbol{w}_h, \boldsymbol{q}_h) \in \boldsymbol{X}_h$ and define $p_h := \sum_l^L p_{l,h}, q := \sum_l^L q_{l,h}$ as sum over the components of $\boldsymbol{p}_h, \boldsymbol{q}_h$. Then, we can write the operator in each cell as

$$\begin{aligned}
(\boldsymbol{A}_{K,h}(\boldsymbol{v}_h, \boldsymbol{p}_h), (\boldsymbol{w}_h, \boldsymbol{q}_h))_{0,K} = &-(\nabla p_{K,h}, \boldsymbol{w}_{K,h})_{0,K} - (\operatorname{div} \boldsymbol{v}_{K,h}, q_{K,h})_{0,K}\\
&- \Big(\sum_{f\in\mathcal{F}_K} (\alpha_1 [p_h]_{K,f}, q_{K,h})_{0,f}\\
&\quad + (\alpha_3 \boldsymbol{n}_{K,f} \cdot [\boldsymbol{v}_h]_{K,f}, q_{K,h})_{0,f}\\
&\quad + (\alpha_4 [p_h]_{K,f}, \boldsymbol{n}_{K,f} \cdot \boldsymbol{w}_{K,h})_{0,f}\\
&\quad + (\alpha_2 \boldsymbol{n}_{K,f} \cdot [\boldsymbol{v}_h]_{K,f}, \boldsymbol{n}_{K,f} \cdot \boldsymbol{w}_{K,h})_{0,f} \Big)\,.
\end{aligned}$$

Then, the semi-discrete problem for the forward problem is

$$\boldsymbol{M}_h \partial_t \boldsymbol{u}_h(t) + (\boldsymbol{A}_h + \boldsymbol{M}_h \boldsymbol{D}_h)\boldsymbol{u}_h(t) = \boldsymbol{f}_h(t), \text{ for } t \in [0, T]\,.$$

Analogously, for the discretization of the adjoint operator $\boldsymbol{A}^*$, we have $\boldsymbol{A}_h^* = \sum_{K\in\mathcal{K}} \boldsymbol{A}_{K,h}^*$

$$\begin{aligned}
\big(\boldsymbol{A}_{K,h}^*(\boldsymbol{v}_h, \boldsymbol{p}_h), (\boldsymbol{w}_h, \boldsymbol{q}_h)\big)_{0,K} = &(\nabla p_{K,h}, \boldsymbol{w}_{K,h})_{0,K} + (\operatorname{div} \boldsymbol{v}_{K,h}, q_{K,h})_{0,K}\\
&- \Big(\sum_{f\in\mathcal{F}_K} (\alpha_1 [p_h]_{K,f}, q_{K,h})_{0,f}\\
&\quad - (\alpha_3 \boldsymbol{n}_{K,f} \cdot [\boldsymbol{v}_h]_{K,f}, q_{K,h})_{0,f}\\
&\quad - (\alpha_4 [p_h]_{K,f}, \boldsymbol{n}_{K,f} \cdot \boldsymbol{w}_{K,h})_{0,f}\\
&\quad + (\alpha_2 \boldsymbol{n}_{K,f} \cdot [\boldsymbol{v}_h]_{K,f}, \boldsymbol{n}_{K,f} \cdot \boldsymbol{w}_{K,h})_{0,f} \Big)\,,
\end{aligned} \tag{3.2}$$

for all $(\boldsymbol{v}_h, \boldsymbol{p}_h), (\boldsymbol{w}_h, \boldsymbol{q}_h) \in \boldsymbol{X}_h$. Note that $-(\boldsymbol{A}_h) \neq (-\boldsymbol{A})_h = \boldsymbol{A}_h^*$ since the signs of the flux terms are incompatible.

This semi-discretization of $\boldsymbol{A}$ and $\boldsymbol{A}^*$ is stable and consistent. Further, for sufficiently smooth functions it can be shown that the L^2 error of the semi-discretization is convergent of order $k + 1/2$ [HPS+15].

In general, operators with subscript h are representations of the operators on finite dimensional linear subspaces. Underlined objects are the representations with respect to the finite element basis. Let

$\{\boldsymbol{\varphi}_i\}_{i=1,\dots,\dim \boldsymbol{X}_h}$ be a basis of $\boldsymbol{X}_h$. Then, the entries of the matrices are given by

$$(\underline{\boldsymbol{A}}_h)_{i,j} = \left(\boldsymbol{A}_h\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right)_{0,\Omega}\,,\ (\underline{\boldsymbol{M}}_h)_{i,j} = \left(\boldsymbol{M}_h\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right)_{0,\Omega}\,, \qquad (3.3)$$
$$(\underline{\boldsymbol{D}}_h)_{i,j} = \left(\boldsymbol{D}_h\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right)_{0,\Omega}\,,\ (\underline{\boldsymbol{G}}_h)_{i,j} = \left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right)_{0,\Omega}\,.$$

A vector in $\boldsymbol{x}_h \in \boldsymbol{X}_h$ can be written as $\boldsymbol{x}_h = \sum_{i=1}^{\dim \boldsymbol{X}_h} \underline{\boldsymbol{x}}_h^i \boldsymbol{\varphi}_i$. Due to the nature of the dG space the matrices $\underline{\boldsymbol{M}}_h, \underline{\boldsymbol{D}}_h, \underline{\boldsymbol{G}}_h$ are block diagonal with block size $(k+1)^2$. The matrix $\underline{\boldsymbol{A}}_h$ is block-banded. The bandwidth is the number of neighbors per cell plus one. In our case this is always four, since we use structured meshes with quadrilaterals of same size.

3.2 Time Discretization

Let $N \in \mathbb{N}$ be the number of points in our time grid and set $t_n = n \cdot \triangle t$ for $n = 0, \dots, N$ with $\triangle t = T/N$. Define the fully discretized space of wave fields $\boldsymbol{Y}_h = \boldsymbol{X}_h^N$. In this thesis we will use two time discretization schemes. The first one is an exponential integrator which has been shown to be efficient for wave equations in conjunction with our dG discretization in [HPS+15]. A very brief introduction and using only formal calculations is given here. For a thorough introduction for a very wide scope of problems we refer to the review on exponential integrators by Hochbruck and Ostermann [HO10]. We use the exponential notation $\exp(P)$ for a general linear operator P freely. For formal explanations see [RR04]. The idea of an exponential integrator is that we can express (mild) solutions of an initial value problem of the form $\partial_t u + Pu = f,\ u(0) = u_0$ using the variation of constants formula

$$u(t) = \exp(-Pt)u_0 + \int_0^t \exp(-(t-\tau)P)f(\tau)\,\mathrm{d}\tau\,.$$

Evaluating this formula at discrete time steps yields

$$u(t_{n+1}) = \exp(-P\triangle t)u(t_n) + \int_{t_n}^{t_{n+1}} \exp(-(t_{n+1}-\tau)P)f(\tau)\,\mathrm{d}\tau$$
$$\overset{\triangle t = t_{n+1}-t_n}{=} \exp(-P\triangle t)(t_n) + \int_0^{\triangle t} \exp(-(\triangle t-\tau)P)f(\tau + t_n)\,\mathrm{d}\tau\,.$$

The integral is then typically approximated by interpolation of f and exact integration of the resulting integral. In this case we interpolate f at the midpoint of the interval yielding

$$\int_0^{\triangle t} \exp(-(\triangle t-\tau)P)f(\tau+t_n)\,\mathrm{d}\tau \approx \int_0^{\triangle t} \exp(-(\triangle t-\tau)P)f((t_n+t_{n+1})/2)\,\mathrm{d}\tau = p(-\triangle tP)\,f((t_n+t_{n+1})/2)\,,$$

where $p(t) = (\exp(t)-1)/t$ (this function is typically called φ_1 which we avoid due to clash of notation). Now, we apply this formula to our discrete setting. From the proof of Theorem 2 we know that the discrete operator in our case is $\underline{\boldsymbol{P}}_h = -\underline{\boldsymbol{M}}_h^{-1}(\underline{\boldsymbol{A}}_h + \underline{\boldsymbol{M}}_h\underline{\boldsymbol{D}}_h)$. Therefore, we get a fully discretized time stepping scheme

$$\boldsymbol{y}_{h,n+1} = \exp(-\triangle t\underline{\boldsymbol{P}}_h)\boldsymbol{y}_{h,n} + \triangle t\, p(-\underline{\boldsymbol{P}}_h\triangle t)\,\boldsymbol{f}_{h,n+1/2}\,.$$

Evaluation of the exponential and p is costly for large matrices. Therefore, we reduce it to only one evaluation using the identity $-\triangle t \cdot p(-\triangle tP)P = \exp(-\triangle tP) - I$, which, after some rearrangement, yields

$$\boldsymbol{y}_{h,n+1} = \boldsymbol{y}_{h,n} + \triangle t\, p(-\underline{\boldsymbol{P}}_h\triangle t)(\underline{\boldsymbol{P}}_h\boldsymbol{y}_{h,n} + \boldsymbol{f}_{h,n+1/2})\,. \qquad (3.4)$$

Algorithm 3.1 Fixed point iteration

```
Require: $\underline{C}$ % Matrix; $\underline{b}$ % right-hand side  $\underline{x}_0$ % starting value  $\varepsilon$ % relative tolerance
$\underline{r}_0 \leftarrow \underline{b} - \underline{C}\underline{x}_0$
$\underline{B} \leftarrow \mathrm{BJ}(\underline{C})$ % build Block-Jacobi Preconditioner
while $\|\underline{r}_k\| > \varepsilon\|\underline{r}_0\|$ do
    $\underline{x}_k \leftarrow \underline{x}_{k-1} + \underline{B}\underline{r}_{k-1}$
    $\underline{r}_k \leftarrow \underline{r}_{k-1} - \underline{C}\underline{x}_k$
end while
return $\underline{x}_k$
```

We have not discussed how to evaluate the function p. If we use an iterative Arnoldi process it is only necessary to evaluate p on a (comparatively) small matrix. The evaluation of p can be done by evaluating a matrix exponential of a slightly bigger matrix as discussed in [AMH11]. The matrix exponential in turn is calculated by a modified scaling and squaring approach as proposed in [AMH09]. Although the results in [HPS$^+$15] encourage the use of an implicit scheme, we found that in our case the explicit scheme is more efficient and shows good stability properties. Therefore, we evaluate the exponential by an explicit Arnoldi process which can also be found in more detail in [HPS$^+$15].

Later in this thesis the exponential integration rule will be inconvenient to use. As to why this is the case will be discussed there. Instead, we will use the trapezoidal rule which is an unconditionally stable implicit two stage Runge-Kutta method of order two. This yields a method of the form $\boldsymbol{y}_{h,n+1} = \boldsymbol{y}_{h,n} + \Delta\boldsymbol{y}$ with the linear system

$$\left(\underline{\boldsymbol{M}}_h - \frac{\Delta t}{2}(\underline{\boldsymbol{A}}_h + \underline{\boldsymbol{M}}_h\underline{\boldsymbol{D}}_h)\right)\Delta\boldsymbol{y} = \Delta t(\underline{\boldsymbol{A}}_h + \underline{\boldsymbol{M}}_h\underline{\boldsymbol{D}}_h)\boldsymbol{y}_{h,n} + \frac{1}{2}(\boldsymbol{f}_{h,n} + \boldsymbol{f}_{h,n+1})\,.$$

The arising linear system is preconditioned with a block-wise Jacobi method and solved with a fixed point iteration see Algorithm 3.1 which was faster than `GMRES` for the linear systems we have to solve.

3.3 The finite element library `M++`

The numerical calculations in this thesis are realized with the help of the open-source parallel finite element software `M++` [BW21]. This library is written in `C++` using OpenMPI (Open Message Passing Interface) [GFB$^+$04] to realize its distributed memory parallelization, that is, one calculation can be run on several physical machines (nodes) at the same time. Therefore, it benefits from modern large scale computing clusters which are made of many nodes (such as the *Hochleistungsrechner Karlsruhe* (HoreKa)) and makes calculations possible that would not fit on a single machine. For a more detailed description of `M++`, its development, and references of more research that uses `M++` see [Bau23]. To realize the features used in my own FWI project, I developed features in `M++` and helped to maintain the library as well. The FWI project is not part of the library but rather includes `M++` and uses its functionality. All code developed in the FWI software is programmed in such a way that the results are the same (to machine precision) even for a varying number of processor cores. The final version of the FWI code used in this thesis can be found at https://gitlab.kit.edu/kit/mpp/fwi/-/releases/Dissende. Additionally, all data and software needed to produce the figures and instructions to run the calculations can be found at https://doi.org/10.35097/1898.

3.4 Numerical Convergence

In [BFE+21] it was shown in two different studies that the space discretization from Section 3.1 with polynomial degree k yields convergence order of $k+1$. The first study compared numerical solutions with a known smooth solution and the other one in a heterogeneous material that is closer to the application and without known solution. Here, we conduct another study that lies in between these two. We want to qualitatively compare the numerical solution of (2.8) with compactly supported sources and homogeneous material parameters to solutions calculated with Green's functions. Some explanations regarding Green's function for the wave equation are given in the appendix Chapter A.

3.4.1 Source implementation

In our forward calculations the source terms are of the form $\boldsymbol{f}(t,\boldsymbol{x}) = (\boldsymbol{0}, f_{p_0}(t,\boldsymbol{x}), 0, \dots, 0)^\top$. For the p_0 component we have $f_{p_0}(t,\boldsymbol{x}) = \phi(t-t_0)\delta(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{src},0})$ with δ being the Dirac delta distribution and ϕ a smooth real function which is zero for negative times (otherwise it would be called 'acausal'). The delta distribution is of low regularity and therefore some caution has to be put into its implementation. Regularized delta distributions are discussed in [HNS16]. A method to increase order based on the ideas of Bramble and Schatz [BS77] involving a convolution can be found in [YS13]. We opt for a rather straight forward implementation, that is, for any test function $\boldsymbol{\varphi} \in \boldsymbol{X}_h$ we get

$$(\boldsymbol{f}(t), \boldsymbol{\varphi})_{0,\Omega} = \phi(t-t_0)\boldsymbol{\varphi}(\boldsymbol{x}_{\mathrm{src},0})\,.$$

The approximation is a Galerkin approximation of the delta distribution. We can find a representation of the delta distribution that is contained in the closure of a cell $\bar{K}$ with basis elements $\boldsymbol{\varphi}_j \in \boldsymbol{X}_h$. Then, $\delta_{h,K} = \sum_{j\in\mathcal{I}_K} \underline{\delta}_{h,j}\boldsymbol{\varphi}_j$ with $\mathcal{I}_K$ being the set of indices corresponding to the cell K. We enforce the defining property of the delta distribution by

$$\delta_{h,K}(\boldsymbol{\varphi}_{i,K}) = \sum_{j\in\mathcal{I}_K} \underline{\delta}_{h,j}\big(\boldsymbol{\varphi}_{j,K}, \boldsymbol{\varphi}_{i,K}\big)_{0,K} \overset{!}{=} \boldsymbol{\varphi}_{i,K}(\boldsymbol{x}_{\mathrm{src},0}),\ \forall i \in \mathcal{I}_K\,.$$

The coefficients of the approximation can be calculated by solving the linear system

$$\underline{G}\underline{d} = \underline{s},\ \text{with}\ \underline{G}_{i,j} = \big(\boldsymbol{\varphi}_{j,K}, \boldsymbol{\varphi}_{i,K}\big)_{0,K}\,,\ \underline{d}_i = \underline{\delta}_{h,i}\ \text{and}\ \underline{s}_i = \boldsymbol{\varphi}_i(\boldsymbol{x}_{\mathrm{src},0})\,,\ \forall i,j \in \mathcal{I}_K. \tag{3.5}$$

The Gramian matrix $\underline{G}$ is symmetric positive definite and therefore the linear system in (3.5) is uniquely solvable for any $\underline{s}$. By construction it holds

$$\big(\delta - \delta_{h,K}, \boldsymbol{\varphi}_{i,K}\big)_{0,K} = 0\ \forall i \in \mathcal{I}_K\,,$$

showing Galerkin orthogonality. Changing the source position also changes the coefficients of $\delta_{h,K}$, since we change the right-hand side of the linear system (3.5). Therefore, the approximations are not translations of each other, not even in the conforming case of the *Spectral Element Method* (SEM) as discussed the book by Fichtner [Fic11]. Also note that even if we choose $\boldsymbol{x}_{\mathrm{src},0}$ to be the nodal point corresponding to $\boldsymbol{\varphi}_{i,K}$ the approximation of the delta distribution is not a scaled version of $\boldsymbol{\varphi}_{i,K}$. This would only be the case for an orthogonal discrete basis where the nodal points coincide with the quadrature points like it is the case in SEM. Fichtner also comments in his book that this implementation yields a low pass filtered version of the delta distribution. Therefore, in the vicinity of the source the solution will not be correct. However, if the cells are small enough this low-pass filtering effect vanishes at the distance of the diameter of one cell in the case of SEM in [Fic11]. We want to investigate this behavior for our dG method in the vicinity of the source and consider the domain $\Omega = [-400\text{m}, 400\text{m}] \times [-400\text{m}, 400\text{m}]$ and choose $v_\text{p} = 3500\,{}^\text{m}/_\text{s}$, $\varrho = 2000\,{}^\text{kg}/_{\text{m}^3}$. Define the relative error

$$\text{err}_{rel,p}^{h,k}(\boldsymbol{x}) = \frac{|p(\boldsymbol{x}) - p_{h,k}(\boldsymbol{x})|}{\max_{x\in\Omega} |p(\boldsymbol{x}) - p_{h,k}(\boldsymbol{x})|}\,,$$

where p is the pressure of a solution obtained by a Green's function and $p_{h,k}$ the pressure of a solution of our dG solver with polynomial degree $k = 2, 3, 4$ and mesh width $h = 50\,\text{m}, 25\,\text{m}, 12.5\,\text{m}$ and no damping $(L = 0)$. We place the source at $(0, 0)$ and choose the mesh such that the source is in three different locations within a cell: One at the middle of a cell, one at very close vicinity of the corner of a cell and one precisely on the corner of a cell. In the last case the source has a support of four cells instead of one. In the plot of the error $\text{err}_{rel,p}^{h,k}$ in Fig. 3.1 we see that the behavior of the error in the vicinity is sensitive to the placement of the source within the cell. It is noticeable that placing the source in the middle of the cell produces the best results. It is surprising that polynomial degree $k = 3$ produces smaller errors than $k = 4$. From this small study we deduct that in our synthetic experiments we should always try to choose the position of the sources and receivers – since they also act as source in the adjoint problem – to be close to the cell's midpoint and try to avoid the borders and corners. Further, if it is feasible in terms of degrees of freedom one should choose $k = 3$. In real data sets we cannot choose our sources and receivers freely, they are given by the geometry in the real world. Therefore, it would be necessary to determine how much of a phase difference a change of the position to the cell's midpoint would introduce (that is, distance moved in relation to the propagation speed). Local refinement of the grid at the source and receiver locations could also be a way to reduce the phase difference since a smaller grid reduces the distance to the next midpoint of a cell. Even if we do not measure in the direct vicinity, these errors are of interest since large errors at the source and receiver locations can lead to artifacts in the reconstructions.

3.5 Absorbing boundary conditions

In our simulations we often want to avoid nonphysical reflections at the boundaries of the computational domain Ω. To this end, we extend Ω to Ω_E by a layer of length d. In that layer we keep the impedance $Z(\boldsymbol{x}) = \sqrt{(1 + L\tau_\text{p}(\boldsymbol{x}))/(1 + \alpha\tau_\text{p})(\boldsymbol{x})}\,\varrho(\boldsymbol{x})v_\text{p}(\boldsymbol{x})$ constant to reduce reflections. This reduction is very dependent on the angle of incidence. Waves with incidence of zero degrees produce no reflections (constant impedance gives a transmission coefficient of one in the 1D wave equation). Higher values produce growing reflections. Further, we decrease the phase velocity gradually so that a wave entering this domain stays there for a longer time. In the case of $L > 0$, we additionally increase the value of Q exponentially, damping the amplitude of the wave. To further decrease unwanted reflections at the beginning of the damping layer we make the transition from Ω to Ω_E smooth with a cosine to a linear decrease. Let Q_i, Q_b be the quality factors inside the computational domain and at the boundary of the artificial layer, respectively. Further, $v_{\text{p},i}, v_{\text{p},b}, v_{\text{p},t}$ are the velocity values inside the domain, at the boundary of the artificial layer and at the transition from the cosine to the linear decrease of velocity. Let $r(y) = \min_{x\in\partial\Omega} \max\{|y_1 - x_1|, |y_2 - x_2|\}$ be the distance of a point $y \in \Omega_E \setminus \Omega$ to the domain Ω.

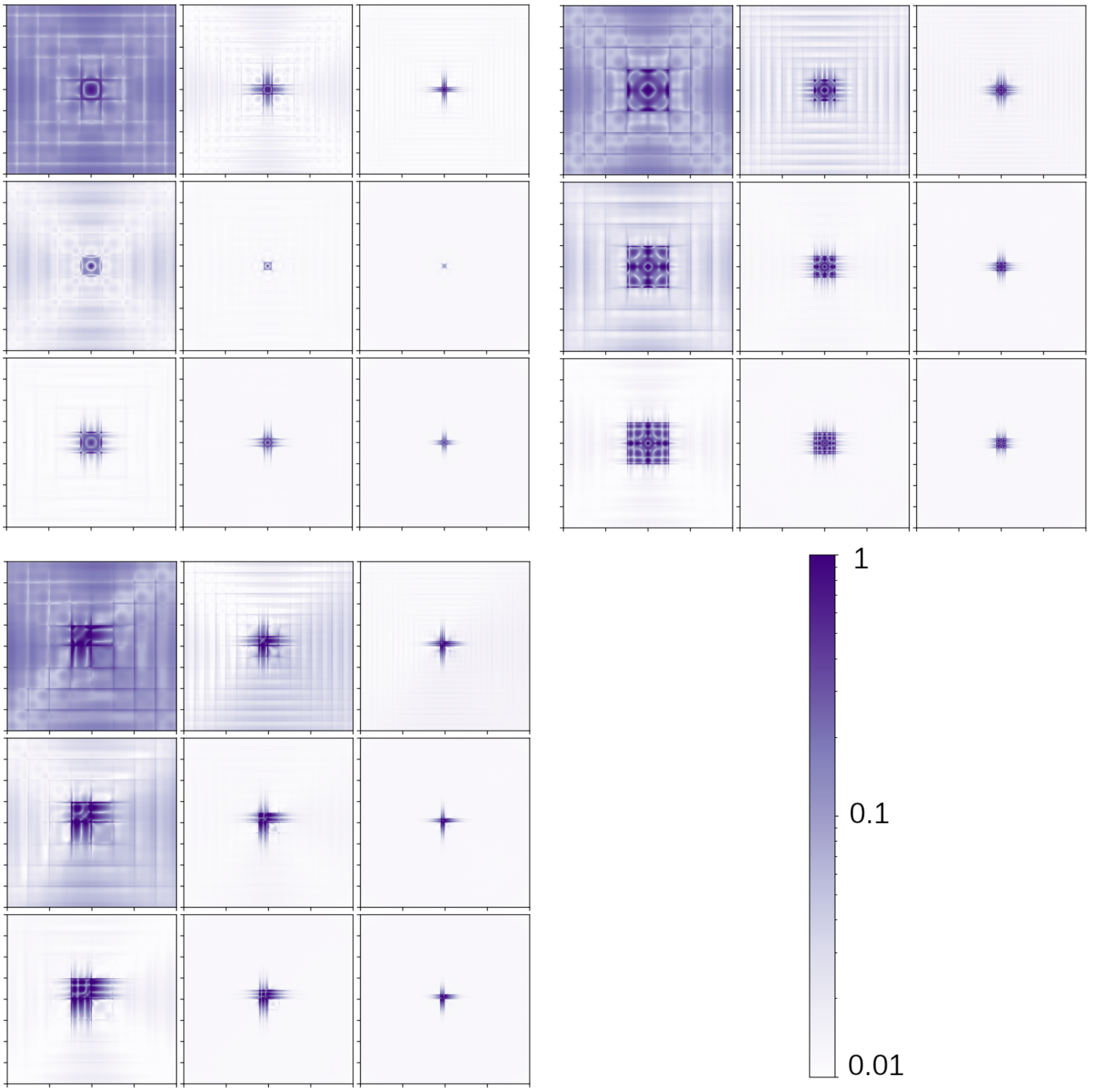


Figure 3.1: Error $\mathrm{err}_{rel,p}^{h,k} = \frac{|p(\boldsymbol{x})-p_{h,k}(\boldsymbol{x})|}{\max_{x\in\Omega}|p(\boldsymbol{x})-p_{h,k}(\boldsymbol{x})|}$ with $p_{h,k}$ being the pressure component of the solution of the dG method for mesh width h and polynomial degree k and a point source in space. For better visibility the error is plotted on a logarithmic scale. Top Left: source is in the middle of the cell. Top Right: source is at the corner of a cell. Bottom left: source is in very close vicinity of a corner. In all plots we have h refinement from left to right and refinement of the polynomial degree k from top to bottom.

Then, for a point $y \in \Omega_E \setminus \Omega$ the values of the parameters are given by

$$Q_l(r(y)) = Q_i \left(\frac{Q_b}{Q_i}\right)^{r(y)/d}, \quad v_{\mathrm{p},l}(r(y)) = \begin{cases} (v_{\mathrm{p},i} - v_{\mathrm{p},t}) \cos\left(\frac{x}{s_t}\frac{\pi}{2}\right) + v_{\mathrm{p},t} & \text{for } r(y) \in [0, s_t], \\ (v_{\mathrm{p},t} - v_{\mathrm{p},b}) \frac{x - s_l}{r - s_l} + v_{\mathrm{p},t} & \text{for } r(y) \in [s_t, d]. \end{cases}$$

We use the values $Q_b = \min\{1.1, Q_{\min}\}, v_{\mathrm{p},b} = 0.2 v_{\mathrm{p},i}, v_{\mathrm{p},t} = 0.85 v_{\mathrm{p},i}, s_t = 0.3d$ which are proposed in [Boh98]. The value $Q_{\min} = \lim_{\tau_{\mathrm{p}} \to \infty} Q(\omega_0) \overset{(2.6)}{=} \alpha_1(\omega_0)/\alpha_2(\omega_2)$ is the minimal value for the quality factor for fixed τ_l's and ω_0. In Fig. 3.2 we can see the values of Q and v as a function of distance. The extension and choice of values yield good absorbing qualities in a very natural way within the model. The size of the layer d is adapted to be between 1-2 wavelengths of the frequency ω_0. A typical setting for FWI is shown in Fig. 3.2: An extension of the domain on three sides where the remaining side is the surface of the earth.

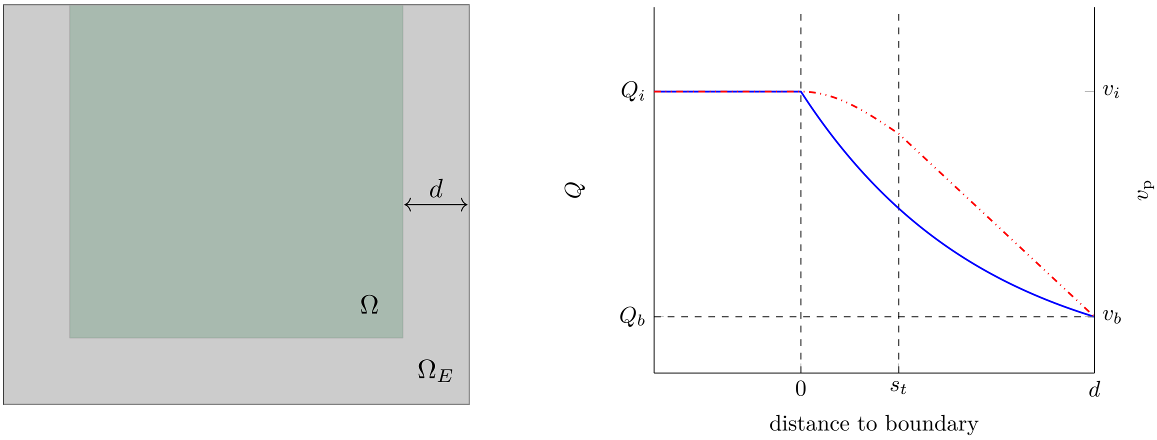


Figure 3.2: Left: example of extended domain, with reflecting bounds at the top. Right: values of Q (blue, line) and v_{p} (red,dash dotted).

CHAPTER

FOUR

INVERSION WITH THE REDUCED FORMULATION

Like we discussed in the the introduction of this thesis, the aim of FWI is to determine material properties beneath the earth's surface using measurements acquired on the surface. To achieve the this aim, we want to determine a wave field $\boldsymbol{u}$ and material parameter μ for a given source term $\boldsymbol{f}$ and measured data $\boldsymbol{s}_{\mathrm{obs}}$ that accomplish two goals at the same time:

1. The *model error* $\|\boldsymbol{\mathfrak{L}}(\mu)\boldsymbol{u} - \boldsymbol{f}\|_{0,[0,T]\times\Omega} \approx 0$.
2. The *data error* $\|\Psi\boldsymbol{u} - \boldsymbol{s}_{\mathrm{obs}}\|_{\boldsymbol{S}} \approx 0$.

Although the meaning of ≈ can vary depending on the specific problem, reducing these quantities to some extent is always part of an inversion. The two requirements guarantee that $\boldsymbol{u}$ is physically meaningful and consistent with the material *as well as* accounting for the data. In this chapter, we introduce one approach to achieve this objective utilizing a suitable mathematical formulation. Later, in Chapter 5, we present another one.

4.1 Reduced formulation

We define an operator Φ that maps a material distribution to the corresponding measurement and call it *parameter-to-seismogram map*. Let $\boldsymbol{s}_{\mathrm{obs}}$ be the measured data that usually is contaminated with noise, $\boldsymbol{P}_{\mathrm{adm}}$ the set of admissible parameters from (2.12), and $\boldsymbol{S}^{\Xi}$ the seismogram space from (2.24) containing all Ξ shots. Then one way to formulate the parameter identification problem is

$$\text{find} \quad \mu \in \boldsymbol{P}_{\mathrm{adm}} \quad \text{such that} \quad \Phi(\mu) = \boldsymbol{s}_{\mathrm{obs}} \in \boldsymbol{S}^{\Xi}\,. \tag{RED}$$

In this chapter we will define the parameter-to-seismogram map and its Fréchet derivative and adjoint thereof in a precise mathematical manner. We will present some analysis, and briefly discuss the ill-posedness of the problem. We call the formulation in this chapter (that is, (RED)) the *reduced* formulation of FWI.

4.1.1 Theory of the reduced operator

We start of with the definition of the mapping from above.

Definition 3. *Let $\boldsymbol{u}_0 = 0$ and $\boldsymbol{f}_\xi \in W^{1,1}([0,T],\boldsymbol{X}) \cap C^1([0,T],\boldsymbol{X})$, $\xi = 0,\dots,\Xi-1$. The parameter-to-seismogram map is defined by*

$$\Phi\colon \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \boldsymbol{S}^{\Xi},\ \mu \longmapsto (\Psi\boldsymbol{u}^0,\dots,\Psi\boldsymbol{u}^{\Xi-1})^\top \mathit{with}\ \boldsymbol{u}_\xi = \mathcal{F}_\xi(\mu)\,,$$

with the parameter-to-wave field map

$$\mathcal{F}_\xi\colon \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \boldsymbol{Y},\ \mu \longmapsto \boldsymbol{u}_\xi,\ \boldsymbol{u}\ \mathit{fulfills}\ \boldsymbol{\mathfrak{L}}(\mu)\boldsymbol{u}_\xi(t) = \boldsymbol{f}_\xi(t),\ t \in [0,T]\ \mathit{and}\ \boldsymbol{u}(0) = \boldsymbol{0}\,,$$

with $\boldsymbol{\mathfrak{L}}$ being the differential operator of the visco − acoustic wave equation from Theorem 2.

Remark 3. *With Definition 3, we set the model error to zero by construction and therefore we only have to address the data error in the inversion.*

To avoid unnecessary notation, most of the analysis is done on $\mathcal{F} = \mathcal{F}_0$ for $\Xi = 1$. The results carry over to $\Xi \in \mathbb{N}$. Further, they carry over to Φ since it is only a composition with a linear operator. The maps $\mathcal{F}, \Phi$ are well-defined due to the existence results from Section 2.2. It can also be shown that these maps are also (locally) Lipschitz continuous for reasonable assumptions on $\boldsymbol{f}$ and $\boldsymbol{u}_0$ cf. [KR16]. Now, we can formulate the problem (RED) in a mathematically well-defined way. The motivation to investigate analytical properties of the problem (RED) is that it is notoriously hard to solve. It can even be proven that it is locally ill-posed. First, we want to recall that definition.

Definition 4. *Let $K\colon U \longrightarrow V$ be an operator between Banach spaces U and V. Then, the equation*

$$K(x) = y$$

is locally ill-posed *at $\hat{x} \in \mathcal{D}(K)$, satisfying $K(\hat{x}) = y$, if in any neighborhood of $\hat{x}$ there exists a sequence $\{x_k\}_{k\in\mathbb{N}} \subset \mathcal{D}(K)$ such that*

$$\lim_{k\to\infty} \|K(x_k) - K(\hat{x})\|_V = 0,\ \mathit{however}\ \|x_k - \hat{x}\|_U \not\to 0\ \mathit{for}\ k \to \infty\,.$$

In this work we mainly talk about nonlinear operators. Outside of theorems and proofs we will mostly use the term *ill-posedness* (which is a mathematical term relevant in the theory of linear operators) if we actually mean local ill-posedness. Now, we want to show that we can always construct a sequence showing ill-posedness of the parameter-to-solution map for any interior point of the admissible parameters.

Theorem 3. *Let $\mathcal{F}\colon \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \boldsymbol{Y}$ be the parameter-to-wave field operator from Definition 3. Then, the equation*

$$\mathcal{F}(\mu) = \boldsymbol{u}$$

is locally ill-posed for any interior point $\mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})^\top$ of $\boldsymbol{P}_{\mathrm{adm}}$.

Proof. To prove the ill-posedness, we follow the steps from [KR19, Theorem 4.3] and apply the arguments shown there to our notation. First, let $\mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})^\top \in \operatorname{int}(\boldsymbol{P}_{\mathrm{adm}})$. Choose a fixed $\hat{\boldsymbol{z}} \in \operatorname{int}(\Omega)$ and $\delta > 0$ such that $B_\delta(\hat{\boldsymbol{z}}) \subset \Omega$. Then, let r_1, r_2, r_3 be small enough such that

$$\mu_k := \begin{pmatrix} \varrho + r_1\chi_k \\ v_{\mathrm{p}} + r_2\chi_k \\ \tau_{\mathrm{p}} + r_3\chi_k \end{pmatrix} \in \boldsymbol{P}_{\mathrm{adm}}\,,\ \text{with}\ \chi_k\colon \Omega \longrightarrow \mathbb{R}\,,\ \boldsymbol{z} \longmapsto \begin{cases} 1 & \text{if } |\hat{\boldsymbol{z}} - \boldsymbol{z}| \le \frac{\delta}{k}\,, \\ 0 & \text{else}\,. \end{cases}$$

Now, it is clear that $\|\mu - \mu_k\|_{\infty,\Omega^3} = \max\{r_1, r_2, r_3\}$, for all $k \in \mathbb{N}$. Therefore, this sequence does not converge. Now, let $\boldsymbol{u}_k = \mathcal{F}(\mu_k)$ and $\boldsymbol{u} = \mathcal{F}(\mu)$. Then, $\boldsymbol{u}_{\Delta,k} := \boldsymbol{u}_k - \boldsymbol{u}$, satisfies the initial value problem

$$\mathfrak{L}(\mu_k)\boldsymbol{u}_{\Delta,k}(t) = (\boldsymbol{M}(\mu) - \boldsymbol{M}(\mu_k))(\partial_t \boldsymbol{u}(t) + \boldsymbol{D}\boldsymbol{u}(t)), \qquad \boldsymbol{u}_{\Delta,k}(0) = \boldsymbol{0}.$$

From [KR19] page 3, we know that the solution of our differential equation has a continuous dependency on the right-hand side. For $\boldsymbol{u}_{\Delta,k}$ this means that there is $C > 0$ (independent of k) such that

$$\|\boldsymbol{u}_{\Delta,k}\|_{0,[0,T]\times\Omega} \leq C\|(\boldsymbol{M}(\mu) - \boldsymbol{M}(\mu_k))(\partial_t \boldsymbol{u}(t) + \boldsymbol{D}\boldsymbol{u}(t))\|_{0,[0,T]\times\Omega}. \tag{4.1}$$

Now, we show that $E_k := \boldsymbol{M}(\mu_k) - \boldsymbol{M}(\mu)$ fulfills $\lim_{k\to\infty} E_k \boldsymbol{w} = 0$ for all $w \in \boldsymbol{Y}$. The material parameters μ, μ_k only differ inside the support of χ_k. Therefore, $E_k\boldsymbol{w}$ is also supported on the support of χ_k. It is clear that it is bounded, since it is a component-wise multiplication operator with only positive factors. The scalar function $\nu\colon (r_{\min}, r_{\max}) \times (v_{\min}, v_{\max}) \times (\tau_{\min}, \tau_{\max}) \longrightarrow \mathbb{R}\,,\ \nu(\tilde{r}, v, \tau) = (1+\alpha\tau)/(\tilde{r}v^2)$ represents the factor in front of $\partial_t p_0$ in the wave equation (2.8). The factor $\tilde{\nu}(\tilde{r}, v, \tau) = \nu(\tilde{r}, v, \tau)/\tau$ corresponds to $\partial_t p_l\,,\ l = 1\ldots, L$. With these definitions we can write

$$E_k\boldsymbol{w}|_{\operatorname{supp}\chi_k} = \begin{pmatrix} r_1\boldsymbol{w}_v \\ (\nu(\varrho + r_1, v_{\mathrm{p}} + r_2, \tau_{\mathrm{p}} + r_3) - \nu(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))\boldsymbol{w}_{p_0} \\ (\tilde{\nu}(\varrho + r_1, v_{\mathrm{p}} + r_2, \tau_{\mathrm{p}} + r_3) - \tilde{\nu}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))\boldsymbol{w}_{p_1} \\ \vdots \\ (\tilde{\nu}(\varrho + r_1, v_{\mathrm{p}} + r_2, \tau_{\mathrm{p}} + r_3) - \tilde{\nu}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))\boldsymbol{w}_{p_L} \end{pmatrix}$$

where $\nu(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})$ denotes the point-wise composition of the functions. Now, we apply the the mean value theorem on $\nu, \tilde{\nu}$:

$$\nabla\nu(\tilde{r}, v, \tau) = \begin{pmatrix} -\frac{1+\alpha\tau}{\tilde{r}^2v^2} \\ -2\frac{1+\alpha\tau}{\tilde{r}v^3} \\ \frac{\alpha}{\tilde{r}v^2} \end{pmatrix}, \qquad \nabla\tilde{\nu}(\tilde{r}, v, \tau) = \begin{pmatrix} -\frac{1+\alpha\tau}{\tilde{r}^2v^2\tau} \\ -2\frac{1+\alpha\tau}{\tilde{r}v^3\tau} \\ \frac{1}{\tilde{r}v^2\tau^2} \end{pmatrix}$$

therefore we can (note that $\alpha > 0$)

$$|\nabla\nu(\tilde{r}, v, \tau) \cdot (r_1, r_2, r_3)^\top| \leq \underbrace{\left|\frac{1+\alpha\tau_{\max}}{r_{\min}^2 v_{\min}^2}\right| r_1 + \left|2\frac{1+\alpha\tau_{\max}}{r_{\min}v_{\min}^3}\right| r_2 + \left|\frac{\alpha}{r_{\min}v_{\min}^2}\right| r_3}_{=:b_1(r_1,r_2,r_3)}$$

and

$$|\nabla\tilde{\nu}(\tilde{r}, v, \tau) \cdot (r_1, r_2, r_3)^\top| \leq \underbrace{\left|\frac{1+\alpha\tau_{\max}}{r_{\min}^2 v_{\min}^2 \tau_{\min}}\right| r_1 + \left|2\frac{1+\alpha\tau_{\max}}{r_{\min}v_{\min}^3\tau_{\min}}\right| r_2 + \left|\frac{1}{r_{\min}v_{\min}^2\tau_{\min}^2}\right| r_3}_{=:b_2(r_1,r_2,r_3)}.$$

Using these inequalities yields

$$\begin{aligned} \|E_k\boldsymbol{w}\|_{0,\Omega}^2 &= \|E_k\boldsymbol{w}\|_{0,\operatorname{supp}\chi_k}^2 \\ &\leq r_1^2\|\boldsymbol{w_v}\|_{0,\operatorname{supp}\chi_k}^2 + b_1(r_1, r_2, r_3)^2\|\boldsymbol{w}_{p_0}\|_{0,\operatorname{supp}\chi_k}^2 \\ &\qquad + b_2(r_1, r_2, r_3)^2 \sum_{l=1}^{L} \|\boldsymbol{w}_{p_l}\|_{0,\operatorname{supp}\chi_k}^2 . \end{aligned}$$

Now, fix $r < 1$ and choose $r_1, r_2, r_3 > 0$, such that $\max\{r_1^2, b_1(r_1, r_2, r_3)^2, b_2(r_1, r_2, r_3)^2/L\} < r^2$. This choice is possible since b_1, b_2 are multilinear (therefore continuous and $b_1(0, 0, 0) = b_2(0, 0, 0) = 0$) functions. For any $\boldsymbol{w} \in \boldsymbol{Y}$ this yields

$$\|E_k\boldsymbol{w}\|_{0,\Omega}^2 \leq r^2\|\boldsymbol{w}\|_{0,\operatorname{supp}\chi_k}^2 .$$

The measure of the support of χ_k vanishes as k tends to infinity which implies $\|\boldsymbol{w}\|^2_{0,\mathrm{supp}\chi_k}$. Hence, we get $\lim_{k\to\infty} E_k\boldsymbol{w} = 0$. It further shows that the norm of the sequence is bounded:

$$\|E_k\|^2 \le r^2.$$

Due to the boundedness of the operators, we can use dominated convergence:

$$\begin{aligned}\lim_{k\to\infty} \|E_k(\partial_t\boldsymbol{u} + \boldsymbol{D}\boldsymbol{u})\|_{0,[0,T]\times\Omega} &= \lim_{k\to\infty}\int_0^T \|E_k(\partial_t\boldsymbol{u} + \boldsymbol{D}\boldsymbol{u})\|_{0,\Omega}\\ &= \int_0^T \lim_{k\to\infty} \|E_k(\partial_t\boldsymbol{u} + \boldsymbol{D}\boldsymbol{u})\|_{0,\Omega}\\ &= 0\,.\end{aligned}$$

Therefore by (4.1) we get $\lim_{k\to\infty}\|\boldsymbol{u}_{\Delta,k}\|_{0,[0,T]\times\Omega}$ =0. This shows $\lim_{k\to\infty}\|\mathcal{F}(\mu_k) - \mathcal{F}(\mu)\|_{0,[0,T]\times\Omega} = 0$, however by construction $\|\mu_k - \mu\|_{0,\Omega} \not\longrightarrow 0$ for $k \longrightarrow \infty$ and thus concluding the proof. □

In the forward equations (2.4) and (2.8) we have seen that we can choose different sets of parameters to describe the material laws. Note that the proof of Theorem 3 only utilizes that the mapping of the parameters to the factors in the equation has to be smooth and with bounded derivatives. Therefore, for any choice of physical parameters that have this property we could easily repeat the arguments from Theorem 3. From the proof we can see that the ill-posedness is directly connected to the topology of the domain and the range of $\mathcal{F}$. While the L^∞-norm is not affected by shrinking the support of indicator functions, it does have this effect on the L^2-norm. This raises the question whether or not one should look at other spaces than L^∞ for the parameters. But looking at the results of the existence theory in Section 2.2 we see that L^∞ is the proper space, since we can ensure existence and uniqueness of the solutions. Also, the result in Theorem 3 implies the ill-posedness of (RED), since the measurement operator Ψ is a bounded linear operator.
We want to solve the inverse problem (RED) with a Newton-type method. Therefore, we need the Fréchet derivative of $\mathcal{F}$ and its adjoint. It is more convenient to analyze this on an operator basis instead of the parameters themselves. The explicit formulas for a given parameterization then follow with the chain rule.

Definition 5. *Let $\mathcal{S}(\boldsymbol{X}) = \{P \in \mathcal{L}(\boldsymbol{X})\colon P^* = P\}$ and $\mathcal{B} = \{B \in \mathcal{S}(\boldsymbol{X})\colon \beta_-\|x\|^2_{\boldsymbol{X}} \le (Bx,x)_{0,\Omega} \le \beta_+\|x\|^2_{\boldsymbol{X}}\}$ for given $0 < \beta_- < \beta_+ < \infty$. Further, assume $\boldsymbol{f} \in C^\infty([0,T],\boldsymbol{X})$ with compact support, bounded derivatives, and $\boldsymbol{f}^{(i)}(0) = \boldsymbol{0}$ $\forall i \in \mathbb{N}$. Define the* operator-to-solution *map*

$$F\colon \mathcal{B} \subset \mathcal{S}(\boldsymbol{X}) \longrightarrow \boldsymbol{Y},\ \boldsymbol{\Pi} \longmapsto \boldsymbol{u}\,,$$

where $\boldsymbol{u}$ solves $\boldsymbol{\Pi}\partial_t\boldsymbol{u}(t) + (\boldsymbol{A} + \boldsymbol{\Pi}\boldsymbol{D})\boldsymbol{u}(t) = \boldsymbol{f}(t),\ \forall t \in [0,T]$ and $\boldsymbol{u}(0) = \boldsymbol{0}$.

From [KR19, Theorem 3.2] we know that this map is well-defined. Together with the regularity assumptions on $\boldsymbol{f}$ the solutions are smooth in time.

Theorem 4. *The map from Definition 5 is Fréchet differentiable at any interior point $\boldsymbol{\Pi} \in \mathcal{B}$ with*

$$F'(\boldsymbol{\Pi})[\boldsymbol{H}] = \overline{\boldsymbol{u}},$$

where $\overline{\boldsymbol{u}}$ solves

$$\boldsymbol{\Pi}\partial_t\overline{\boldsymbol{u}}(t) + (\boldsymbol{A} + \boldsymbol{\Pi}\boldsymbol{D})\overline{\boldsymbol{u}}(t) = -\boldsymbol{H}(\partial_t F(\boldsymbol{\Pi})(t) + \boldsymbol{D}F(\boldsymbol{\Pi})(t)),\ \forall t \in [0,T] \tag{4.2}$$

and $\overline{\boldsymbol{u}}(0) = \mathbf{0}$.

Proof. See [KR19, Theorem 3.2], the theorem there is for weaker assumptions on $\boldsymbol{f}$ and only yields that $\overline{\boldsymbol{u}}$ is a mild solution (see Definition 2). The additional assumption $\boldsymbol{f} \in C^\infty([0,T],\mathbb{R})$ with compact support and bounded derivatives with $\boldsymbol{f}^{(i)}(0) = \mathbf{0}\ \forall i \in \mathbb{N}$, results in higher of $F(\boldsymbol{\Pi})$. This implies that the right-hand side of (4.2) possesses sufficient regularity for Fréchet derivative to be a classical solution. □

We cannot apply Theorem 4 directly to our problem as it involves operators and not parameters in our sense. Therefore, we need to apply the theory to the map $\boldsymbol{M}\colon \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \mathcal{L}(\boldsymbol{X}),\ \mu \longmapsto \boldsymbol{M}(\mu)$.

Lemma 1. *Let $\boldsymbol{P} = L^\infty(\Omega)^3$ be the space of all parameters, then*

$$\boldsymbol{M}\colon \boldsymbol{P} \supset \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \mathcal{L}(\boldsymbol{X})$$

is Fréchet differentiable for $\mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}) \in \operatorname{int}(\boldsymbol{P}_{\mathrm{adm}})$ and $\eta = (\eta_\varrho, \eta_{v_{\mathrm{p}}}, \eta_{\tau_{\mathrm{p}}}) \in \boldsymbol{P}$ with

$$\boldsymbol{M}'(\mu)[\eta] = \operatorname{diag}\Bigg(\eta_\varrho, \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2}\left(-\frac{1}{\varrho}\eta_\varrho - \frac{2}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} + \frac{\alpha}{1+\alpha\tau_{\mathrm{p}}}\eta_{\tau_{\mathrm{p}}}\right), \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2\tau_{\mathrm{p}}}\left(-\frac{1}{\varrho}\eta_\varrho - \frac{2}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} - \frac{1}{(1+\alpha\tau_{\mathrm{p}})\tau_{\mathrm{p}}}\eta_{\tau_{\mathrm{p}}}\right), \\ \ldots, \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2\tau_{\mathrm{p}}}\left(-\frac{1}{\varrho}\eta_\varrho - \frac{2}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} - \frac{1}{(1+\alpha\tau_{\mathrm{p}})\tau_{\mathrm{p}}}\eta_{\tau_{\mathrm{p}}}\right)\Bigg).$$

Proof. Recall the definitions of $\nu, \tilde{\nu}\colon (r_{\min}, r_{\max}) \times (v_{\min}, v_{\max}) \times (\tau_{\min}, \tau_{\max}) \longrightarrow \mathbb{R}$, $\nu(r,v,\tau) = (1+\alpha\tau)/(rv^2)$, $\tilde{\nu}(r,v,\tau) = \nu(r,v,\tau)/\tau$ from the proof of Theorem 3. Then the formula from above can also be written as

$$\boldsymbol{M}'(\mu)[\eta] = \operatorname{diag}\left(\eta_\varrho, \nabla\nu(\mu)\cdot\eta, \nabla\tilde{\nu}(\mu)\cdot\eta, \ldots, \nabla\tilde{\nu}(\mu)\cdot\eta\right).$$

To prove this formula we consider a more general setting. Let g_i for $i \in 1,\ldots,D \in \mathbb{N}$ be a smooth bounded function $\mathbb{R}^P \longrightarrow \mathbb{R}$ with bounded derivatives. Consider the point-wise composition

$$G\colon L^\infty(\Omega,\mathbb{R}^P) \longrightarrow \mathcal{L}(\boldsymbol{X}), m \longmapsto \operatorname{diag}(g_1(m), g_2(m), \ldots, g_D(m)).$$

Then, for $m \in L^\infty(\mathbb{R}^P,\mathbb{R})$ it holds that

$$G'(m)\colon L^\infty(\Omega,\mathbb{R}^P) \longrightarrow \mathcal{L}(\boldsymbol{X}), d \longmapsto \operatorname{diag}(\nabla g_1(m)\cdot d, \nabla g_2(m)\cdot d, \ldots, \nabla g_D(m)\cdot d).$$

Further, it holds

$$(G(m+d) - G(m) - G'(m)d)_i = g_i(m+d) - g_i(m) - \nabla g_i(m)\cdot d,$$

where i denotes the it-h component. Since g is smooth, we can use Taylor's expansion and obtain

$$\begin{aligned}\|g_i(m+d) - g_i(m) - \nabla g_i(m)\cdot d\|_\infty &= \operatorname{ess\,sup}_{x\in\Omega}|g_i(m(x)+d(x)) - g_i(m(x)) - \nabla g_i(m(x))\cdot d(x)| \\ &\le C_i \cdot \operatorname{ess\,sup}_{x\in\Omega}\|d(x)\|_\infty^2 \\ &\le C_i \cdot \|d\|_\infty^2.\end{aligned}$$

Defining $\widetilde{C} := \max\{C_1,\ldots,C_D\}$ yields

$$\|G(m+d) - G(m) - G'(m)d\|_\infty \le \widetilde{C}\|d\|_\infty^2,$$

showing that G is Fréchet differentiable. The operator $\boldsymbol{M}$ can be defined by choosing $P = L + 2$ and $g_1(r, v, \tau) = r,\ g_2(r, v, \tau) = \nu(r, v, \tau) = g_3(r, v, \tau) = \ldots, g_{L+2}(r, v, \tau) = \tilde{\nu}(r, v, \tau)$, which yields the assertion. □

With this representation in place we can apply the theory from above to our setting.

Theorem 5. *The parameter-to-wave field map is Fréchet differentiable for any $\mu \in \operatorname{int}(\boldsymbol{P}_{\mathrm{adm}})$ with*

$$\mathcal{F}'(\mu)\colon \boldsymbol{P} \longmapsto \boldsymbol{Y},\ \mathcal{F}'(\mu)[\eta] = \overline{\boldsymbol{u}}\,,$$

where $\overline{\boldsymbol{u}}$ solves

$$\boldsymbol{M}(\mu)\partial_t\overline{\boldsymbol{u}}(t) + (\boldsymbol{A} + \boldsymbol{M}(\mu)\boldsymbol{D})\overline{\boldsymbol{u}}(t) = \boldsymbol{\mathfrak{L}}'_{\mu,\mathcal{F}(\mu)}[\eta](t),\ t \in [0, T], \quad \overline{\boldsymbol{u}}(0) = \boldsymbol{0}\,. \tag{4.3}$$

With

$$\boldsymbol{\mathfrak{L}}'_{\mu,\boldsymbol{u}}\colon \boldsymbol{P} \longrightarrow \boldsymbol{Y},\ \boldsymbol{\mathfrak{L}}'_{\mu,\boldsymbol{u}}[\eta] = -\boldsymbol{M}'(\mu)[\eta](\partial_t\boldsymbol{u} + \boldsymbol{D}\boldsymbol{u})$$

we denote the operator that linearly maps the direction of the derivative to the right-hand side of (4.3) for $\boldsymbol{u} \in C^1([0, T], \boldsymbol{X})$.

Proof. In the notation of Theorem 4, the chain rule for the Fréchet derivative gives us $(F(\boldsymbol{M}(\mu)))'[\eta] = F'(\boldsymbol{M}(\mu))[\boldsymbol{M}'(\mu)[\eta]]$. □

Application of Theorem 4 yields the same results for the parameter-to-seismogram map.

Corollary 1. *The parameter-to-seismogram map defined in Definition 3 is also Fréchet differentiable for any $\mu \in \operatorname{int}(\boldsymbol{P}_{\mathrm{adm}})$ and $\eta \in \boldsymbol{P}$ with*

$$\Phi'(\mu)[\eta] = (\Psi\mathcal{F}'_0(\mu)[\eta], \ldots, \Psi\mathcal{F}'_{\Xi-1}(\mu)[\eta])^\top \in \boldsymbol{S}^\Xi\,.$$

Proof. This follows directly from the fact that Ψ is a bounded linear operator and the result of Theorem 4. □

From Theorem 5, we can conclude that to compute the Fréchet derivative, the same equation as in the forward problem must be solved. The right-hand side depends on the evaluation of the parameter-to-wave field at μ. As mentioned earlier, we want to evaluate the adjoint of the Fréchet derivative. The adjoint of the Fréchet derivative can be expressed using $\boldsymbol{\mathfrak{L}}'^*_{\mu,\boldsymbol{u}}\colon \boldsymbol{Y} \longrightarrow \boldsymbol{P}'$. In the next lemma, we will first compute the adjoint of an auxiliary operator in order to achieve this goal. To compress the notation but not lose sight of the dependencies, going further, we will put “inactive” components as subscript like we did in the definition of $\boldsymbol{\mathfrak{L}}'_{\mu,\boldsymbol{u}}$. This also helps to emphasize in what sense we understand it as linear operator and in what sense we want to calculate the adjoint.

Lemma 2. *Let $\boldsymbol{u} \in \boldsymbol{Y}$ and $\mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}) \in \boldsymbol{P}_{\mathrm{adm}}$. Then, define the linear operator*

$$\boldsymbol{M}'_{\mu,\boldsymbol{u}}\colon \boldsymbol{P} \longrightarrow \boldsymbol{Y},\ \eta \longmapsto \boldsymbol{M}'(\mu)[\eta]\boldsymbol{u}\,.$$

Its adjoint has the representation

$$\boldsymbol{M}'^*_{\mu,\boldsymbol{u}}\colon \boldsymbol{Y} \longrightarrow L^1(\Omega)^3 \subset \boldsymbol{P}'\,,\ \boldsymbol{M}'^*_{\mu,\boldsymbol{u}}\boldsymbol{y} = \begin{pmatrix} R - \frac{\nu(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})}{\varrho}T_0 - \frac{\tilde{\nu}(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})}{\varrho}\sum_{l=1}^L T_l \\ -\frac{2\nu(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})}{v_{\mathrm{p}}}T_0 - \frac{2\tilde{\nu}(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})}{v_{\mathrm{p}}}\sum_{l=1}^L T_l \\ \frac{\nu(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})\alpha}{1+\alpha\tau_{\mathrm{p}}}T_0 - \frac{\tilde{\nu}(\varrho,v_{\mathrm{p}},\tau_{\mathrm{p}})}{(1+\alpha\tau_{\mathrm{p}})\tau_{\mathrm{p}}}\sum_{l=1}^L T_l \end{pmatrix}$$

with $\boldsymbol{y} = (\boldsymbol{y_v}, \boldsymbol{y}_{p_0}, \boldsymbol{y}_{p_1}, \dots \boldsymbol{y}_{p_L})$ *and*

$$R(\boldsymbol{x}) = (\boldsymbol{y_v}(\cdot,\boldsymbol{x}), \boldsymbol{u_v}(\cdot,\boldsymbol{x}))_{0,[0,T]}\,,\ T_i(\boldsymbol{x}) = \big(\boldsymbol{y}_{p_i}(\cdot,\boldsymbol{x}), (1/\tau_i)\,\boldsymbol{u}_{p_i}(\cdot,\boldsymbol{x})\big)_{0,[0,T]}\ \ \textit{for } i = 0,\dots,L$$

with $(\boldsymbol{u},\boldsymbol{v})_{0,[0,T]} = \int_0^T \boldsymbol{u}(t)\boldsymbol{v}(t)\,\mathrm{d}t$ *and* $\boldsymbol{x} \in \Omega$*. We suppressed the spatial dependencies of* $\varrho, v_\mathrm{p}, \tau_\mathrm{p}, R, T_0, \dots, T_L$ *in the representation above.*

Proof. We calculate the representation of the adjoint directly. Therefore, we start with

$$\begin{aligned}\big(\boldsymbol{y}, \boldsymbol{M}'_{\mu,\boldsymbol{u}}[\eta]\big)_{0,[0,T]\times\Omega} &= \big(\boldsymbol{y}, \boldsymbol{M}'(\mu)[\eta]\boldsymbol{u}\big)_{0,[0,T]\times\Omega}\\ &= \int_\Omega \int_{[0,T]} \boldsymbol{y}(t,\boldsymbol{x}) \cdot (\boldsymbol{M}'(\mu)[\eta](\boldsymbol{x})\boldsymbol{u}(t,\boldsymbol{x}))\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x}\,.\end{aligned}$$

The scalar product leads to a sum over the components of the vectors. We look at those components one by one and start with the velocity component:

$$\begin{aligned}\int_\Omega \int_{[0,T]} \boldsymbol{y_v}(t,\boldsymbol{x}) \cdot (\eta_\varrho(\boldsymbol{x})\boldsymbol{u_v}(t,\boldsymbol{x}))\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x} &= \int_\Omega \eta_\varrho(\boldsymbol{x}) \int_{[0,T]} \boldsymbol{y_v}(t,\boldsymbol{x}) \cdot \boldsymbol{u_v}(t,\boldsymbol{x})\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x}\\ &= \big\langle \eta_\varrho, (\boldsymbol{y_v},\boldsymbol{u_v})_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\,.\end{aligned}$$

For the first pressure component we get

$$\begin{aligned}&\int_\Omega \int_{[0,T]} \boldsymbol{y}_{p_0}(t,\boldsymbol{x}) \cdot \nu(\varrho(\boldsymbol{x}), v_\mathrm{p}(\boldsymbol{x}), \tau_\mathrm{p}(\boldsymbol{x})) \left(\frac{-\eta_\varrho(\boldsymbol{x})}{\varrho(\boldsymbol{x})} - \frac{2\eta_{v_\mathrm{p}}(\boldsymbol{x})}{v_\mathrm{p}(\boldsymbol{x})} + \frac{\alpha\eta_{\tau_\mathrm{p}}(\boldsymbol{x})}{1+\alpha\tau_\mathrm{p}(\boldsymbol{x})}\right) \boldsymbol{u}_{p_0}(t,\boldsymbol{x})\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x}\\ &= \big\langle \eta_\varrho, -\frac{\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})}{\varrho}\big(\boldsymbol{y}_{p_0},\boldsymbol{u}_{p_0}\big)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\\ &\quad + \big\langle \eta_{v_\mathrm{p}}, -\frac{2\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})}{v_\mathrm{p}}\big(\boldsymbol{y}_{p_0},\boldsymbol{u}_{p_0}\big)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\\ &\quad + \big\langle \eta_{\tau_\mathrm{p}}, \frac{\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})\alpha}{1+\alpha\tau_\mathrm{p}}\big(\boldsymbol{y}_{p_0},\boldsymbol{u}_{p_0}\big)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}.\end{aligned}$$

For the remaining pressure components we get

$$\begin{aligned}&\int_\Omega \int_{[0,T]} \boldsymbol{y}_{p_l}(t,\boldsymbol{x}) \cdot \tilde\nu(\varrho(\boldsymbol{x}), v_\mathrm{p}(\boldsymbol{x}), \tau_\mathrm{p}(\boldsymbol{x})) \left(\frac{-\eta_\varrho(\boldsymbol{x})}{\varrho(\boldsymbol{x})} + \frac{-2\eta_{v_\mathrm{p}}(\boldsymbol{x})}{v_\mathrm{p}(\boldsymbol{x})} + \frac{-\eta_{\tau_\mathrm{p}}(\boldsymbol{x})}{(1+\alpha\tau_\mathrm{p}(\boldsymbol{x}))\tau_\mathrm{p}(\boldsymbol{x})}\right) \frac{1}{\tau_l}\boldsymbol{u}_{p_l}(t,\boldsymbol{x})\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x}\\ &= \big\langle \eta_\varrho, \frac{-\tilde\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})}{\varrho}\left(\boldsymbol{y}_{p_l}, \frac{1}{\tau_l}\boldsymbol{u}_{p_l}\right)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\\ &\quad + \big\langle \eta_{v_\mathrm{p}}, \frac{-2\tilde\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})}{v_\mathrm{p}}\left(\boldsymbol{y}_{p_l}, \frac{1}{\tau_l}\boldsymbol{u}_{p_l}\right)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\\ &\quad + \big\langle \eta_{\tau_\mathrm{p}}, \frac{-\tilde\nu(\varrho,v_\mathrm{p},\tau_\mathrm{p})}{(1+\alpha\tau_\mathrm{p})\tau_\mathrm{p}}\left(\boldsymbol{y}_{p_l}, \frac{1}{\tau_l}\boldsymbol{u}_{p_l}\right)_{0,[0,T]} \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\,.\end{aligned}$$

To obtain the final representation we have to add all equations above and group the summands by the η terms. Define $R(\boldsymbol{x}) := (\boldsymbol{y_v}(\boldsymbol{x}), \boldsymbol{u_v}(\boldsymbol{x}))_{0,[0,T]}$ and $T_0(\boldsymbol{x}) := \big(\boldsymbol{y}_{p_0}(\boldsymbol{x}), \boldsymbol{u}_{p_0}(\boldsymbol{x})\big)_{0,[0,T]}$, $T_i(\boldsymbol{x}) :=$

$\left(\boldsymbol{y}_{p_i}(\boldsymbol{x}), \frac{1}{\tau_i}\boldsymbol{u}_{p_i}(\boldsymbol{x})\right)_{0,[0,T]}$, $i = 1, \ldots, L$. Combining the results above we get

$$\begin{aligned}
\left(\boldsymbol{y}, \boldsymbol{M}'_{\mu,\boldsymbol{u}}[\eta]\right)_{0,[0,T]\times\Omega} &= \Big\langle \eta_\varrho, R - \frac{\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{\varrho} T_0 - \frac{\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{\varrho} \sum_{l=1}^{L} T_l \Big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'} \\
&+ \Big\langle \eta_{v_\mathrm{p}}, -\frac{2\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{v_\mathrm{p}} T_0 - \frac{2\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{v_\mathrm{p}} \sum_{l=1}^{L} T_l \Big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'} \\
&+ \Big\langle \eta_{\tau_\mathrm{p}}, \frac{\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})\alpha}{1+\alpha\tau_\mathrm{p}} T_0 - \frac{\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{(1+\alpha\tau_\mathrm{p})\tau_\mathrm{p}} \sum_{l=1}^{L} T_l \Big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'} \\
&= \Big\langle \eta, \begin{pmatrix} R - \frac{\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{\varrho} T_0 - \frac{\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{\varrho} \sum_{l=1}^{L} T_l \\ -\frac{2\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{v_\mathrm{p}} T_0 - \frac{2\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{v_\mathrm{p}} \sum_{l=1}^{L} T_l \\ \frac{\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})\alpha}{1+\alpha\tau_\mathrm{p}} T_0 - \frac{\tilde{\nu}(\varrho, v_\mathrm{p}, \tau_\mathrm{p})}{(1+\alpha\tau_\mathrm{p})\tau_\mathrm{p}} \sum_{l=1}^{L} T_l \end{pmatrix} \Big\rangle_{L^\infty(\Omega)^3\times(L^\infty(\Omega)^3)'},
\end{aligned}$$

proving the representation. Note that $R, T_0, \ldots, T_L$ are in $L^1(\Omega)$ since $\boldsymbol{u}, \boldsymbol{y}$ are in the L^2 space $\boldsymbol{Y}$, the parameter functions are bounded, and Ω is bounded. This yields $\boldsymbol{M}'^{*}_{\mu,\boldsymbol{u}}\boldsymbol{y} \in L^1(\Omega)^3 \subset \boldsymbol{P}'$. □

In a similar vein we can define an adjoint of the linearized differential operator.

Corollary 2. *The adjoint of the linear operator $\mathfrak{L}'_{\mu,\boldsymbol{u}}$ from Theorem 5 is*

$$\mathfrak{L}'^{*}_{\mu,\boldsymbol{u}} \colon \boldsymbol{Y} \longrightarrow \boldsymbol{P}'_\mathrm{adm}, \quad \mathfrak{L}'^{*}_{\mu,\boldsymbol{u}}[\boldsymbol{y}] = -\boldsymbol{M}'^{*}_{\mu,\partial_t\boldsymbol{u}+\boldsymbol{D}\boldsymbol{u}}[\boldsymbol{y}]\,.$$

Proof. We can calculate the adjointness directly by

$$\begin{aligned}
\left(\boldsymbol{y}, \mathfrak{L}'_{\mu,\boldsymbol{u}}[\eta]\right)_{0,[0,T]\times\Omega} &= \left(\boldsymbol{y}, -\boldsymbol{M}'_{\mu,\partial_t\boldsymbol{u}+\boldsymbol{D}\boldsymbol{u}}[\eta]\right)_{0,[0,T]\times\Omega} \\
&= \big\langle -\boldsymbol{M}'^{*}_{\mu,\partial_t\boldsymbol{u}+\boldsymbol{D}\boldsymbol{u}}[\boldsymbol{y}], \eta \big\rangle_{L^\infty(\Omega)\times L^\infty(\Omega)'}\,.
\end{aligned}$$

□

Now, we can formulate an equation for the adjoint of the Fréchet derivative. Here, we need the adjoint equation from Section 2.3.

Theorem 6. *Using the notation from Theorem 5, we set $\mathcal{F}'_\mu[\eta] = \mathcal{F}'(\mu)[\eta]$ and utilize the definition of $\mathfrak{L}^\star$ given in (2.22), we have*

$$\mathcal{F}'^{*}_\mu \colon \boldsymbol{Y} \longrightarrow \boldsymbol{P}', \;\; \boldsymbol{y} \longmapsto \mathcal{F}'^{*}_\mu[\boldsymbol{y}] = -\mathfrak{L}'^{*}_{\mu,\mathcal{F}(\mu)}[\boldsymbol{z}]\,,$$

where $\boldsymbol{z} \in C([0,T], \boldsymbol{X})$ is the mild solution of $\mathfrak{L}^\star\boldsymbol{z}(t) = \boldsymbol{y}(t)$, a.e. in $[0,T]$, $\boldsymbol{z}(T) = \boldsymbol{0}$.

Proof. Mild solutions are solutions with less regularity (see Definition 2). Plugging in the assumption $\mathfrak{L}^\star\boldsymbol{z} = \boldsymbol{y}$ into the scalar product yields

$$\begin{aligned}
\left(\mathcal{F}'_\mu[\eta], \boldsymbol{y}\right)_{0,[0,T]\times\Omega} &= \left(\mathcal{F}'_\mu[\eta], \mathfrak{L}^\star\boldsymbol{z}\right)_{0,[0,T]\times\Omega} \\
&= \left(\mathfrak{L}\mathcal{F}'(\mu)[\eta], \boldsymbol{z}\right)_{0,[0,T]\times\Omega} \\
&\overset{(4.3)}{=} \left(-\mathfrak{L}'_{\mu,\mathcal{F}(\mu)}[\eta], \boldsymbol{z}\right)_{0,[0,T]\times\Omega} \\
&= \big\langle \eta, -\mathfrak{L}'^{*}_{\mu,\mathcal{F}(\mu)}[\boldsymbol{z}] \big\rangle_{L^\infty(\Omega)^3\times(L^\infty(\Omega)^3)'}\,.
\end{aligned}$$

□

With this in place, we can calculate the adjoint of the linearized parameter-to-seismogram map.

Corollary 3. *With the notation of Theorem 6 and the definition $\Phi'_\mu[\eta] = \Phi'(\mu)[\eta]$ we can formulate the adjoint of Φ'_μ by*

$$\Phi'^*_\mu \colon \boldsymbol{S}^\Xi \longrightarrow \boldsymbol{P}', \ \boldsymbol{s} \longmapsto \sum_{\xi=0}^{\Xi-1} \mathcal{F}'^*_{\mu,\xi}[\Psi^* \boldsymbol{s}_\xi] \, .$$

Proof. Again, we calculate it directly:

$$\begin{aligned}
\big(\Phi'_\mu[\eta], \boldsymbol{s}\big)_{\boldsymbol{S}^\Xi} &= \sum_{\xi=0}^{\Xi-1} \big(\Psi \mathcal{F}'_{\mu,\xi}[\eta], \boldsymbol{s}_\xi\big)_{\boldsymbol{S}} \\
&= \sum_{\xi=0}^{\Xi-1} \big(\mathcal{F}'_{\mu,\xi}[\eta], \Psi^* \boldsymbol{s}_\xi\big)_{0,[0,T]\times\Omega} \\
&= \big\langle \eta, \sum_{\xi=0}^{\Xi-1} \mathcal{F}'^*_{\mu,\xi}[\Psi^* \boldsymbol{s}_\xi]\big\rangle_{L^\infty(\Omega)^3 \times (L^\infty(\Omega)^3)'} \, .
\end{aligned}$$

□

Remark 4. *In the following we will continue to use the parameterization with $(\varrho, v_\mathrm{p}, \tau_\mathrm{p})$. In our calculations we want to avoid non-physical negative values for the parameters. Hence, we use a parameterization that enforces positivity for all parameters and invert for $\tilde\varrho, \tilde v_\mathrm{p}, \tilde\tau_\mathrm{p}$ with the definitions $\varrho = \exp(\tilde\varrho), v_\mathrm{p} = \exp(\tilde v_\mathrm{p}), \tau_\mathrm{p} = \exp(\tilde\tau_\mathrm{p})$. For the explicit formulas we just have to calculate the derivative and its adjoint of $\widetilde{\boldsymbol{M}}(\tilde\varrho, \tilde v_\mathrm{p}, \tilde\tau_\mathrm{p}) = \boldsymbol{M}(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))$.*

$$\begin{aligned}
\widetilde{\boldsymbol{M}}'(\tilde\varrho, \tilde v_\mathrm{p}, \tilde\tau_\mathrm{p})[\tilde\eta] = \mathrm{diag}(\, &\exp(\tilde\varrho) \cdot \tilde\eta_{\tilde\varrho}, \nabla\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))[\exp(\tilde\varrho)\tilde\eta_{\tilde\varrho}, \exp(\tilde v_\mathrm{p})\tilde\eta_{\tilde v_\mathrm{p}}, \exp(\tilde\tau_\mathrm{p})\tilde\eta_{\tilde\tau_\mathrm{p}}], \\
&\nabla\tilde\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))[\exp(\tilde\varrho)\tilde\eta_{\tilde\varrho}, \exp(\tilde v_\mathrm{p})\tilde\eta_{\tilde v_\mathrm{p}}, \exp(\tilde\tau_\mathrm{p})\tilde\eta_{\tilde\tau_\mathrm{p}}])
\end{aligned}$$

and adjoint $\widetilde{\boldsymbol{M}}'^*_{\tilde\mu,\boldsymbol{u}} \colon \boldsymbol{Y} \longrightarrow \boldsymbol{P}'_\mathrm{adm}$, which is expressed by

$$\widetilde{\boldsymbol{M}}'^*_{\mu,\boldsymbol{u}} \boldsymbol{y} = \begin{pmatrix} R - \nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))T_0 - \tilde\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}) \sum_{l=1}^L T_l \\ -2\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))T_0 - 2\tilde\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p})) \sum_{l=1}^L T_l \\ \frac{\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))\exp(\tilde\tau_\mathrm{p})\alpha}{1+\alpha\exp(\tilde\tau_\mathrm{p})} T_0 - \frac{\tilde\nu(\exp(\tilde\varrho), \exp(\tilde v_\mathrm{p}), \exp(\tilde\tau_\mathrm{p}))}{1+\alpha\exp(\tilde\tau_\mathrm{p})} \sum_{l=1}^L T_l \end{pmatrix} .$$

In the following sections we drop the (mathematically correct) notation with the dual pairings for the materials and use a scalar product notation. This is justified, since from now on we assume to be in discrete spaces.

4.2 Discretization aspects for the Fréchet derivative and its adjoint

With the results from the last section we have the formulas we need to calculate the Fréchet derivative and its adjoint. To evaluate them, it is necessary to have access to $\partial_t \mathcal{F}(\mu)$ and $\mathcal{F}(\mu)$ at the same time. We only save $\mathcal{F}(\mu)$ and calculate $\partial_t \mathcal{F}(\mu)$ on the fly with a one-sided finite difference. For large scale applications, for example in three dimensions, saving full wave fields is often not feasible. Therefore, several methods to reduce RAM storage have been developed. For instance, only saving snapshots of

$\mathcal{F}(\mu)$ and calculating them on the fly for the adjoint. This adds calculation time. Several techniques for FWI are presented in [BHDlPF16]. They include using fewer bits for each stored value, reducing the degree of the ansatz polynomial, temporal compression by saving less time steps, and using interpolation. We do not apply such methods, we have sufficient capacity for our calculations on the machines we use (up to 1 TB RAM).

Let $\boldsymbol{P}_h$ be the discrete dG parameter space as in Section 3.1, equipped with the L^2 scalar product, hence $\boldsymbol{P}_h \subset L^2(\Omega)^3$. In our case we assume the polynomial degree to be zero and therefore the parameters are cell-wise constant. Now, we want to discuss the actual formulas that have to be implemented to calculate $\Phi'^{*}_{\mu}[\boldsymbol{s}]$ for a $\boldsymbol{s} \in \boldsymbol{S}$. Let $\boldsymbol{z} \in C^1([0,T], \boldsymbol{X})$ fulfill $\boldsymbol{\mathfrak{L}}^{\star}\boldsymbol{z} = \Psi^{*}\boldsymbol{s}$. Then, we have to solve the variational problem

$$\text{find } \eta^{*} \in \boldsymbol{P}_h \text{ such that } (\eta^{*}, \eta_i)_{0,\Omega^3} = \big(\boldsymbol{\mathfrak{L}}'^{*}_{\mu,\boldsymbol{u}}[\boldsymbol{z}], \eta_i\big)_{0,\Omega^3} \, \forall \eta_i \in \boldsymbol{P}_h \,,$$

with $\boldsymbol{\mathfrak{L}}'^{*}_{\mu,\boldsymbol{u}}$ from Corollary 2. To find the coefficients of η^{*}, we have to solve a linear system where the stiffness matrix contains the pairwise scalar products of the η_i's. For the discontinuous Galerkin ansatz space this is a block diagonal matrix. Since we assume cell-wise constant material parameters, this matrix becomes a diagonal matrix with the size of the cell on the diagonal. With $\boldsymbol{z}$ from above and $\boldsymbol{u} = \mathcal{F}(\mu)$ we can explicitly give the formulas for the coefficients of η^{*} for each material parameter. For better readability we suppress the dependence on t and $\boldsymbol{x}$

$$\begin{aligned}
\eta^{*}_{\varrho}|_K &= |K|^{-1} \int_0^T \int_K -\boldsymbol{z}_{\boldsymbol{v}} \boldsymbol{u}_{\boldsymbol{v}} + \frac{\tilde{\nu}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})}{\varrho}|_K \left(\tau_{\mathrm{p}}|_K \boldsymbol{z}_{p_0} \partial_t \boldsymbol{u}_{p_0} + \sum_{l=1}^{L} \frac{1}{\tau_l} \boldsymbol{z}_{p_l} \boldsymbol{u}_{p_l} \right) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \,, \\
\eta^{*}_{v_{\mathrm{p}}}|_K &= |K|^{-1} \int_0^T \int_K \frac{2\tilde{\nu}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})}{v_{\mathrm{p}}}|_K \left(\tau_{\mathrm{p}} \boldsymbol{z}_{p_0} \partial_t \boldsymbol{u}_{p_0} + \sum_{l=1}^{L} \frac{1}{\tau_l} \boldsymbol{z}_{p_l} \boldsymbol{u}_{p_l} \right) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \,, \qquad (4.4) \\
\eta^{*}_{\tau_{\mathrm{p}}}|_K &= |K|^{-1} \int_0^T \int_K -\frac{\nu(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})\alpha}{1 + \alpha\tau_{\mathrm{p}}}|_K \boldsymbol{z}_{p_0} \partial_t \boldsymbol{u}_{p_0} + \frac{\tilde{\nu}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})}{(1 + \alpha\tau_{\mathrm{p}})\tau_{\mathrm{p}}}|_K \sum_{l=1}^{L} \frac{1}{\tau_l} \boldsymbol{z}_{p_l} \boldsymbol{u}_{p_l} \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \,.
\end{aligned}$$

4.3 The `Red-CG-REGINN` algorithm

Remember that in this section the goal is to solve

$$\text{find} \quad \mu \in \boldsymbol{P}_{\text{adm}} \quad \text{such that} \quad \Phi(\mu) = \boldsymbol{s}_{\text{obs}} \in \boldsymbol{S}^{\Xi}$$

with a Newton-type method and noisy data $\boldsymbol{s}_{\text{obs}}$. Define $G(\mu) = \Phi(\mu) - \boldsymbol{s}_{\text{obs}}$ and apply the Newton method to the root finding problem $G(\mu) = 0$. Choosing an inexact Newton method yields in the k-th nonlinear step

$$\mu^{k+1} = \mu^k + \triangle\mu, \text{ with } G'(\mu^k)[\triangle\mu] \approx -G(\mu^k)\,.$$

Since there is no guarantee that the linearized problem even has a solution, we want to find a solution in the least squares sense. Therefore, we turn to the normal equation

$$G'^*(\mu^k)G'(\mu^k)[\triangle\mu] \approx G'^*(\mu^k)[-G(\mu^k)] \Leftrightarrow \Phi'^*_{\mu^k}\Phi'(\mu^k)[\triangle\mu] \approx \Phi'^*_{\mu^k}[\boldsymbol{s}_{\text{obs}} - \Phi(\mu^k)]. \tag{4.5}$$

The combination of a regularized inexact Newton method and a regularized `CG` method for the linearized problem is called `CG-REGINN` (where `REGINN` is the abbreviation for `REG`ularization by `IN`exact `N`ewton and was proposed in [Rie99]). The combination of these two algorithms is well-suited for a nonlinear locally ill-posed problem since in [Rie05] it is shown that this is a regularization method. Further, we cycle through the shots and update after every shot. This makes it a Kaczmarz method which is well-known variant of iterative methods in the inverse problem community [HLS07, HKLS07]. We call the combination of the algorithms in the reduced case `Red-CG-REGINN`. Note that even if Φ is locally ill-posed everywhere, this does not imply that the linear problem is ill-posed (in the sense of Nashed, that is, the range of the linear operator is not closed). There are properties for nonlinear operators that do imply the ill-posedness for the linearization [Hof99], but they are typically hard to prove. In the case of FWI these properties were not shown yet. Generally, we are on the safe side when we just assume ill-posedness of the linearized problem. Convergence analysis of Newton-like schemes often need that a structural assumption on the nonlinearity, one of which is the *tangential cone condition*[1] (TCC), holds. The TCC is very hard to prove in the continuous setting for most nonlinear problems. However, in [ER21] Eller and Rieder show a TCC for semi-discrete FWI in the acoustic regime and for the elastic case in [EGR22]. Note that when we introduced the inexact Newton method, we skipped over some mathematical problems. The `CG` method is a Hilbert space method, and the domain of $\Phi'(\mu)$ for any $\mu \in \text{int}(\boldsymbol{P}_{\text{adm}})$ is $L^\infty(\Omega)^3$, which is a non-reflexive Banach space. Hence, we cannot apply the `CG` algorithm directly in the continuous setting. However, since we are not interested in a convergence analysis on the continuous problem, we look for the solution in the discrete spaces $\boldsymbol{P}_h \subset L^2(\Omega)^3$ and $\boldsymbol{Y}_h$, which both are Hilbert spaces. We then search for a solution in the discrete spaces. A `REGINN` variant that is applicable even for L^∞ is presented in [PR23]. To complete the algorithm we employ the stopping criterion for the `CG` method, which is from [Win16] (other strategies are also possible, see [Rie99, Sec. 6] for an explanation). In `REGINN`, we call the while loop starting at line 2 and ending at line 13 of Algorithm 4.2 the *outer loop*. We call the repeat loop that spans the lines 5-11 in Algorithm 4.2 the *inner loop*. In the inner loop, the update is

[1] V, W Banach spaces, $F\colon V \supset \mathcal{D} \longrightarrow W$ a Fréchet differentiable nonlinear operator. F satisfies the TCC at $x^+ \in \text{int}(\mathcal{F})$ if $\|F(v) - F(w) - F'(w)(v-w)\|_W \le \omega\|F(v) - F(w)\|_W$, for all $v, w \in B_r(x^+)$ for a $\omega < 1$, where $B_r(x^+)$ is the open ball in V with radius $r > 0$ about x^+.

calculated by iteratively solving (4.5). For shot ξ we set the maximal iteration number of the inner loop to $l^k_{\max} = \max\{\tilde{l}^k_{\max,\xi}, l_{\max,\text{glob}}\}$ with

$$\tilde{l}^k_{\max,\xi} = \begin{cases} 1 & \text{for } k = 0\,, \\ 2 & \text{for } k = 1\,, \\ l^{k-1}_\xi + l^{k-2}_\xi & \text{for } k \geq 2\,. \end{cases} \tag{4.6}$$

By l^{k-2}_ξ, l^{k-1}_ξ we denote the amount of iterations the inner loop needed to terminate in the last two inner loop calls. Further, $l_{\max,\text{glob}}$ is a preset number of maximal inner iterations to limit the maximal cost of each inner loop call. Let $\tilde{\boldsymbol{r}}^l$ be the residual in the inner loop, then the relative reduction factor $\vartheta^k_\xi \in (0,1)$ for which we break the inner loop when the condition $\|\tilde{\boldsymbol{r}}^l\|_{\boldsymbol{S}} \leq \vartheta^k_\xi \|\tilde{\boldsymbol{r}}^0\|_{\boldsymbol{S}}$ is defined by

$$\vartheta^k_\xi = \min\left\{0.999,\, \widetilde{\vartheta}^k_\xi\right\}, \text{ with } \widetilde{\vartheta}^k_\xi = \begin{cases} 1 & \text{for } k = 0\,, \\ \|\tilde{\boldsymbol{r}}^1\|_{\boldsymbol{S}} \,/\, \|\tilde{\boldsymbol{r}}^0\|_{\boldsymbol{S}} & \text{for } k = 1\,, \\ 1 - (l^{k-2}_\xi / l^{k-1}_\xi)\; (1 - \vartheta^{k-1}_\xi) & \text{for } l^{k-1}_\xi > l^{k-2}_\xi \;\wedge\; k \geq 2\,, \\ \gamma\, \vartheta^{k-1}_\xi & \text{otherwise}\,. \end{cases} \tag{4.7}$$

This gives two conditions when to break the loop, either we reach $l_{\max,k}$, or we achieve a relative reduction in the residual by ϑ^k_ξ. The resulting algorithm for the inner loop can be seen in Algorithm 4.1. Finally, we add the possibility of using a preconditioner in the `CG` algorithm. Let us assume that the measurement is contaminated with noise, therefore we write $\boldsymbol{s}^\delta_{\text{obs}}$. If we were able to measure without noise $\boldsymbol{s}_{\text{obs}}$ would the "true" measurement. Then, the noise level δ is $\|\boldsymbol{s}_{\text{obs}} - \boldsymbol{s}^\delta_{\text{obs}}\|_{0,\Omega} \leq \delta$. A crucial part of nonlinear regularization methods is the stopping rule for the outer loop. A widespread choice for inverse problems is Morozov's discrepancy principle see ([EHN96]), which stops the algorithm when for some $D > 1$ it holds

$$\|\boldsymbol{s}_{\text{obs}} - \Phi(\mu^K)\|_{\boldsymbol{S}} \leq D\|\boldsymbol{s}_{\text{obs}} - \boldsymbol{s}^\delta_{\text{obs}}\|_{\boldsymbol{S}} < \|\boldsymbol{s}_{\text{obs}} - \Phi(\mu^k)\|_{\boldsymbol{S}} \qquad k = 0, \ldots, K-1.$$

Therefore, we stop the iteration when the nonlinear residual is close to the noise level and K the smallest index that fulfills the discrepancy principle. This discrepancy principle is used as parameter choice in many algorithms for linear and nonlinear inverse problems. It ensures that the algorithm stops and is in fact also used in the analysis of `CG-REGINN` in [Rie05]. In realistic applications, however, we often do not know the exact level of noise, and we have to hand-tune the parameter D. In geophysical applications there is no "gold standard" with regard to the stopping rule for the outer iteration. Also, most publications do not focus on the stopping rule. Some publications use a fixed maximal number of iterations [KFBI18, EHO+12, YBM+18], stop if an update has little effect on the norm of the residual [FKIB09, ABG+13], or the norm of the update is small [FKIB10, Kur12]. Sometimes the stopping criterion is not even specified. In our experiments, we will always use a fixed number of iterations that was determined by balancing the need to minimize calculation times and to have a solution that is good enough. We do not investigate stopping criteria further. Possible choices for rules that do not depend on the (unknown) level of noise are for example the method by Hanke(-Bourgeois) and Raus (see [HR96]) or the L-Curve criterion (see [EHN96]). The whole `Red-CG-REGINN` algorithm is shown in Algorithm 4.2. Throughout this thesis, the term "one iteration" shall refer to an entire cycle through all the sources.

Remark 5. *From the solution of well-posed problems it is well-known that Kaczmarz methods can be drastically improved [SV09] by choosing the shots randomly (uniformly or with probabilities, depending*

Algorithm 4.1 Preconditioned Conjugate Gradient (PCG) algorithm for the normal equation restricted to $\boldsymbol{P}_h$

```
Require: $\boldsymbol{r}^k \in \boldsymbol{S}$, $\mu^k \in \boldsymbol{P}_h$, $\vartheta \in (0,1)$, $l_{\max} \in \mathbb{N}$, Precon
Ensure: $\triangle\mu^k \in \boldsymbol{P}_h$, $l_k \in \mathbb{N}$
$l \leftarrow 0$, $\beta \leftarrow 0$, $\tilde{\boldsymbol{r}}^0 \leftarrow \boldsymbol{r}^k$
$p^0, f^0 \leftarrow 0 \in \boldsymbol{P}_h$
repeat
    $l \leftarrow l + 1$
    $d \leftarrow \Phi'^{*}_{\mu^k}[\tilde{\boldsymbol{r}}^{l-1}] \in \boldsymbol{P}_h$ % solution of one adjoint wave equation as shown in Corollary 3
    $d_{PC} \leftarrow \text{Precon}(d)$
    $p^l \leftarrow d_{PC} + \beta\,(d_{PC}, d)_{0,\Omega}\, p^{l-1}$
    $\boldsymbol{q} \leftarrow \Phi'_{\mu^k}[p^l] \in \boldsymbol{S}$ % solution of one forward wave equation as shown in Corollary 1
    $\alpha \leftarrow (d_{PC}, d)_{0,\Omega} \,/ \|\boldsymbol{q}\|_{\boldsymbol{S}}$
    $f^l \leftarrow f^{l-1} + \alpha\, p^l$
    $\tilde{\boldsymbol{r}}^l \leftarrow \tilde{\boldsymbol{r}}^{l-1} - \alpha\, \boldsymbol{q}$
    $\beta \leftarrow 1/(d_{PC}, d)_{0,\Omega}$
until $\|\tilde{\boldsymbol{r}}^l\|_{\boldsymbol{S}} \leq \vartheta \|\boldsymbol{r}^k\|_{\boldsymbol{S}}$ or $l \geq l_{\max}$
$\triangle\mu^k \leftarrow f^l$, $l^k \leftarrow l$
return $(\triangle\mu^k, l^k)$
```

on the norm of the residual). In [Shi19] the shots are chosen randomly for FWI in a stochastic inversion process, and the choice shows good convergence properties.

Remark 6. *In the beginning of this section we mentioned that* CG-REGINN *is a suitable regularization method. Here, we want to elaborate a little bit on the term regularization. In the theory of inverse problems a regularization method yields a reasonable (generalized) solution for an ill-posed inverse problem, even if the input data is perturbed by noise. In our case of* CG-REGINN *this is achieved by early-stopping the* CG*-iteration and not "over solving" it in each iteration (regularization of the linearized problem) as well as adding a suitable stopping criterion for the nonlinear iterations (regularization of the nonlinear problem). Another form of regularization (often called* variational regularization*) that is widely used in many applications is by adding a penalty-term $\mathcal{R}\colon \boldsymbol{P}_h \longrightarrow \mathbb{R}$ to the functional one is minimizing. In the case of the reduced problem* (RED) *the functional then has the form*

$$\textit{find } \mu \textit{ such that } J(\mu) = \frac{1}{2}\|\Phi(\mu) - \boldsymbol{s}_{\text{obs}}\|^2_{\boldsymbol{S}} + \beta\mathcal{R}(\mu) \longrightarrow \min \quad \textit{subject to } \boldsymbol{\mathfrak{L}}(\mu)\boldsymbol{u} - \boldsymbol{f} = 0,\ \beta \in \mathbb{R}_+ . \tag{4.8}$$

The choice of $\mathcal{R}$ can stabilize the minimization process or promote certain features in our minimization. One regularization function commonly used in FWI is $\mathcal{R}_T(\mu) = \|\mu - \mu_{\text{ref}}\|^2_{0,\Omega}$, which is the classical Tikhonov regularization for $\mu_{\text{ref}} = 0$ and promotes solutions that are close to μ_{ref}. This regularization term usually produces smooth solutions. Another regularization is the total variation $\mathcal{R}_{\text{TV}}(\mu) = \int_\Omega |\nabla_h \mu| \,\mathrm{d}\boldsymbol{x}$, where ∇_h is a discrete gradient. This regularization term comes from the context of image processing. It is known for its tendency to for sharp edges making it a suitable choice for layered media. With the additional parameter β we can steer how much the penalty influences the algorithm. Usually, we want to choose the parameter not too large to stay close to the original problem but still large enough to profit

Algorithm 4.2 `Red-CG-REGINN`

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess;   $\boldsymbol{s}_{\text{obs}} \in \boldsymbol{S}^{\Xi}$ % observed seismograms
Ensure: $\mu^k \in \boldsymbol{P}_h$ % approximate solution of (RED)
$k \leftarrow 0$, $\xi \leftarrow 0$,
while not termination do
    $\mu_0^k \leftarrow \mu^k$
    $\xi \leftarrow 0$
    repeat
        $\boldsymbol{r}_\xi^k \leftarrow \boldsymbol{s}_{\text{obs},\xi} - \Phi_\xi(\mu_\xi^k)$
        determine $l_{\max,\xi}^k$ and $\vartheta_\xi^k$ % according to (4.7) and (4.6)
        $(\triangle\mu_\xi^k, l_\xi^k) \leftarrow \texttt{PCG}(\boldsymbol{r}_\xi^k, \mu_\xi^k, \vartheta_\xi^k, l_{\max,\xi}^k)$ % call of Algorithm 4.1
        $\mu_{\xi+1}^k = \mu_\xi^k + \triangle\mu_\xi^k$
        $\xi \leftarrow \xi + 1$
    until $\xi > \Xi - 1$
    $\mu^{k+1} \leftarrow \mu_{\Xi-1}^k$
    $k \leftarrow k + 1$
end while
return $\mu^k$
```

from the regularization.

Even discretizing a continuous problem can already be considered as regularization. In our case this effect is negligible, because we do not use hierarchical methods or problem-adapted ansatz functions such as wavelets. For further reading and rigorous mathematical analysis of iterative methods for nonlinear ill-posed problems we refer to [KNS08]. For the standard theory on linear ill-posed problems [EHN96] and for the more geophysical side of the topic we refer to [Tar05, Fic11].

4.3.1 Connection to geophysics

In this section we want to elaborate on how the problem is formulated in most geophysical publications. In geophysics, the problem of FWI is considered a nonlinear optimization problem and was first formulated by Tarantola in [Tar84]. To this day physicists tackle the problem [VO09] this way. We mentioned the formulation in (4.8). Now set $\mathcal{R} \equiv 0$ and obtain

$$\text{find } \mu \text{ such that } J(\mu) = \frac{1}{2}\|\Phi(\mu) - \boldsymbol{s}_{\text{obs}}\|_{\boldsymbol{S}}^2 \longrightarrow \min \ \text{ subject to } \mathcal{F}(\mu) = \boldsymbol{u}.$$

Since the differential operator does not appear in the functional that is minimized, this formulation is called *reduced* formulation. The equation is 'reduced' from the functional and appears as a constraint. If we consider FWI as an optimization problem, we need a descent direction. A standard way is to use the negative gradient. Formally taking the derivative of J yields

$$\begin{aligned} J'(\mu)[\eta] &= (\Psi\mathcal{F}(\mu) - \boldsymbol{s}_{\text{obs}}, \Psi\mathcal{F}'(\mu)[\eta])_{\boldsymbol{S}} \\ &= (\mathcal{F}'^*(\mu)\Psi^*[\Psi\mathcal{F}(\mu) - \boldsymbol{s}_{\text{obs}}], \eta)_{0,\Omega^3} \\ &= (\Phi'^*(\mu)[\Phi(\mu) - \boldsymbol{s}_{\text{obs}}], \eta)_{0,\Omega^3} \,. \end{aligned}$$

Therefore, $\nabla J(\mu) = -\Phi'^*(\mu)[\Phi(\mu) - \boldsymbol{s}_{\text{obs}}]$ is often called the gradient in FWI contexts. If $\mathcal{R} \not\equiv 0$ and is differentiable, we can still carry out this procedure, and it would yield a different gradient. Formally differentiating one more time gives us

$$\begin{aligned} J''(\mu)[\eta_1,\eta_2] &= (\Phi'(\mu)[\eta_2], \Phi'(\mu)[\eta_1])_{\boldsymbol{S}} + (\Psi\mathcal{F}(\mu) - \boldsymbol{s}_{\text{obs}}, \Psi\mathcal{F}''(\mu)[\eta_1,\eta_2])_{\boldsymbol{S}} \\ &= (\Phi'^*(\mu)\Phi'(\mu)[\eta_1], \eta_2)_{0,\Omega^3} + (\Phi''^*(\mu)[\eta_1,\cdot][\Phi(\mu) - \boldsymbol{s}_{\text{obs}}], \eta_2)_{\boldsymbol{S}} \\ &= \left(\Phi'^*(\mu)\Phi'(\mu)[\eta_1] + \Phi''^*_\mu[\eta_1,\cdot][\Phi(\mu) - \boldsymbol{s}_{\text{obs}}], \eta_2\right)_{0,\Omega^3} . \end{aligned}$$

To find a minimizer of J, we can use the Newton method applied to J'. This yields the Newton equation

$$\Phi'^*(\mu)\Phi'(\mu)[\Delta] + \Phi''^*(\mu)[\Delta,\cdot][\Phi(\mu) - \boldsymbol{s}_{\text{obs}}] = \Phi'^*(\mu)[\Phi(\mu) - \boldsymbol{s}_{\text{obs}}] . \tag{4.9}$$

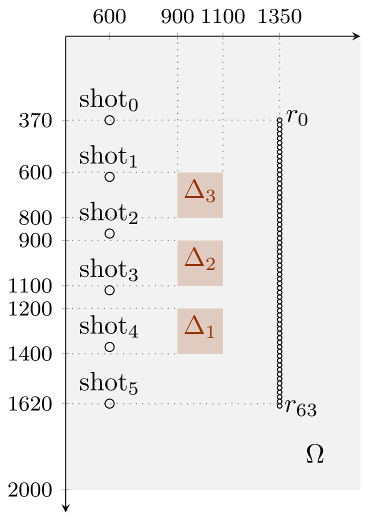
Neglecting the second derivative yields the Gauß-Newton approximation. Methods that do not incorporate information about higher derivatives of Φ than the first one are often called gradient methods. Methods that approximate the second derivative are, for instance, Quasi-Newton methods like L-BFGS. This method is a standard algorithm for large scale optimization problems [NN91] as well as in geophysics e.g. in [DCC17]. Truncated Newton methods solve equation (4.9) approximately as was done in [MBB+14]. They could also be called Inexact-Newton methods for the problem $J' = 0$. Note that the use of the name Newton method differs. While we used the name Newton method for the iterative root-finding algorithm in general, in optimization contexts it specifically means applying the root finding algorithm to $\nabla J(\mu) = 0$. The formulas necessary for the calculation of the second derivative of Φ and its adjoint can be found in [KR19], where they are derived in a rigorous mathematical framework.

4.4 Red-CG-REGINN Mono-parameter reconstruction

In this section we show viability of Red-CG-REGINN for mono-parameter settings, that is, we only invert for one of the parameters while leaving the others constant. We look at a transmission problem and then at the more complex and layered Marmousi model.

4.4.1 Transmission problem

We discuss our first inversion setup. We consider the transmission geometry as shown in Fig. 4.1 and conduct experiments in a computational domain $\Omega = (406.25, 1706.25)\text{m} \times (0, 2000)\text{m}$ and the extended domain $\Omega_E = (106.25, 2006.25)\text{m} \times (-400, 2400)\text{m}$. In the absorbing layer Ω_E we use the parameters $Q_b = 15, v_{\text{p},t} = 0.85 v_{\text{p},i}, v_{\text{p},b} = 0.2 v_{\text{p},i}, s_t = 300$ as explained in Section 3.5. The computational mesh has the width $h = 12.5\,\text{m}$ and we use a polynomial degree of $k = 2$ for the ansatz functions. The forward and adjoint wave equations are solved with the exponential midpoint rule (3.4) with end time $T = 0.6\,\text{s}$ and $\triangle t = 0.0005\,\text{s}$. The number of degrees of freedom in space is $\text{dof}_h = 1198080$. We apply a fixed number of 10 iterations and set $\gamma = 0.9$ as the reduction factor for the inner stopping rule (4.7). In the domain are three areas $\Delta_1 = [900, 1100]\text{m} \times [600, 800]\text{m}, \Delta_2 = [900, 1100]\text{m} \times [900, 1100]\text{m}, \Delta_3 = [900, 1100]\text{m} \times [1200, 1400]\text{m}$, where we change the values of the parameters $\varrho, v_{\text{p}}, \tau_{\text{p}}$. We place $\Xi = 6$ equally spaced sources on a line between $\boldsymbol{x}_{\text{src},0} = (600\,\text{m}, 370\,\text{m})^\top$ and $\boldsymbol{x}_{\text{src},5} = (600\,\text{m}, 1620\,\text{m})^\top$ on the left side of $\Delta_i, i = 1, 2, 3$. Further, we place $R = 64$ equally distanced receivers on the right side of the



T	0.6
$\triangle t$	0.0005
L	5
k	2
h	12.5
dof_h	1198080

Figure 4.1: Left: geometry, sources, inclusions, and receivers for the transmission problem. Right: parameters of the numerical calculation.

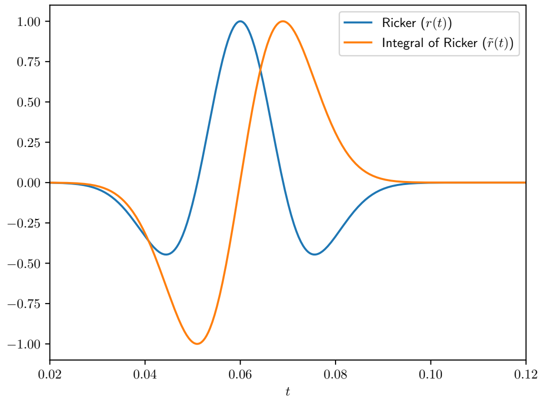


Figure 4.2: Ricker wavelet and its integral with $f_c = 25$Hz and $t_{\text{shift}} = 1.5/f_c$.

$\Delta_i, i = 1,2,3$ starting at $\boldsymbol{r}_0 = (1350\,\mathrm{m}, 370\,\mathrm{m})^\top$ and ending at $\boldsymbol{r}_{63} = (1350\,\mathrm{m}, 1630\,\mathrm{m})^\top$. For the source signal in time we choose a Ricker wavelet

$$r(t) = (1 - 2 \cdot \sigma(t) \cdot \sigma(t)) \cdot \exp(-\sigma(t) \cdot \sigma(t)),\ \sigma(t) = \pi \cdot f_c \cdot (t - t_{\text{shift}}),\ f_c, t_{\text{shift}} \in \mathbb{R}, \tag{4.10}$$

which is a standard source signal for seismic calculations. In this case we choose $f_c = 25\,\mathrm{Hz}$ and $t_{\text{shift}} = 1.5/f_c$. The form of the signal is shown in Fig. 4.2. In all transmission experiments in this thesis the background values are $v_{\text{bg}} = 3500\,{}^{\mathrm{m}}/_{\mathrm{s}}$, $\varrho_{\text{bg}} = 2000\,{}^{\mathrm{kg}}/_{\mathrm{m}^3}$, and $\tau_{\text{bg}} = 0.0767$. For the parameters that model the frequency dependent behavior of the quality factor, we choose $L = 5$ and $\boldsymbol{\tau_l} = (0.3207\,\mathrm{s}, 0.0748\,\mathrm{s}, 0.0153\,\mathrm{s}, 0.0034\,\mathrm{s}, 0.0013\,\mathrm{s})$. Together with $\tau_{\text{p,bg}}$ from above this yields a background dissipation factor of $1/Q_{\text{bg}} = 1/15$. This is a typical value for sediments close to the seabed. In our inversions for v_{p} and τ_{p} we want to avoid the *inverse crime* (IC), that is, producing the observed data in synthetic examples with the same solver for the wave equation. In this case we use data that is produced by SOFI2D [Boh02]. There, the visco-acoustic wave equation in time domain is discretized by *finite differences* (FDTD) in space and time. Committing the inverse crime typically makes inversions easier, because the observed data contains the same structural discretization error as the data we try to fit it with [CK19]. Since sources are implemented differently (in dG and FDTD) each FDTD shot is scaled to match the maximal amplitude of the dG data. The rescaling is done with data that is produced with a higher degree of the ansatz functions. In Fig. 4.3 we can see the difference of the FDTD data and the data if we had produced it with the dG solver. Note that we used the same layout of the v_{p} and τ_{p} inversion in the publication [BFE+21]. Here, we used a finer spatial grid $h = 12.5$ compared to $h = 25$ for the inversion. Also, we included the sources and receivers in the error calculation and used a different data set to rescale the FDTD data.

For all inversions we calculate and plot errors of the iterates. By μ^{true} we always denote the true material distribution in all reconstructions. If we mean one specific material parameter, we put the parameter in the subscript. Further we define $\delta\mu_\bullet^k := \mu_\bullet^{\text{true}} - \mu_\bullet^k$ with $\bullet \in \{\varrho, v_{\text{p}}, \tau_{\text{p}}\}$. The subscript can also contain other, additional, information. The normalized error in the k-th iteration is defined by

$$\mathrm{err}_{2,k}^\bullet := \frac{\|\delta\mu_\bullet^k\|_{0,\Omega}}{\|\delta\mu_\bullet^0\|_{0,\Omega}} \ \text{and}\ \ \mathrm{err}_{\infty,k}^\bullet := \frac{\sup_{x\in\Omega} |\delta\mu_\bullet^k(x)|}{\sup_{x\in\Omega} |\delta\mu_\bullet^0(x)|} \ \text{for}\ \bullet \in \{\varrho, v_{\text{p}}, \tau_{\text{p}}\}\,. \tag{4.11}$$

We will only look at the L^∞-error for the transmission examples and only in this chapter. In most reconstructions the L^∞-error will not decrease. Therefore, we stick to the L^2-error to measure the error in the latter part of the thesis. The value of the full nonlinear residual at the point μ^k is

$$\mathrm{res}_{\text{full}}^k = \frac{\sqrt{\sum_{\xi=0}^{\Xi-1} \|\Phi_\xi(\mu^k) - \boldsymbol{s}_{\text{obs}}\|_{\boldsymbol{S}}^2}}{\sqrt{\sum_{\xi=0}^{\Xi-1} \|\Phi_\xi(\mu^0) - \boldsymbol{s}_{\text{obs}}\|_{\boldsymbol{S}}^2}}.$$

To calculate this quantity Ξ wave equations have to be solved since we never have access to $\Phi_\xi(\mu^k)$ for all shots at the same time in the calculation. To avoid this costly calculation but to still have access to a quantity that approximates the residual for a nonlinear iteration we use

$$\mathrm{res}^k = \frac{\sqrt{\sum_{\xi=0}^{\Xi-1} \|\tilde{\boldsymbol{r}}_{\text{end},\xi}^k\|_{\boldsymbol{S}}^2}}{\sqrt{\sum_{\xi=0}^{\Xi-1} \|\tilde{\boldsymbol{r}}_{\text{end},\xi}^0\|_{\boldsymbol{S}}^2}}, \tag{4.12}$$

where $\tilde{\boldsymbol{r}}_{\text{end},\xi}^k$, $k > 0$, is the final residual of the inner loop for shot ξ. For $k = 0$ we use the initial residual of the first inner loop. Calculating this quantity in the algorithm does not add any numerical cost. The

(a) v_p reconstruction

v_bg	$3500\,\mathrm{m/s}$
$v_{\Delta_1,\mathrm{p}}$	$3675\,\mathrm{m/s}$
$v_{\Delta_2,\mathrm{p}}$	$3850\,\mathrm{m/s}$
$v_{\Delta_3,\mathrm{p}}$	$3325\,\mathrm{m/s}$
ϱ_bg	$2000\,\mathrm{kg/m^3}$
τ_bg	0.0767

(b) τ_p reconstruction

τ_bg	0.0767
τ_{Δ_1}	0.3375
τ_{Δ_2}	0.0345
τ_{Δ_3}	0.1304
ϱ_bg	$2000\,\mathrm{kg/m^3}$
v_bg	$3500\,\mathrm{m/s}$

(c) ϱ reconstruction

ϱ_bg	$2000\,\mathrm{kg/m^3}$
ϱ_{Δ_1}	$2500\,\mathrm{kg/m^3}$
ϱ_{Δ_2}	$3000\,\mathrm{kg/m^3}$
ϱ_{Δ_3}	$1500\,\mathrm{kg/m^3}$
τ_bg	0.0767
v_bg	$3500\,\mathrm{m/s}$

Table 4.1: Parameters for the mono-parameter reconstructions for the transmission geometry.

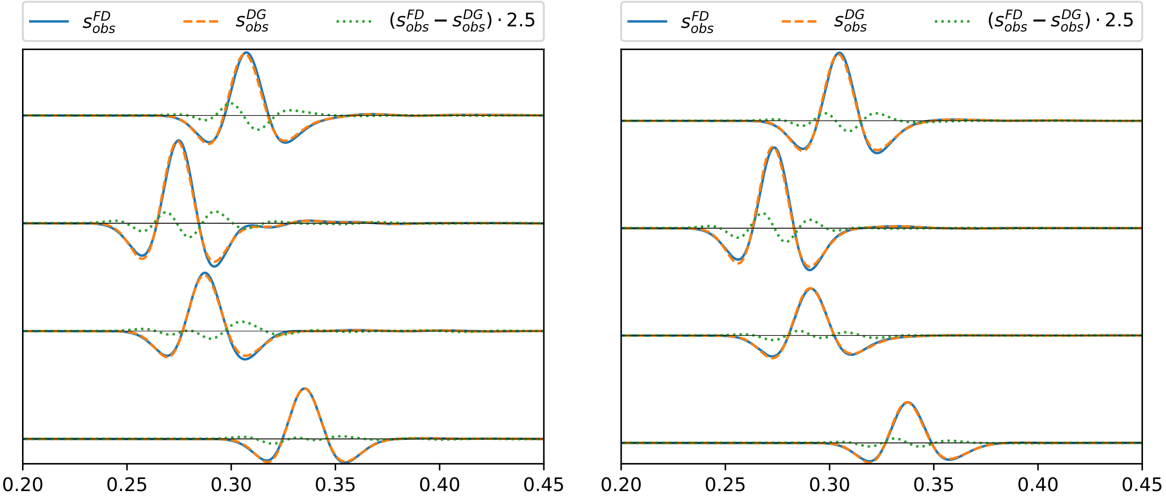


Figure 4.3: Comparison of the data we would invert if we committed inverse crime $s^\mathrm{DG}_\mathrm{obs}$ with the data from the FDTD solver (with adjusted amplitude) $s^\mathrm{FD}_\mathrm{obs}$. Seismograms of the receivers $6, 21, 46$, and 61 of shot 3. Left: v_p inversion. Right: τ_p inversion.

values of $\mathrm{res}^k_\mathrm{full}$ and res^k will deviate. The deviation depends on how large the linearization error is since res^k is a composition of residuals of the linearized problem. Further, materials are updated after each shot ξ therefore the value of $\tilde{\boldsymbol{r}}^k_{\mathrm{end},\tilde{\xi}}$ for $\tilde{\xi} = 0, \dots \xi$ was calculated for a different value of the material. The discrepancy can be gauged by $\|(\Phi_\xi(\mu^{k+1}) - \boldsymbol{s}_{\mathrm{obs},\xi}) - \tilde{\boldsymbol{r}}^k_{\mathrm{end},\xi}\|_{\boldsymbol{S}}$ In our experiments this norm was small, and therefore we use res^k in our considerations. Similarly to the relative error, for mono-parameter inversions we put the parameter we invert for in the superscript.

Inversion of v_p

In the inversion for v_p we choose the values in the Δ regions to be relative deviations of $5\%, 10\%$ and -5% of the background value. The values of the parameters can be found in Table 4.1. The top left of Fig. 4.10 shows a comparison of the seismogram from the FDTD solver and the seismogram of the initial value from the DG solver. The effect of the change in v_p materializes in the phase shift of the first arrival of the wave. All iterates of the inversion are depicted in Fig. 4.11. The final result of the inversion of v_p compared to the true value can be seen in Fig. 4.4. The inclusions are clearly visible and distinguishable against the artifacts around the source. In the error plots in Fig. 4.9 we can see that the L^2 error behaves

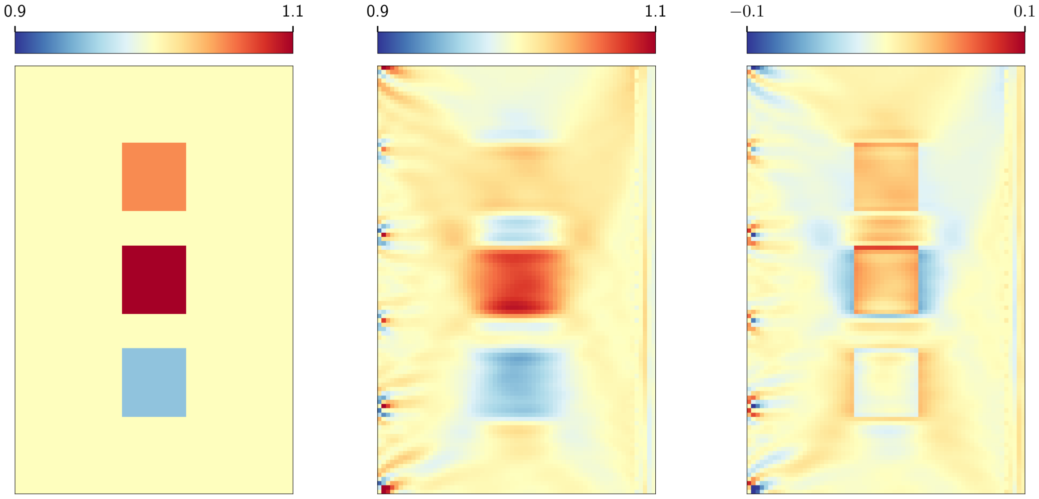


Figure 4.4: v_p mono-parameter inversion. Left: true value $\mu_{v_\mathrm{p}}^{\mathrm{true}}$. Middle: final reconstruction $\mu_{v_\mathrm{p}}^{10}$. Right: difference of the true value and the final reconstruction $\delta\mu_{v_\mathrm{p}}^{10}$. The values of the iterates are divided by $v_{\mathrm{p,bg}} = 3500\,\mathrm{m/s}$. Initial guess is constant $v_{\mathrm{p,bg}}$.

in a semi-convergent manner. It is the smallest in iteration 5 and increases after that. This is due to over fitting. After most of the meaningful data (that is, the part our solver can explain) is fitted, the algorithm tries to fit the difference of the solutions produced by the dG and the FDTD algorithms. This introduces artifacts around the source. These artifacts are responsible for the high values in the L^∞-error in Fig. 4.9: After 6 iterations the error is higher than it is in the beginning. Note that these results do not contradict our results from [BFE+21], where we saw a reduction of the error for all iterates. There we did not include the sources and receivers in the calculation of the error. Although the errors rise from the 5-th iteration, we can see in Fig. 4.8 that the residual $\mathrm{res}_\mathrm{p}^v$ still decreases monotonously showing the ill-posedness of the problem. To circumvent the problem of the increasing error, the iteration should be stopped adaptively instead of using a fixed number of 10 iterations (e.g. Hanke-Raus). Another measure that should reduce artifacts substantially is adding a source time inversion. This is an additional step that is widely used in geophysical applications [Pra99, GSFB14], where the true signal of the sources is usually not known and has to be approximated. In our case we also do not have access to the true signal (due to differing implementations) and therefore the artifacts could be reduced with such approximation.

Inversion of τ_p

For the mono-parameter inversion of τ_p we choose the values of τ_p. Such a way that we get values of the dissipation factor $(1/Q)_{\Delta_1} = 1/5, (1/Q)_{\Delta_2} = 1/10, (1/Q)_{\Delta_3} = 1/30$. The exact values of τ_p to attain that value are in Table 4.1. The values inside the inclusions $\Delta_1, \Delta_2, \Delta_3$ are chosen in such that there is a visible effect on the seismogram: The inclusions are small relative to the wavelength of the waves. Hence, we have to set high values of attenuation inside of them, because the amount of damping depends on the number of cycles a wave travels through the material. In Fig. 4.10 we can see the effect of the change in τ_p: There is a difference in amplitude but not in phase. Moreover, in Fig. 4.3 we can see that for the data produced by the FDTD solver the amplitudes do not match perfectly, but the phases are very

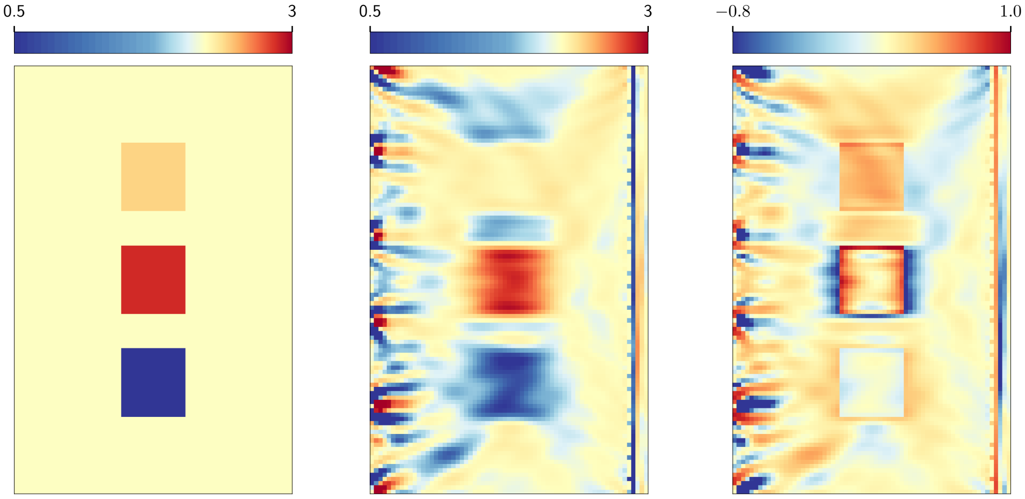


Figure 4.5: Dissipation $Q^{-1}(\tau_\mathrm{p})$ (see (4.13)) values for the τ_p mono-parameter inversion. Left: true value $\mu^\mathrm{true}_{1/\mathrm{Q}}$. Middle: final reconstruction $\mu^{10}_{1/Q}$. Right: difference of the real value with the final reconstruction $\delta\mu^{10}_{1/Q}$. The depicted values are divided by $1/Q_\mathrm{bg} = 1/15$. Initial guess is constant $1/Q_\mathrm{bg}$.

similar. Since the inversion for τ_p is mainly sensitive to the amplitudes, this introduces a lot of additional artifacts. We do not plot the auxiliary parameter τ_p, but rather the physical quantity dissipation with the formula (2.6) which yields

$$Q^{-1}(\tau_\mathrm{p}) = \frac{1 + \tau_\mathrm{p} \sum_{l=1}^{L} \frac{\omega_0 \tau_l}{1+\omega_0^2\tau_l^2}}{1 + \tau_\mathrm{p} \sum_{l=1}^{L} \frac{\omega_0^2 \tau_l^2}{1+\omega_0^2\tau_l^2}}, \text{ with } \omega_0 = 2\pi f_c \,. \tag{4.13}$$

All iterates of the inversion are depicted in Fig. 4.12. The final reconstruction of Q^{-1} can be seen in Fig. 4.5. In the reconstruction we can see that the perturbations are pronounced, but the reconstruction is riddled with artifacts especially around the source locations. Like we mentioned above: Since the inversion is very sensitive to amplitude changes, with source inversion the artifacts could be reduced even more than in the case of the v_p inversion. In Fig. 4.8 we can see that, just like in the v_p reconstruction, the residual decreases in a monotone way underlining the unreliability of the norm of the residual in this inversion.

Inversion of ϱ

Typically, ϱ is harder to reconstruct than the other parameters and typically requires additional care. This is especially true for multi-parameter reconstructions and requires a careful combination of the choice of the parameterization and modifications on the inversion algorithm. Some publications that deal with this topic are [KDNK$^+$12, JLM12, OGP$^+$13]. These considerations are beyond the scope of this thesis. Nevertheless, in the publications mentioned it is discussed that ϱ is hard to construct due to the following reasons: It has no effect on the phase and mainly on the amplitude of reflections and a weak effect on transmitting waves. However, we do want to present a mono-parameter inversion for our transmission example. Since rather realistic deviations from the background like, 5-10%, do not yield any results, we change the materials such that we have relative deviations of 25%, 50%, and -25%. In addition, we carry out this inversion with inverse crime to not make the reconstruction harder than it already is. In

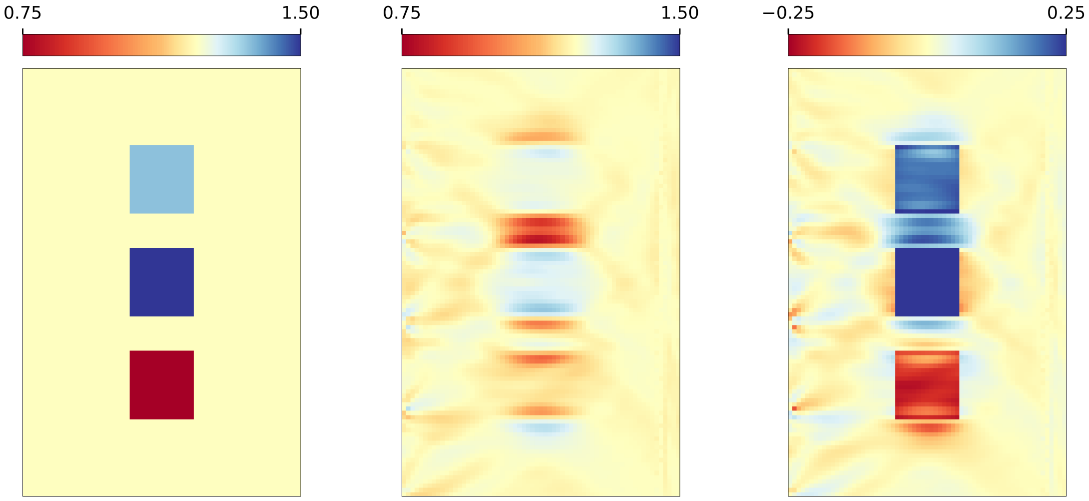


Figure 4.6: ϱ mono-parameter inversion. Left: true values $\mu_{v_\mathrm{p}}^{\mathrm{true}}$. Middle: final reconstruction μ_{ϱ}^{10}. Right: difference of the true value and the final reconstruction $\delta\mu_{\varrho}^{10}$. The depicted values are divided by $\varrho_{\mathrm{bg}} = 2000\,\mathrm{kg}/\mathrm{m}^3$. Initial guess is constant ϱ_{bg}.

Fig. 4.10, the bottom row shows the impact of the change in ϱ on the seismograms. The result of the inversion can be seen in Fig. 4.6. The norm of the residual decreases monotonously to a similar relative value like in the τ_p inversion see Fig. 4.8. Although we can see a decrease in error in Fig. 4.9, it is not very substantial, even though we commit the inverse crime. It is likely that we can observe only the boundaries of the inclusion due to the strong impact of ϱ on the reflection of waves. We add another reconstruction of ϱ: we place receivers in a box around $\Delta_1, \Delta_2, \Delta_3$. The corners are $\boldsymbol{r}_0 = \boldsymbol{r}_{192} = (1350\,\mathrm{m}, 370\,\mathrm{m})^\top$, $\boldsymbol{r}_{64} = (1350\,\mathrm{m}, 1630\,\mathrm{m})^\top$, $\boldsymbol{r}_{96} = (650\,\mathrm{m}, 1630\,\mathrm{m})^\top$, and $\boldsymbol{r}_{160} = (650\,\mathrm{m}, 370\,\mathrm{m})^\top$. In this experiment, we set the number of iterations to 30. In the reconstruction in Fig. 4.7 we can see that the algorithm now detects all edges of the inclusions. The inner part however is still not reconstructed very well. This shows that when using an unrealistic good measurement setup, we can improve the ϱ inversions.
In Fig. 4.8 we can see an example of how many inner loops are necessary to reach the stopping criterion.

All numerical experiments in this section were carried out on a cluster with 12 nodes. Every node has 20 Intel Xeon CPU E5-2609 v2 with a 2.50GHz clock rate processors and 90 GB of memory. We used two nodes for each calculation of the inversions and started the program with 32 processes. With this configuration, one mono-parameter inversion with 10 outer iterations takes about 24 h and the solution of one wave equation takes about 100 s.

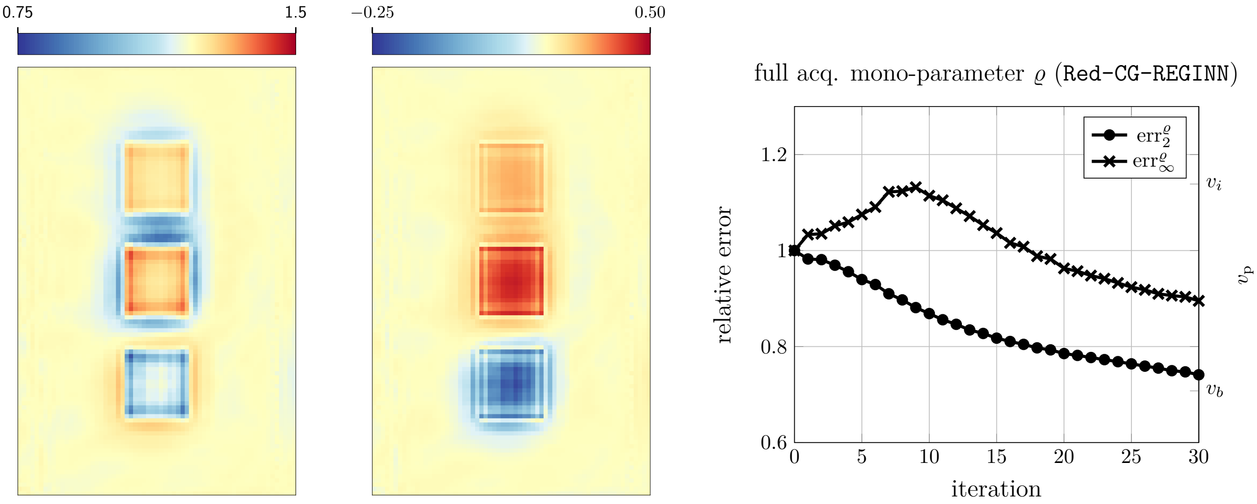


Figure 4.7: ϱ mono-parameter inversion with full acquisition (receivers on all sides). Left: final reconstruction μ_ϱ^{30}. Middle: difference of the real value with the final reconstruction $\delta\mu_\varrho^{30}$. The values of the iterates are divided by $\varrho_\text{bg} = 2000\,\text{kg}/\text{m}^3$ (yellow). Right: relative errors as defined in (4.11) for this inversion.

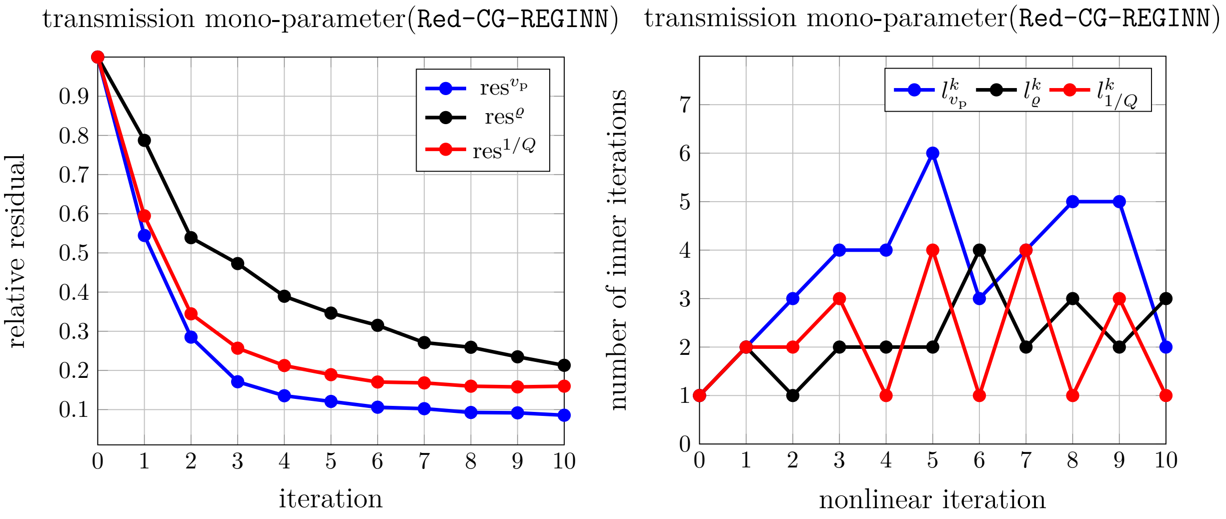


Figure 4.8: Left: relative residuals (as defined in (4.12)) of the mono-parameters experiments from Section 4.4.1. For all experiments they decrease monotonously, although the error does not. Right: number of iterations for the inner loop for shot $\xi = 3$.

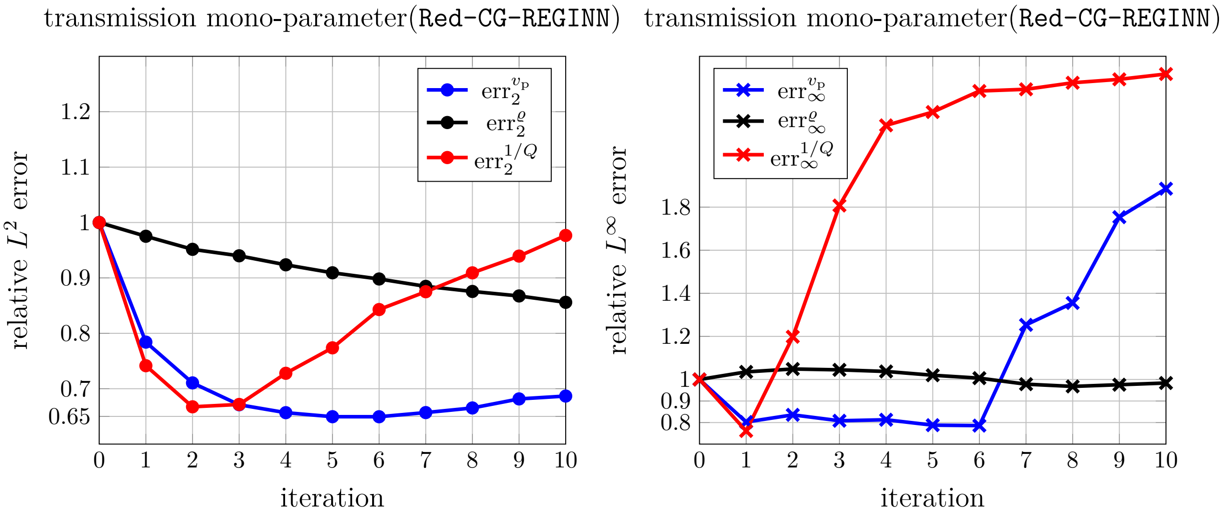


Figure 4.9: Relative errors (as defined in (4.11)) of the mono-parameter experiments from Section 4.4.1. Left: L^2-error. Right: L^∞-error

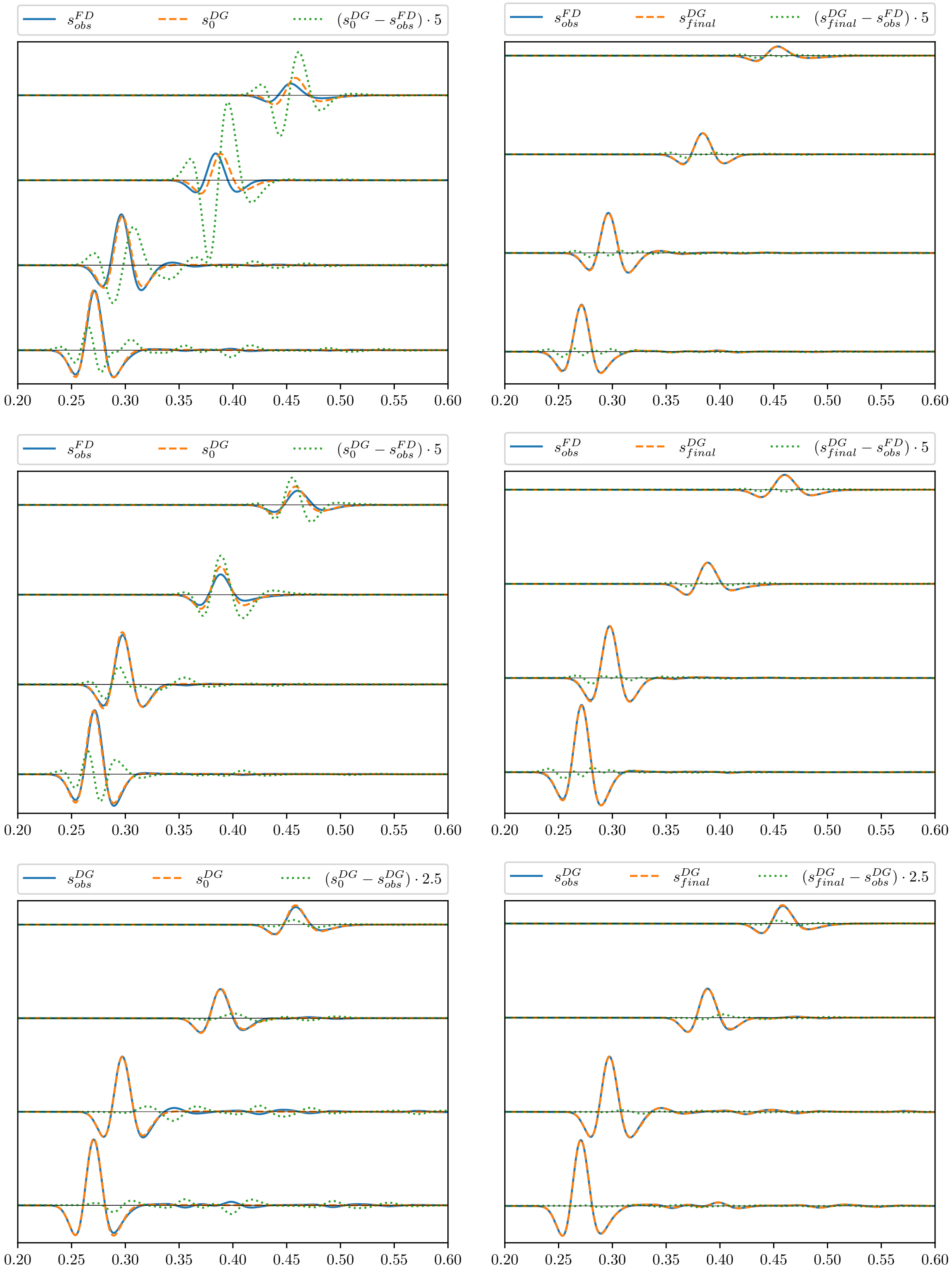


Figure 4.10: Seismograms of receivers $6, 21, 46$, and 61 of shot 0 for the mono-parameter transmission experiment. Left: comparison of (synthetic, adapted FD Data for $v_{\mathrm{p}}, \tau_{\mathrm{p}}$, and dG data for ϱ) observed data and data corresponding to the initial guess. Right: comparison of (synthetic, adapted FD Data for $v_{\mathrm{p}}, \tau_{\mathrm{p}}$, and dG data for ϱ) observed data and of final result of the reconstruction method. From top to bottom: $v_{\mathrm{p}}, \tau_{\mathrm{p}}, \varrho$.

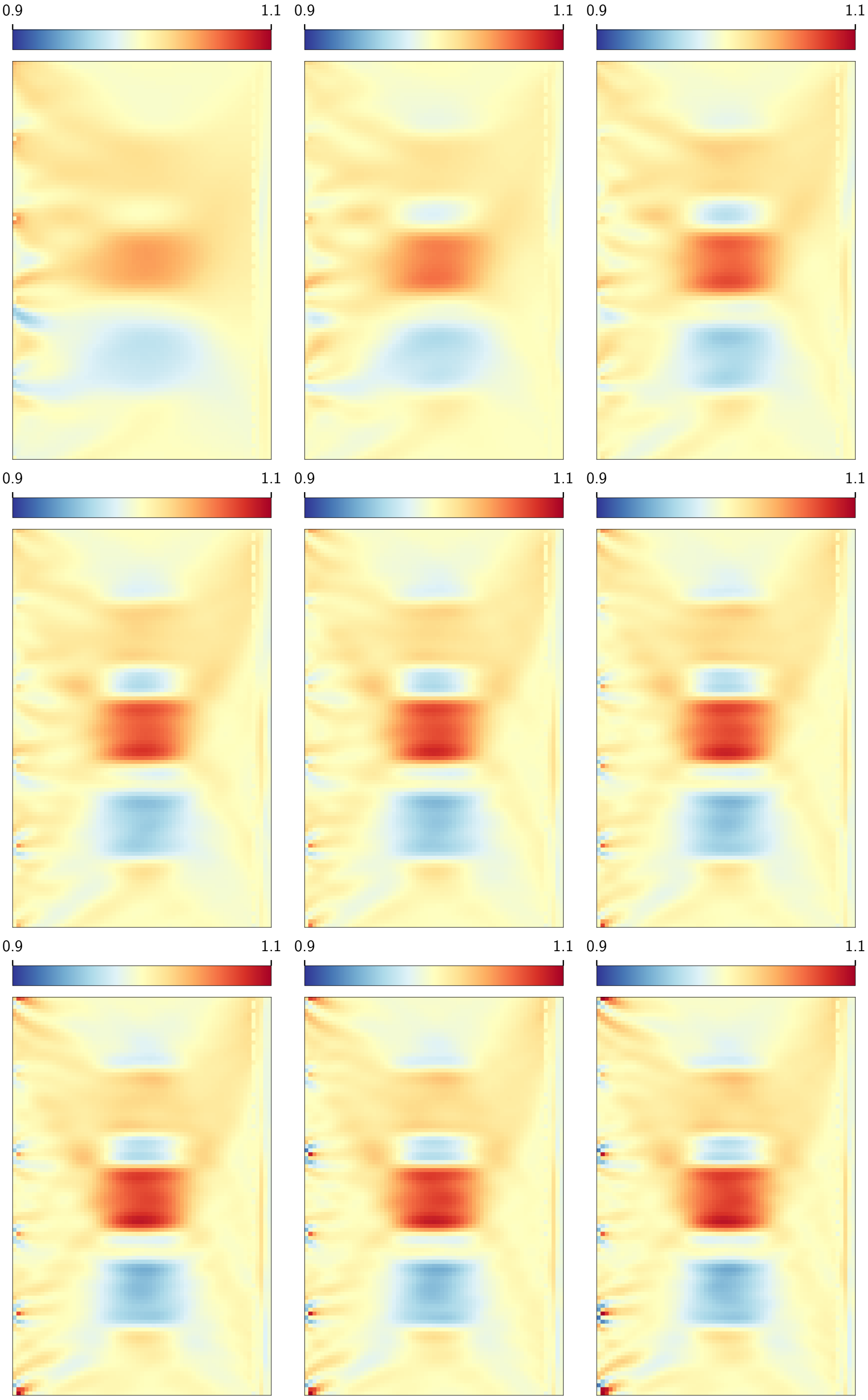


Figure 4.11: Iterates $\mu^k_{v_\mathrm{p}}$ for $k = 1, \ldots, 10$ of the v_p mono-parameter inversions with no inverse crime. The values of the iterates are divided by $v_{\mathrm{p,bg}} = 3500\,{}^\mathrm{m}\!/\mathrm{s}$.

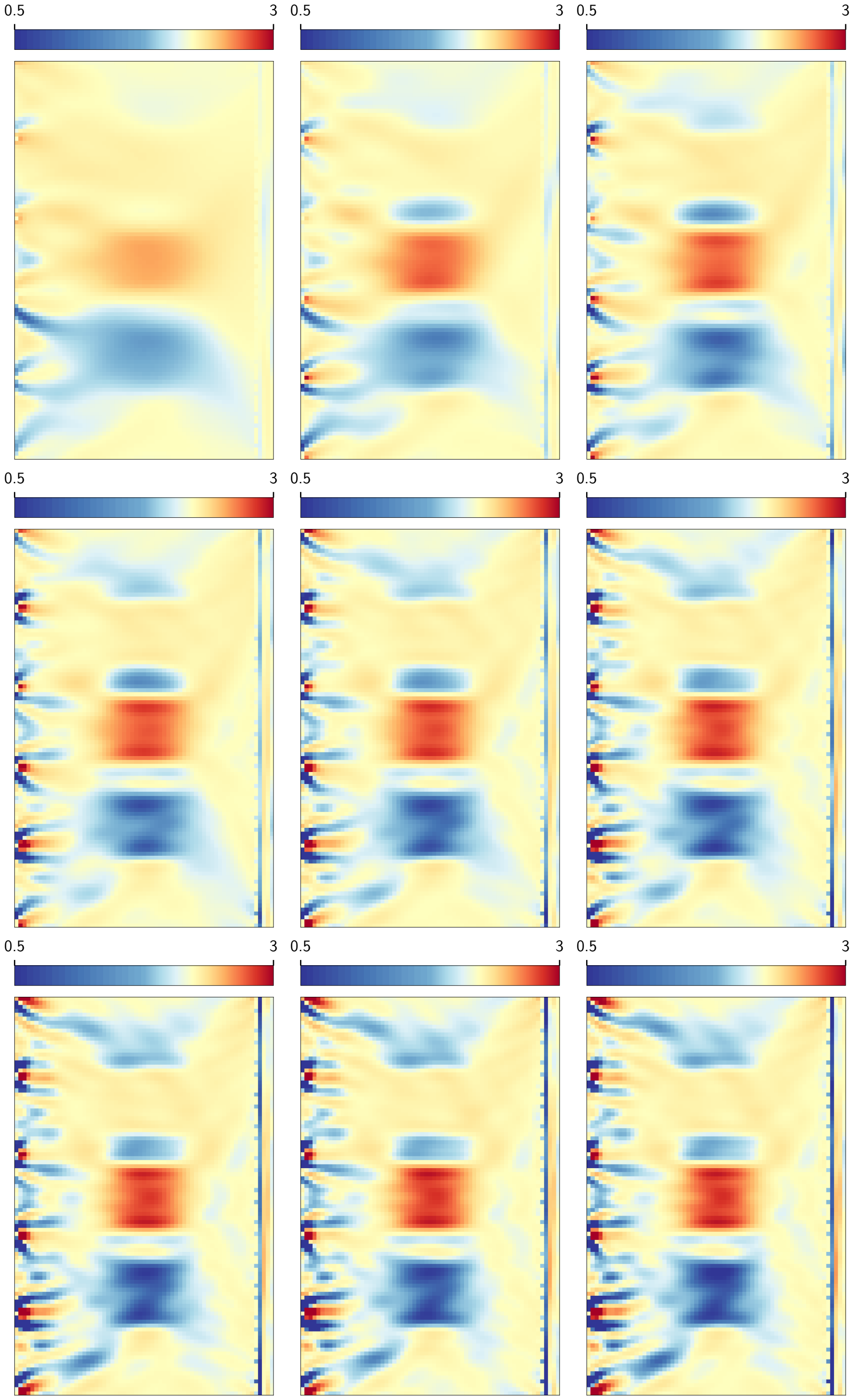


Figure 4.12: Dissipation $Q^{-1}(\tau_{\mathrm{p}})$ (see (4.13)) values for the iterates $\mu_{\tau_{\mathrm{p}}}^{k}$ for $k = 1, \ldots, 10$ of the τ_{p} mono-parameter inversions with no inverse crime. The values of the iterates are divided by $1/Q_{\mathrm{bg}} = 1/15$.

4.4.2 Marmousi

We test the algorithm with a more complex layered geometry. We use a part of the pressure wave velocity distribution called Marmousi2 (we will refer to this geometry as *Marmousi*) [MWM06] which is a very popular model to test inversion algorithms in geophysics. In Fig. 4.13 on the left we can see the true distribution of v_p. The Marmousi model is an example of a marine data acquisition and very different to the example from Section 4.4.1. On top there is a water layer with depth of 450 m. Below the water layer there are sediments with varying wave propagation speeds. In the water layer the propagation speed is set to 3500 $^\mathrm{m}/_\mathrm{s}$, the density to 1000 $^\mathrm{kg}/_\mathrm{m^3}$ and the dissipation to 10^{-6}. Below the water, ϱ is constant set to 2000 $^\mathrm{kg}/_\mathrm{m^3}$ and the dissipation to 1/60. In our case we do a mono-parameter inversion for v_p. For the source signal in time we use the integral

$$\tilde{r}(t) = \int_0^t r(s)\ \mathrm{d}s, \tag{4.14}$$

with r being the Ricker wavelet from (4.10) with central frequency $f_c = 9\,\mathrm{Hz}$ and $t_\mathrm{shift} = 0.5\,\mathrm{s}$. We choose $L = 3$ and $\boldsymbol{\tau_l} = (1.0540\,\mathrm{s}, 0.0825\,\mathrm{s}, 0.0084\,\mathrm{s})$ and $\tau_\mathrm{p} = 0.0001$ in the water layer which results in a dissipation of 0.000625. To attain the dissipation of 1/60 in the rest of the model we set $\tau_\mathrm{p} = 0.0274$. We set $v_\mathrm{p,min} = 1500\,^\mathrm{m}/_\mathrm{s}$ and $v_\mathrm{p,max} = 4000\,^\mathrm{m}/_\mathrm{s}$ in the true model. The combination of central frequency of the source, number of damping parameters $\boldsymbol{\tau_l}$, maximal and minimal values for v_p $v_\mathrm{p,min}$, $v_\mathrm{p,max}$ and dissipation values is taken from the Marmousi2 reconstruction in [Kur12]. The computational domain is $\Omega = (0, 8000)\mathrm{m} \times (0, 3000)\mathrm{m}$. Along the top boundary, we prescribe Dirichlet boundary conditions such that the sum of all pressure components is zero on the boundary. In geophysics this is called *free surface boundary*. On the other three boundaries we extend the domain by 600 m which yields $\Omega_E = (-600, 8600)\mathrm{m} \times (0, 3600)\mathrm{m}$. In the absorbing layer $\Omega_E \setminus \Omega$ we use the following parameters in our Marmousi calculations: $Q_b = 1.1,\ v_{\mathrm{p},t} = 0.85 \cdot v_{\mathrm{p},i},\ v_{\mathrm{p},b} = 0.1 \cdot v_{\mathrm{p},i},\ s_t = 0.3 \cdot 600\,\mathrm{m}$ as explained in Section 3.5. The computational mesh has the width $h = 25$ and we use a polynomial degree of $k = 1$. The wave equations for the forward and adjoint problem are solved with the exponential midpoint rule with end time $T = 5.88\,\mathrm{s}$ and $\triangle t = 0.0056\,\mathrm{s}$. The number of spatial degrees of freedom is $\mathrm{dof}_h = 1618176$. We place $\Xi = 25$ equally distanced sources with equal distance on a line between $\boldsymbol{x}_{\mathrm{src},0} = (100.5\,\mathrm{m}, 20.0\,\mathrm{m})^\top$ and $\boldsymbol{x}_{\mathrm{src},24} = (7899.5\,\mathrm{m}, 20.0\,\mathrm{m})^\top$ close to the surface of the water. Further, $R = 350$ equally distanced receivers are placed on a line close to the seabed at starting at $\boldsymbol{r}_0 = (0.5\,\mathrm{m}, 435.0\,\mathrm{m})^\top$ and ending at $\boldsymbol{r}_{349} = (7999.5\,\mathrm{m}, 435.0\,\mathrm{m})^\top$. Moreover, we choose $\gamma = 0.95$ in (4.7), which is higher than in the transmission experiment. Increasing γ makes the accuracy, to which the inner iteration is solved, increase slower. Therefore, it decreases the number of inner iterations. Additionally, we set a maximal number of iterations for the CG algorithm to $l_\mathrm{max,glob} = 15$. Eventually, we set the number of outer iterations to 10 as termination criterion in Algorithm 4.2 and produce the data with the same solver (dG) we use for the inversion. We commit inverse crime in all Marmousi experiments.

If we just apply Red-CG-REGINN without additional measures, the inversion would fail. Therefore, we add some techniques that are widely used in the geophysics community to make the inversion possible in this geometry with this algorithm.

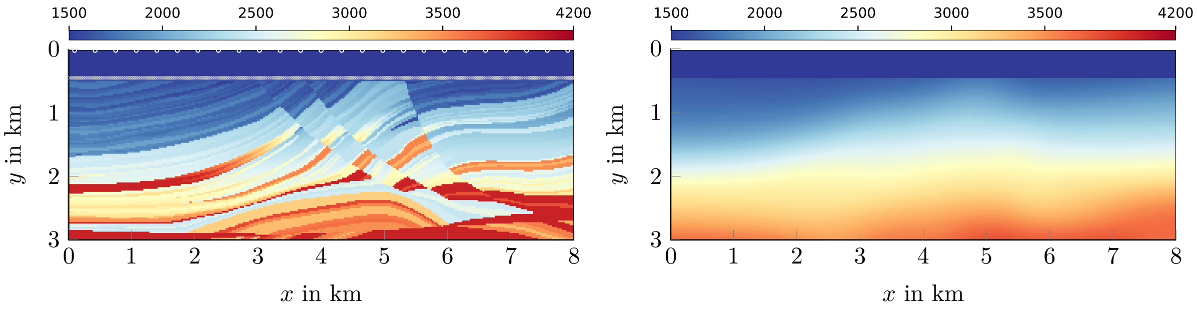


Figure 4.13: Left: ground truth $\mu_{v_\mathrm{p}}^{\mathrm{true}}$ of v_p (in m/s) in the Marmousi model, white circles mark source locations, white crosses mark receiver locations. Right: initial guess $\mu_{v_\mathrm{p}}^{0}$ for the v_p inversion.

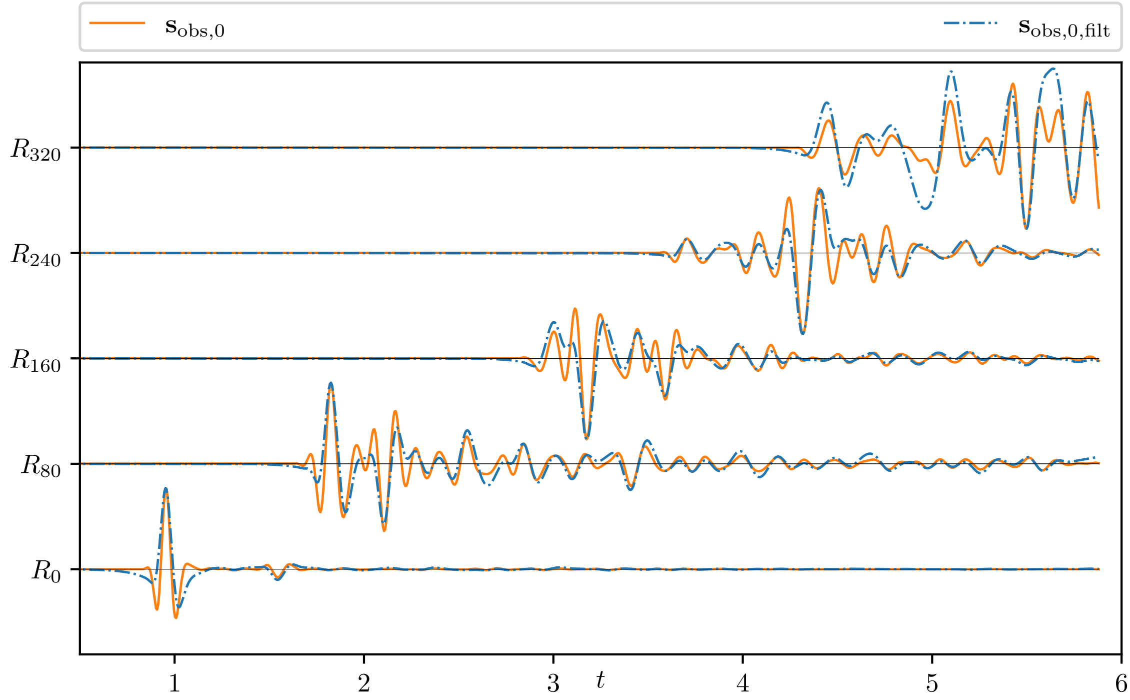
Frequency filtering

We add a well-known ansatz from the geophysics community. This technique is described in [BSZC95]. It consists of an algorithm with multiple stages, where we low-pass filter the data and the source signal and include more frequencies in each stage. This is done to reduce the number of local minima and is widely used to improve the convergence behavior in FWI algorithms [VO09]. Now, choose $\boldsymbol{F} = (f_0, f_1, \ldots, f_{\nu_{\max}-1})^\top \in \mathbb{R}^{\nu_{\max}} \cup \{\infty\}$ with increasing f_i which yields an algorithm with $\nu_{\max}$ stages. Let $\boldsymbol{s}_{\mathrm{obs}}$ be the measured data and $\phi \in C_c^\infty(\mathbb{R})$ a the source signal in time (in the case of the Marmousi inversion this is $\tilde{r}$). Now, for each $f_i \in \boldsymbol{F}$ we apply a Butterworth[2] low-pass filter with peak frequency f_i to each component of $\boldsymbol{s}_{\mathrm{obs}}$ and to ϕ and run the inversion algorithm `Red-CG-REGINN` with these. With the result of the i-th stage we start the same process with the next $f_{i+1} \in \boldsymbol{F}$. The resulting algorithm can be found in Algorithm 4.3. If the last frequency $f_{\nu_{\max}-1}$ is set to ∞, no filter is applied in that stage. Naively, one can think of such a filtering process in terms of Fourier transforms. We can Fourier transform the data/signal, then decrease the magnitude of the coefficients higher f_i in frequency domain, and then apply an inverse Fourier transform to get a time dependent signal again. In reality using the Fourier transform in that way is not helpful, since local changes in the Fourier transform can lead to global changes in the signal and distort the signal heavily. Therefore, a careful filter design is necessary. We use an infinite impulse response Butterworth low-pass filter which is a standard variant of a digital filter design as described for example in [Lyo97]. We use the implementation of this filter from the free and open source library DSPFilters (https://github.com/vinniefalco/DSPFilters). A filtered source signal for the Marmousi inversion can be seen in Fig. 4.16 and filtered measured data can be found in Fig. 4.14.

Pseudo-Hessian preconditioner

In addition to the filtering technique, we add a pseudo-Hessian preconditioner. The idea was first introduced by Shin [SJM01] in the frequency domain and developed further in several publications. We use the version adapted to time domain and the possibility to include multiple parameters in [YBM+18]. This kind of preconditioner assumes zero-offset geometry, that is, source and receiver are at the same

[2] This is a filter that does not change the phase of the signal. Frequencies above the peak frequency f_{peak} are damped by 24dB($10^{2.4} \approx 254$) per octave, that is, doubling of the frequency. This can be expressed by the damping factor $D(f) \approx 10^{-2.4\cdot(f-f_{\mathrm{peak}})/f_{\mathrm{peak}}}, f > f_{\mathrm{peak}}$. Note that this is an idealized description but should suffice in this context.



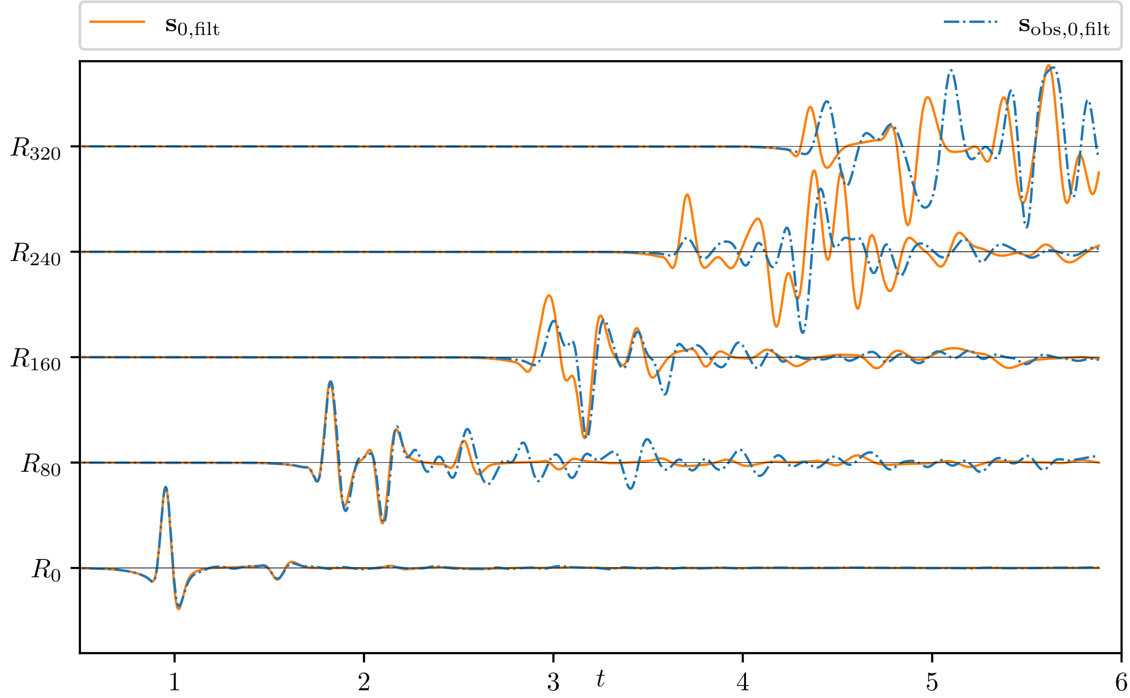


Figure 4.14: Top: unfiltered observed (simulated, dG) data and low-pass (1.7Hz, Butterworth order 2) filtered observed (simulated, dG) data. Bottom: seismogram for the initial guess. The seismograms of the first shot at $\boldsymbol{x}_{\mathrm{src},0} = (20, 100.5)$.

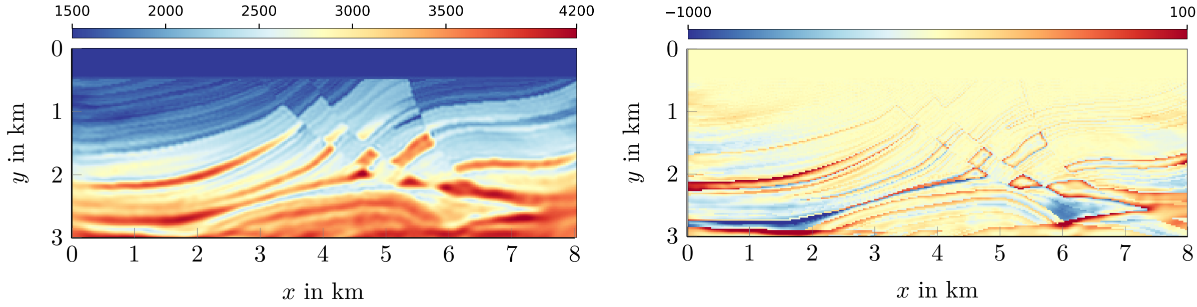
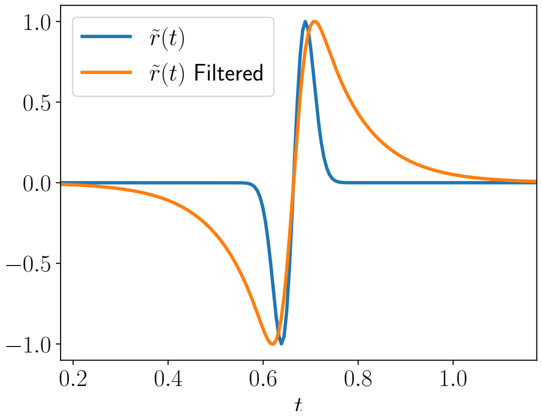


Figure 4.15: Left: Result $\mu_{v_\mathrm{p}}^{\infty,10}$ after the 10 iterations of the last stage without preconditioner and no frequency filtering for the Marmousi experiment (v_p values in $^\mathrm{m}/_\mathrm{s}$). Right: Difference $\delta\mu_{v_\mathrm{p}}^{\infty,10}$ of the true model and the final inversion.



T	5.88
$\triangle t$	0.0056
L	3
k	1
h	25
dof_h	1618176

Figure 4.16: Left: integrated Ricker wavelet $\tilde{r}$ and its low-pass (1.7Hz) filtered version. Right: information about the discretization for the Marmousi geometry.

position. While this is not entirely true in our Marmousi model, this layout is still close enough. For a transmission-like problem layout this preconditioner should not be used. Parameter values closer to the source and receivers are typically more influential on the seismogram than values deeper in the model. The preconditioner amplifies values further away from the sources/receivers and dampens values close to them . This makes inversions of deeper layers inside the model possible. Although we assumed in this section that we only invert for one parameter, we formulate the Pseudo-Hessian preconditioner in a setting where we invert for N_inv parameters. With the definition of $\boldsymbol{\mathfrak{L}}'_{\mu_\xi^k,\boldsymbol{u}}$ from Theorem 5, in the k-th iteration, for shot ξ and for every cell K we calculate

$$\boldsymbol{G}(i,j)_K = \int_0^T \Big(\boldsymbol{\mathfrak{L}}'_{\mu_\xi^k,\mathcal{F}_\xi(\mu_\xi^k)}[\boldsymbol{e}_i], \mathcal{F}_\xi(\mu_\xi^k)(t)\Big)_{0,K} \Big(\boldsymbol{\mathfrak{L}}'_{\mu_\xi^k,\mathcal{F}_\xi(\mu_\xi^k)}[\boldsymbol{e}_j], \mathcal{F}_\xi(\mu_\xi^k)(t)\Big)_{0,K} \,\mathrm{d}t \;\forall i,j = 1,\dots,N_\mathrm{inv}, \tag{4.15}$$

where $\boldsymbol{G}(i,j)_K$ denotes the $(i,j)-$th entry of the block matrix corresponding to the cell K and $\boldsymbol{e}_i$ are vectors that are constant one on the cell K for one parameter and zero for the others. In the mono-parameter case $\boldsymbol{G}$ is a diagonal matrix. For more than one parameter this yields a block diagonal matrix. In both cases the inverse is easily calculated. In (4.15) the value of $\Big(\boldsymbol{\mathfrak{L}}'_{\mu_\xi^k,\mathcal{F}_\xi(\mu_\xi^k)}[\boldsymbol{e}_i], \mathcal{F}_\xi(\mu_\xi^k)(t)\Big)_{0,K}$ is a

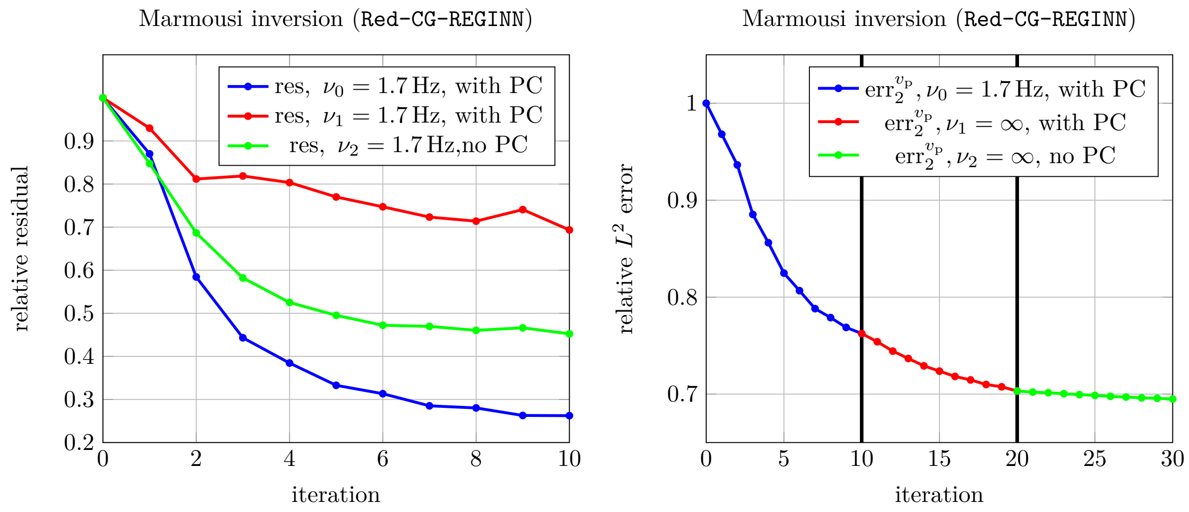


Figure 4.17: Left: relative (to the start of each stage) residual as defined in (4.12). Right: relative L^2-error as defined in (4.11) of the v_p Marmousi inversion. The vertical lines indicate the different stages of the algorithm. The value for $k = 10$ (left) and $k = 20$ (right) is depicted in Fig. 4.18 and $k = 30$ in Fig. 4.15.

weighted integral over the product of forward wave fields. As a result, the preconditioner exhibits larger values in close proximity to the source and smaller at increasing distances. This property allows for a better reconstruction of the deeper layers as we invert $\boldsymbol{G}$ on application and therefore these values are increased. Since the block matrices can become almost singular, this can introduce additional artifacts. To suppress artifacts and stabilize the inversion we add a small $\varepsilon > 0$ as a regularization to the diagonal of $\boldsymbol{G}$ before we invert it:

$$\boldsymbol{G}_\varepsilon := \boldsymbol{G} + \varepsilon \boldsymbol{I}. \tag{4.16}$$

Inversion of v_p

For our v_p inversion, we choose $\nu_{\max} = 2$ and $\boldsymbol{F} = (1.7, \infty)^\top$ for the frequency filtering technique. We apply a zero-phase Butterworth filter of order 2. Note that without frequency filtering, the inversion with

Algorithm 4.3 Red-CG-REGINN with frequency filtering,

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess;   $\boldsymbol{s}_\mathrm{obs} \in \boldsymbol{S}^\Xi$ % seismograms
Ensure: $\mu^\nu \in \boldsymbol{P}_h$
$\mu^\mathrm{init} \leftarrow \mu^0$
for $\nu$ in $\boldsymbol{F}$ do
    $\boldsymbol{s}_{\mathrm{obs},\nu} \leftarrow \mathrm{filter}(\nu, \boldsymbol{s}_\mathrm{obs})$ % apply filtering to the data
    $\mu^\nu \leftarrow$ Red-CG-REGINN($\mu^\mathrm{init}, \boldsymbol{s}_{\mathrm{obs},\nu}, \nu$) % start the inversion with source signal and the data filtered to frequency $\nu$;
    $\mu^\mathrm{init} \leftarrow \mu^\nu$
end for
return $\mu^\nu$
```

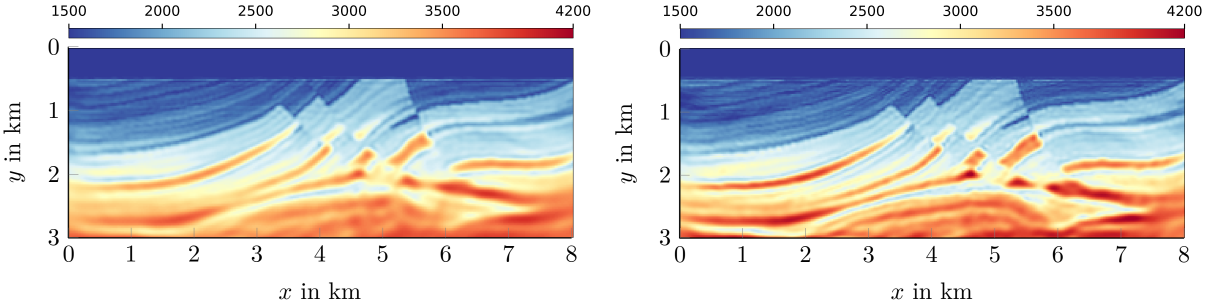


Figure 4.18: Left: result $\mu^{1.7,10}_{v_\mathrm{p},\mathrm{PC}}$ after 10 iterations of the stage with filtering to 1.7 Hz. Right: result $\mu^{\infty,10}_{v_\mathrm{p},\mathrm{PC}}$ after 10 iterations of the stage with no filtering. In both inversions the Pseudo-Hessian preconditioner is applied. The v_p values are in m/s.

`Red-CG-REGINN` for the Marmousi model fails. Further, to speed up convergence, we apply the regularized preconditioner $\boldsymbol{G}_\varepsilon$ from (4.16) with $\varepsilon = 10^{-5}$ in the first stage ($\nu = 1.7$) of the frequency filtering and with $\varepsilon = 10^{-3}$ for the second stage ($\nu = \infty$). We call the k-th iterate in these stages $\mu^{1.7,k}_{v_\mathrm{p},\mathrm{PC}}$, $\mu^{\infty,k}_{v_\mathrm{p},\mathrm{PC}}$. Since the preconditioner can promote artifacts, we add another stage of inversion with no frequency filtering and no preconditioner to smoothen the final result we call the iterates $\mu^{\infty,k}_{v_\mathrm{p}}$. In all stages we set a fixed number of 10 iterations as stopping rule in Algorithm 4.2. Moreover, we assume that the size of the water layer is known and therefore restrict updates to the values below $y = 450\,\mathrm{m}$. As initial value we use a blurred version of the true value, shown in Fig. 4.13 on the right. Further, we add a box constraint: Whenever v_p is updated and its value would be greater than 4500 m/s or smaller than 1500 m/s we set it to 4500 m/s resp. 1500 m/s. In Fig. 4.14, we present a plot of the observed (simulated,dG) data along with its filtered version. In addition, we show the seismograms of the simulated data for the initial guess along with the filtered version of the observed (simulated,dG) data. The unfiltered and filtered signal is shown in Fig. 4.16. In Fig. 4.15 we can see the result of the final stage of the inversion. The upper part of the model (above $y = 2\,\mathrm{km}$) is reconstructed well, the main features of the model are clearly visible. With increasing depth, however, the quality of the reconstruction decreases in general and especially on the left side of the model. This is not a problem of our algorithm or the model itself, but is rather due to bad illumination. We can see in the middle part of the model that the reconstruction is rather good, but on the left side the model is curved in such a way that the layers reflect towards the left boundary and therefore are not captured well with our receiver setup. In Fig. 4.17 we can see the evolution of the error which is decreasing in a monotone way and therefore showing the viability of `Red-CG-REGINN` for this setup. In Fig. 4.18 the intermediate results of the first two stages are depicted. We can see the result of the frequency filtering in comparison with the result of the second stage: the edges are a little more blurry and some of the inclusions are not as well resolved as the after the inversion with no filter. This is typical for frequency filtered inversions: the resolution typically depends on the wavelengths of the data, and if we reduce the frequencies of the data, we also filter the smaller wavelengths and get a lower resolution. In Fig. 4.19 the difference of one stage to the next one is shown, while there are changes in the whole domain from the first to the second stage, the last stage really just removes the oscillations in the upper part of the model. Over the whole inversion, the residuals are reduced monotonously as we can see in Fig. 4.17. Note that the absolute value increases after every filtering step since we include more data.

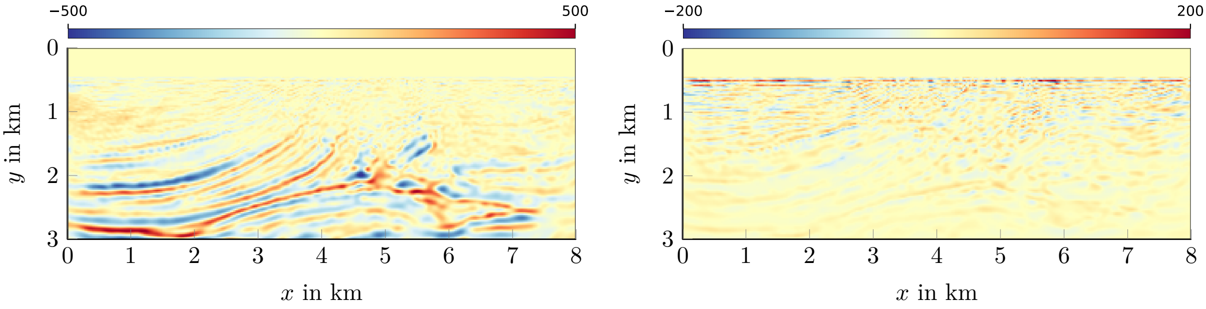


Figure 4.19: Left: difference of the results of the filtered and the first unfiltered stage $\mu^{1.7,10}_{v_\mathrm{p},\mathrm{PC}} - \mu^{\infty,10}_{v_\mathrm{p},\mathrm{PC}}$. Right: difference of the two unfiltered with and without preconditioning $\mu^{\infty,10}_{v_\mathrm{p},\mathrm{PC}} - \mu^{\infty,10}_{v_\mathrm{p}}$. The v_p values are in m/s.

4.4.3 Reconstructions with noise

A typical problem in applications is that the measured data is noisy. This poses a problem in ill-posed problems since they tend to be especially sensitive to noise. The algorithm CG-REGINN is shown to be a regularization algorithm and hence especially well-suited for problems of this kind. We want to see how the Red-CG-REGINN behaves for the already known Marmousi example, so we add artificial Gaussian noise to our synthetically produced observed data to simulate this more realistic scenario. The pseudo-random numbers are generated by the function std::mt19937 from the standard library of C++. This is a realization of the Mersenne-Twister algorithm [MN98]. To avoid correlated noise, we carefully choose the seeds for the pseudo-random number generator differently for each core and trace of the seismogram. Typically, the level of noise given by the *signal-to-noise ratio* (SNR). Assume that any signal can be decomposed into useful data and useless data (noise). Then, the SNR is given by

$$\mathrm{SNR} = \frac{\text{power of useful data}}{\text{power of useless data}}. \tag{4.17}$$

Now, let $\overline{\boldsymbol{s}} \in \boldsymbol{S}$ be a seismogram filled with normally distributed random numbers with expected value 0 and variance 1. Then, we obtain a seismogram with a desired SNR-value (which is typically given in the logarithmic unit dB, which we do not use here) by

$$\boldsymbol{s}^\delta_\mathrm{obs} = \boldsymbol{s}_\mathrm{obs} + \frac{\|\boldsymbol{s}_\mathrm{obs}\|_{\boldsymbol{S}}}{\mathrm{SNR}} \cdot \frac{\overline{\boldsymbol{s}}}{\|\overline{\boldsymbol{s}}\|_{\boldsymbol{S}}}\,.$$

Now, we set different levels of $\mathrm{SNR} = 0.1, 1, 10$. A value of 10 means that the signal content is 10 times higher than the noise level, which is moderate noise. On the other hand $\mathrm{SNR} = 0.1$ means that the noise level is higher than the signal content. The SNR is calculated on the whole seismogram and additive which leads to individual SNR ratios for each receiver. Receivers closer to the source tend to have a higher ratio and receivers further away have a lower ratio due to the amplitude loss with distance. This effect can be seen in the trace-wise normalized seismogram in Fig. 4.21. The 'true' data is barely visible for the 80-th receiver in case of $\mathrm{SNR} = 1$. A typical method in geophysics to reduce the influence of noise is setting parts of the seismograms to zero (called muting), where we know or can see in the data, that the source has not arrived yet and is often done by hand. For example, we could mute the 80-th receiver in the $\mathrm{SNR} = 10$ case until $\approx 3.6\,\mathrm{s}$. The individual SNR values for shots $0, 12$ and 18 are depicted in Fig. 4.22. Close to the source position, the SNR values are higher and decrease with increasing distance.

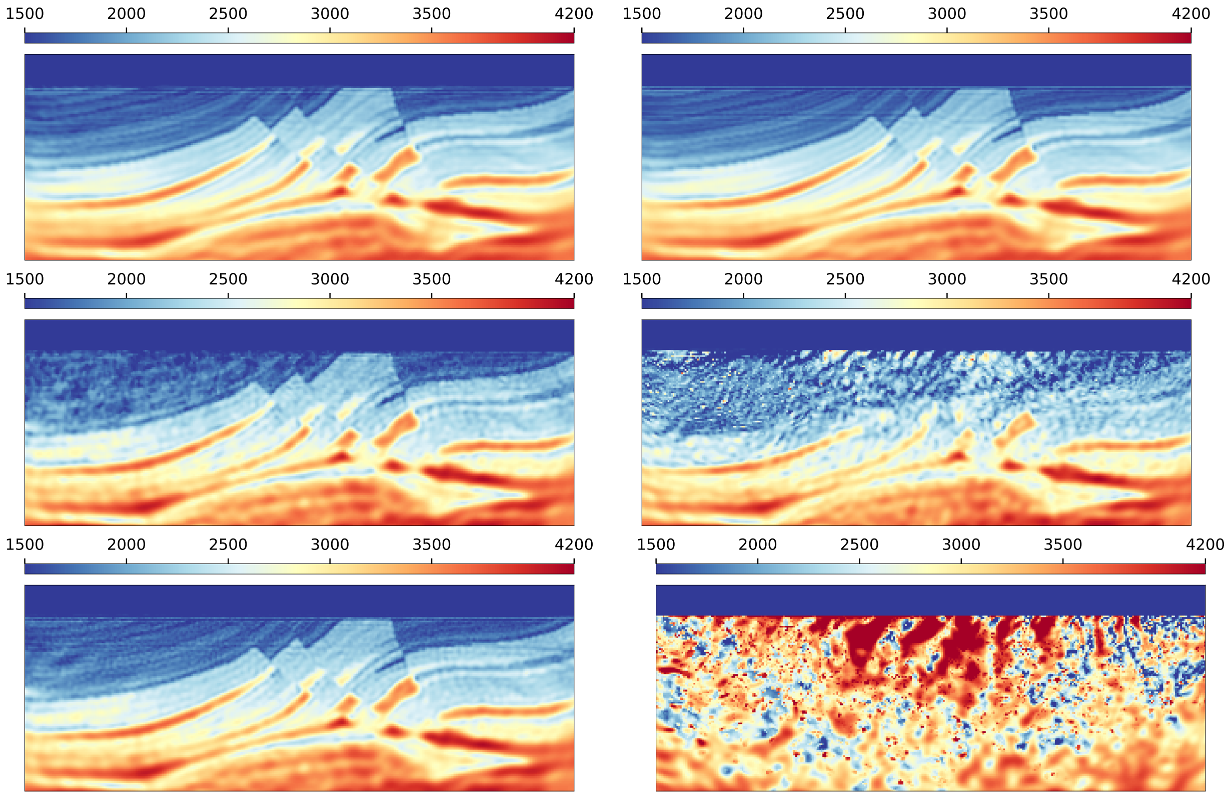


Figure 4.20: Reconstructions of Marmousi with noisy data. Top row: 10-th iterate $\mu_{v_{\mathrm{p}}}^{10}$ for SNR $= 10$. Left: first stage. Right: second stage. Middle row: 10-th iterate $\mu_{v_{\mathrm{p}}}^{10}$ for SNR $= 1$. Left: first stage. Right: second stage. Bottom row: left: best result L^2 error-wise for SNR $= 1$ $\mu_{v_{\mathrm{p}},\mathrm{PC}}^{\infty,5}$, see Fig. 4.23. Right: result for SNR $= 0.1$.

In Fig. 4.20, final and intermediate results of the reconstruction with noise are shown. In the top row the result after the first stage (left) of frequency filtering and after the second (right) are shown for the case of SNR $= 10$ (both stages with fixed of 10 iterations). These results are very similar to the reconstruction without noise in Fig. 4.18. The results for SNR $= 1$ are shown in the second row, but the final result contains a large number of artifacts. However, the result after only 5 iterations, shown in the bottom row on the left, is better than the final after 10 iterations, as shown in the error plot in Fig. 4.23 on the left. Note that the norms of the residuals during in the reconstruction, as depicted in Fig. 4.23 to the right. Thus, the increase in errors could have been avoided by using backtracking (that is, reducing the step size until the residual decreases) or terminating the algorithm when the residuals increase. This indicates the need for careful handling of noise. The first stage's outcome with SNR $= 0.1$ is displayed on the bottom right of Fig. 4.20, which has no similarity to the true model. In total, this demonstrates that this also shows that a substation part of the structure can be reliably extracted from the measurements with noise-corrupted signals for reasonable noise levels (SNR $= 10, 1$).

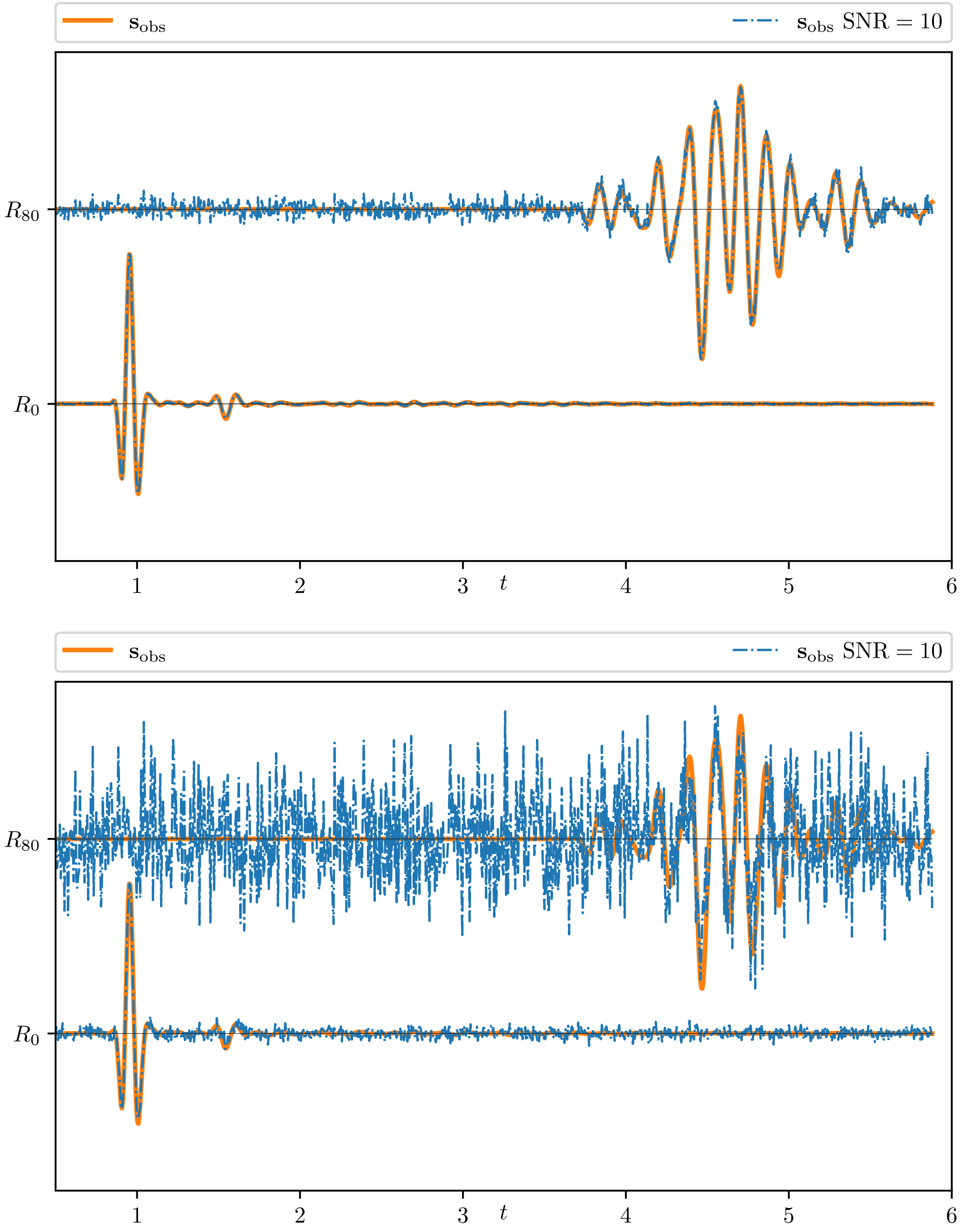


Figure 4.21: Values of Receiver 0 and 80 in the Marmousi experiment for shot 0, comparison of noisy and non-noisy data, each measurement is normalized.

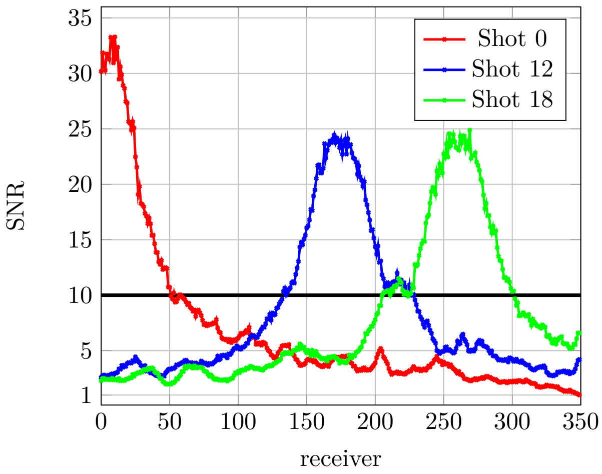


Figure 4.22: Signal-to-noise ratio evaluated receiver-wise for SNR $= 10$ for the shots $0, 12, 18$. For SNR $= 1$ and SNR $= 0.1$ qualitatively the curves look the same but the values of the SNRs are scaled by 0.1 and 0.01.

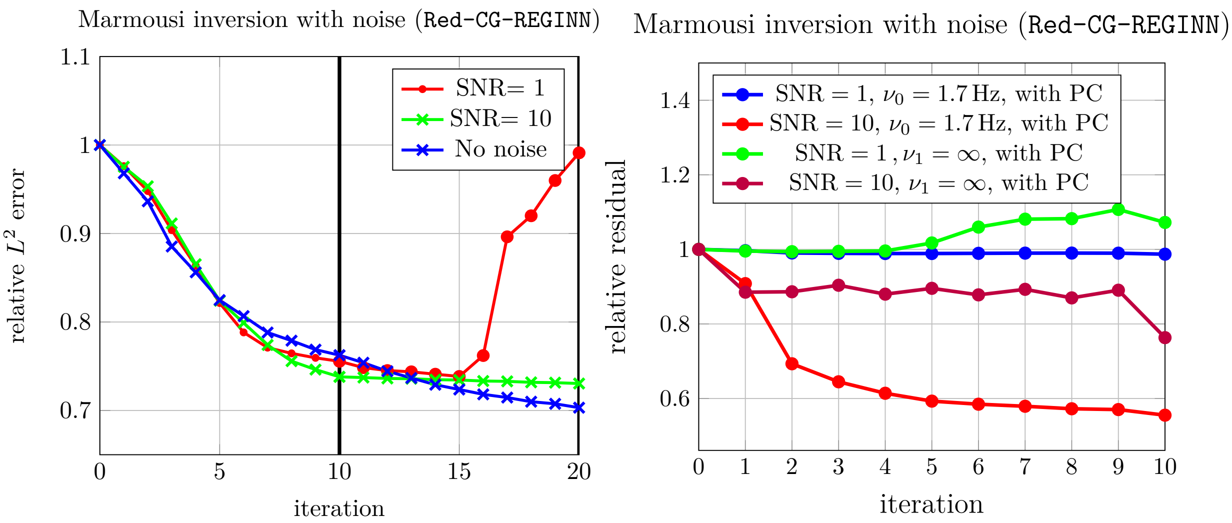


Figure 4.23: Left: relative L^2-error as defined in (4.11) of the Marmousi v_p inversion with noise. The vertical line indicates the change from the first filtered 1.7 Hz stage to the unfiltered stage. The inversion without noise is added for comparison. Results are shown in Fig. 4.20. Right: relative residuals, normalized to the first value of each stage.

4.4.4 Artifacts

Theory

Artifacts are typically present in reconstructions with the reduced formulation around sources and receivers, even when inverse crime is committed. According to Fichtner [Fic11] (quasi) singular values are to be expected in the gradient. Here, attempt to put this behavior into formulas, so we can describe the occurrence of the singularities with a better mathematical foundation and understanding. To this end we examine the gradient for v_p in the special case of constant parameters and no damping. From the formulas in Lemma 2 we know that in this case the gradient is proportional to $(\partial_t \boldsymbol{u}_{p_0}, \boldsymbol{z}_{p_0})_{0,[0,T]}(\boldsymbol{x})$, where $\boldsymbol{z}_{p_0}$ is the p_0 component of the solution of the adjoint problem that is, the solution of

$$\boldsymbol{\mathfrak{L}}^\star \boldsymbol{z}(t) = (\Psi^* \boldsymbol{s})(t), \ [0,T], \ \boldsymbol{z}(T) = \boldsymbol{0}.$$

see Theorem 6 and Corollary 3. Therefore, we can write

$$\triangle\mu_{v_\mathrm{p}}(\boldsymbol{x}) \simeq (\partial_t \boldsymbol{u}_{p_0}, \boldsymbol{z}_{p_0})_{0,[0,T]}(\boldsymbol{x}).$$

Again we will make use of Green's function as we did in Section 3.4.1 and which are briefly described in Chapter A. A typical source in the forward problem is $f(t,\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0})\phi(t)$, with $\phi \in C_c^\infty(\mathbb{R})$, $\mathrm{supp}(\phi) \subset [0,T]$ and non-vanishing derivatives. Thus, we have an explicit representation of the pressure field

$$\boldsymbol{u}_{p_0}(\boldsymbol{x},t) = \big(G * (\delta_{\boldsymbol{x}_{\mathrm{src},0}} \partial_t \phi)\big)(\boldsymbol{x},t) = (G(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}, \cdot) *_t \partial_t \phi)(t),$$

where $*_t$ is the convolution only in time. The right-hand side of the adjoint problem is typically composed of the measured seismogram at the receiver locations. For a single receiver at $\boldsymbol{r}_0$ measuring at only one time γ_i we have $g(\boldsymbol{x},t) = \delta(\boldsymbol{x} - \boldsymbol{r}_0)\delta(t - \gamma_i)$ and can express the solution via

$$\boldsymbol{z}_{p_0}(\boldsymbol{x},t) = -(G * \partial_t g)(\boldsymbol{x}, T-t) = \partial_t (G * g)(\boldsymbol{x}, T-t) = \partial_t G(\boldsymbol{x} - \boldsymbol{r}_0, T - t - \gamma_i).$$

Plugging this in the equation for the update of v_p we get

$$\begin{aligned} \triangle\mu_{v_\mathrm{p}}(\boldsymbol{x}) &\simeq \int_0^T \partial_t \boldsymbol{u}_{p_0}(\boldsymbol{x},t) \cdot \boldsymbol{z}_{p_0}(\boldsymbol{x},t) \ \mathrm{d}t \\ &\simeq \int_0^T (G(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}, \cdot) *_t \partial_t^2 \phi)(t) \cdot \partial_t G(\boldsymbol{x} - \boldsymbol{r}_0, T - t - \gamma_i) \ \mathrm{d}t. \end{aligned} \tag{4.18}$$

We examine this in more detail for the 3D case (we do this because it is easier to do calculations compared to the 2D case) and plug in Green's function

$$G_3(\boldsymbol{x},t) = \frac{1}{4\pi v_\mathrm{p}^2 ||\boldsymbol{x}||_2} \delta\left(t - \frac{||\boldsymbol{x}||_2}{v_\mathrm{p}}\right)$$

into (4.18). This yields

$$\begin{aligned} \triangle\mu_{v_\mathrm{p},\gamma_i}(\boldsymbol{x}) &\simeq \int_0^T \frac{1}{\|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}\|_2} \partial_t^2 \phi\left(t - \frac{\|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}\|_2}{v_\mathrm{p}}\right) \cdot \frac{1}{\|\boldsymbol{x} - \boldsymbol{r}_0\|_2} \partial_t \delta\left(T - t - \gamma_i - \frac{\|\boldsymbol{x} - \boldsymbol{r}_0\|_2}{v_\mathrm{p}}\right) \ \mathrm{d}t \\ &\simeq \frac{1}{\|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}\|_2 \|\boldsymbol{x} - \boldsymbol{r}_0\|_2} \partial_t^3 \phi\left(T - \gamma_i - \frac{\|\boldsymbol{x} - \boldsymbol{r}_0\|_2 + \|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}\|_2}{v_\mathrm{p}}\right). \end{aligned}$$

The terms show singular behavior close to the receiver and source position. In a reconstruction we have a full seismogram where the s_i's are the values measured at the times γ_i, we can make use of the linearity of the adjoint equation, which results in

$$\triangle\mu_{v_\mathrm{p}}(\boldsymbol{x}) \simeq \sum_{i=0}^{N} s_i \int_0^T (G(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{src},0},\cdot) *_t \partial_t^2\phi)(t) \cdot \partial_t G(\boldsymbol{x}-\boldsymbol{r}_0, T-t-\gamma_i)\,\mathrm{d}t. \tag{4.19}$$

This shows that the terms from (4.18) form some kind of basis for the update. For a full seismogram this is a sum over all measurements and therefore the (near) singularities add up, and high values are expected in the vicinity of the sources and the receivers. A physics-based explanation of the artifacts near the source is that close to the sources, changes of the parameters have an effect on a large portion of the energy contained in the wave and hence it will result in a strong change in the resulting wave field.

Numerical examples

In this part present two kinds of numerical calculations. We visualize results of direct (i.e. evaluating the integral in (4.18) with the help of the Green's function) evaluations of the integral (which we will call kernel) from (4.19) for v_p in the case without damping (which corresponds to $L = 0$). We compare them with calculations from the M++ program where we evaluate the integrals from (4.4). Moreover, we also plot the results for ϱ and τ_p. Only in the τ_p case we use damping, the results of the damping and non-damping case are very similar for ϱ and v_p. For the experiment we place one source and one receiver with distance of 750 m and a homogeneous background speed of 3500 m/s and with $T = 0.4\,\mathrm{s}$ and $\triangle t = 0.0005\,\mathrm{s}$. As signal ϕ, we use $\tilde{r}$ from (4.14) with $f_c = 25\,\mathrm{Hz}$ and $t_\mathrm{shift} = 1.5/25.0\,\mathrm{s}$. Then, we evaluate the integral in (4.19) for artificial seismograms that are the canonical unit vectors in the discrete seismogram space, that is,

$$\boldsymbol{S} \ni \boldsymbol{s}_i(t) = \begin{cases} 1 & \text{for } t = \gamma_i, \\ 0 & \text{else.} \end{cases}$$

We do this for every γ_i in our time grid. By doing that we can see how sensitive the integral is to changes at the time γ_i in the seismogram. The results for points on a direct line connecting source and receiver in the v_p case are shown in Fig. 4.24. We can see that the values have the same shape, but are scaled versions of each other. Qualitatively, the results of the direct and the M++ calculations are similar. Further, they take the form of the second derivative of the source signal instead of the third as we would expect in the 3D case. In Fig. 4.25 in the top row we can see that the values for τ_p resemble the negative of the first derivative of the time source, and the values are also scaled versions of the same form along the connecting line. In the case of ϱ, we can see that the shape of the kernels change along the connecting line. At the source/receiver it is similar to the kernel in the v_p case, but it changes quickly to a curve that resembles the first derivative of the source signal. We can also see that the amplitude of the kernels drops quickly and to very low values, which could be part of the explanation why ϱ is hard to reconstruct in the transmission setup. In all calculations we see, that if we change one value of the seismogram in the support of these kernels, we always change all values along the connecting line of the source and receiver. In Fig. 4.26 the quantitative behavior of the amplitudes of these kernels relative to the distance of receiver and source (0 is receiver 1 is source) is depicted. The results are symmetric to the middle of the source and receiver for v_p and ϱ. For τ_p we see very large values around the source compared to the values at the receiver. These seem to be irregular and may be related to the large error of the source in the cell containing the source in our discretization. Once we move away from the cell containing the source, the amplitudes look symmetrical again, and the lowest amplitudes have a similar relative amplitude in the center of the line, compared to the receiver location. This symmetry can be seen on the right in Fig. 4.26. This result underlines the necessity of tapering around the source, which we do by ignoring the source cell in our inversions.

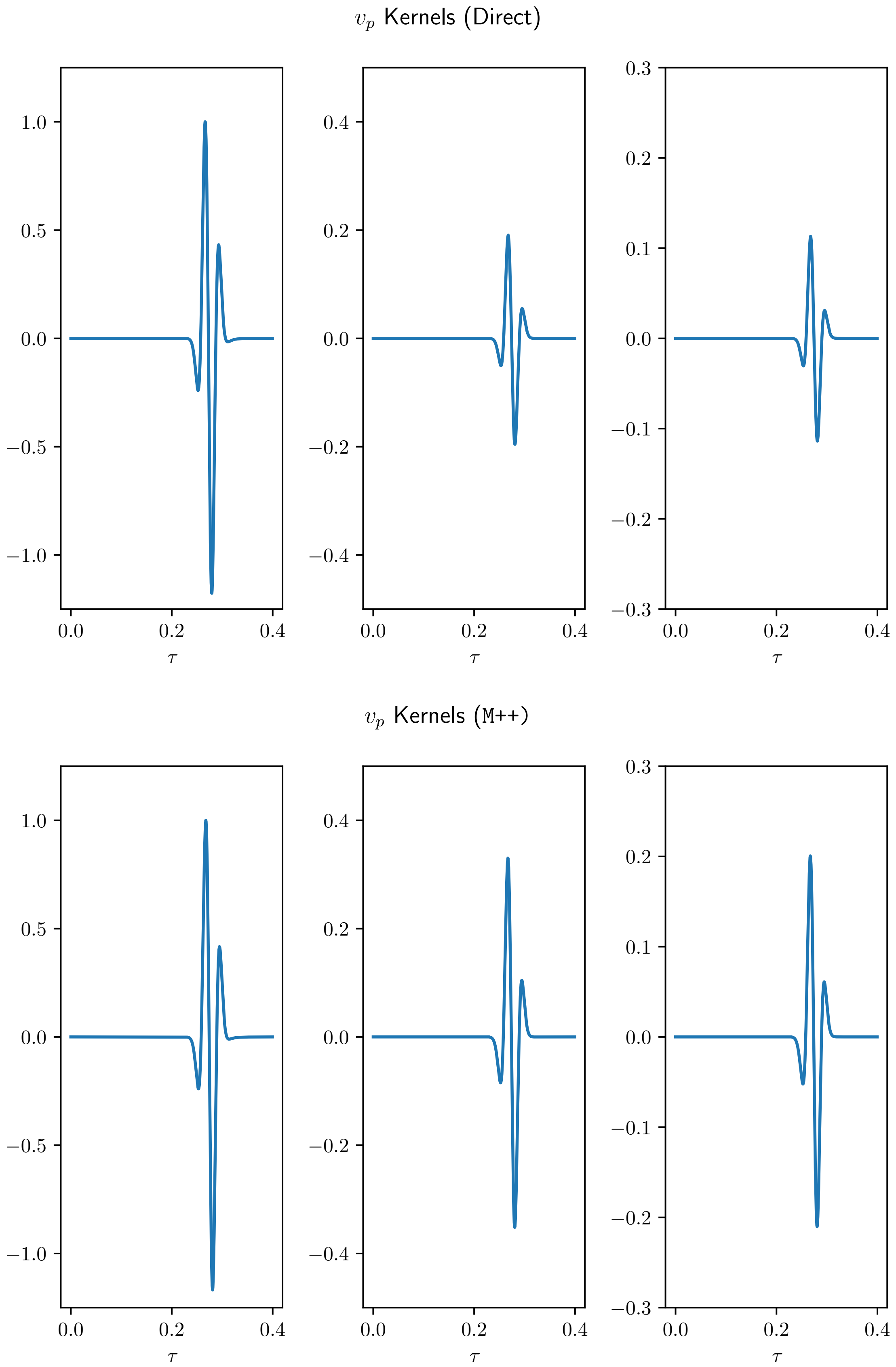
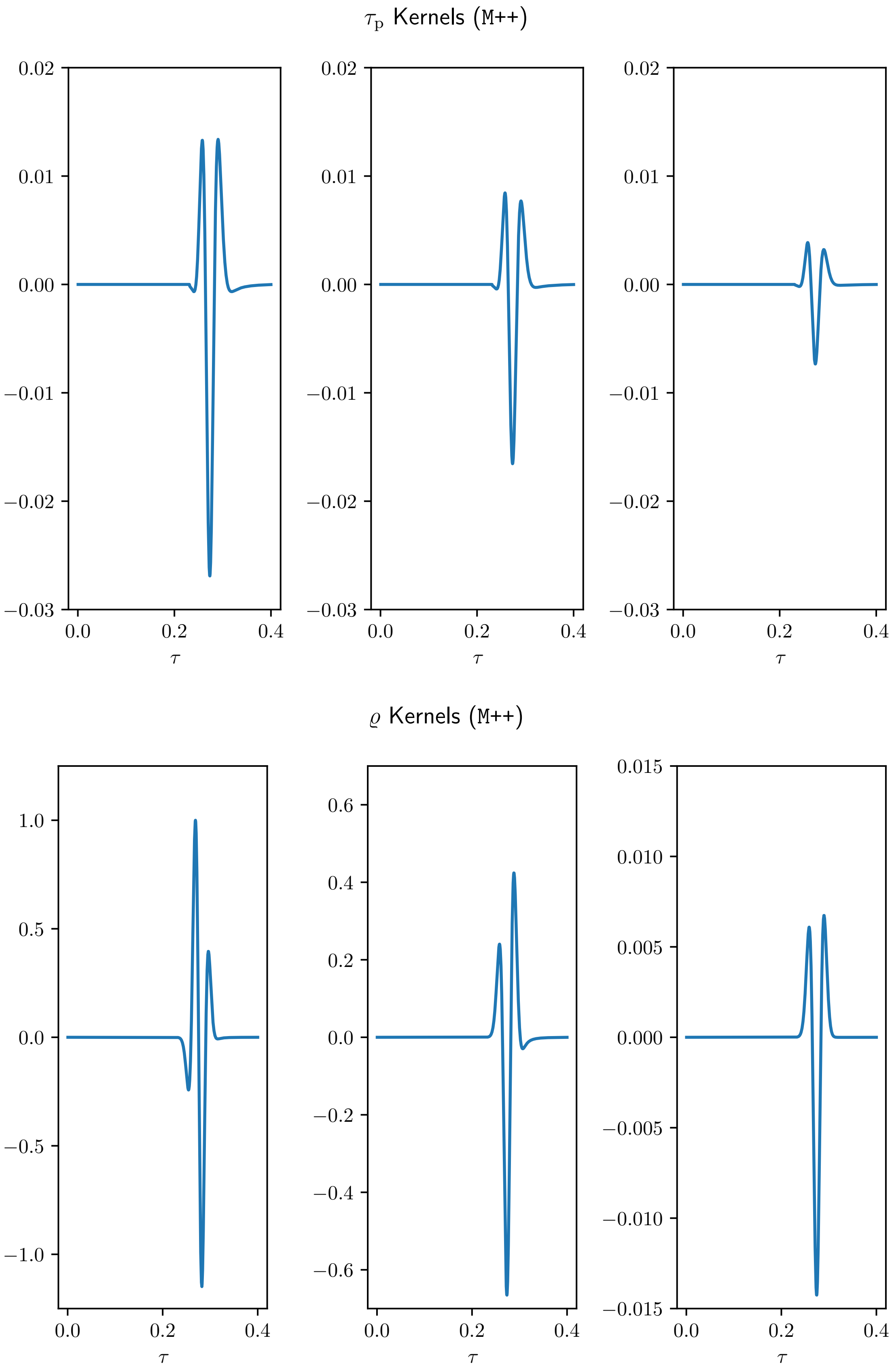


Figure 4.24: Normalized (i.e. to the maximal amplitude) v_p kernels (see (4.18)) at the source, at quarter distance and half distance to the source. Top: direct calculation of via Green's function, bottom: M++ calculation.


τ_p Kernels (`M++`)

ϱ Kernels (`M++`)

Figure 4.25: Normalized (i.e. to the maximal amplitude) of τ_p (top) and ϱ (bottom) kernels at the source, one percent distance and half distance to the source.

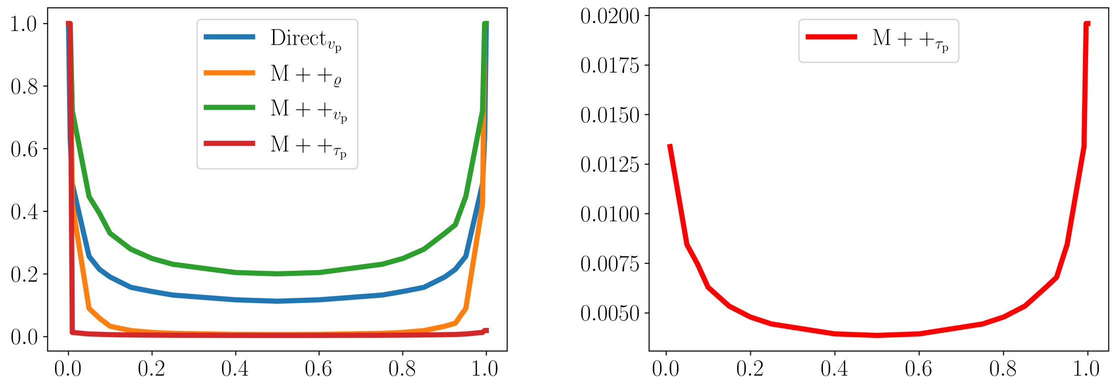


Figure 4.26: Left: normalized (i.e. to the maximal amplitude) amplitude of the kernels from Fig. 4.24 and Fig. 4.25. Right: for better visibility the τ_p part without the 'singular' values close to zero.

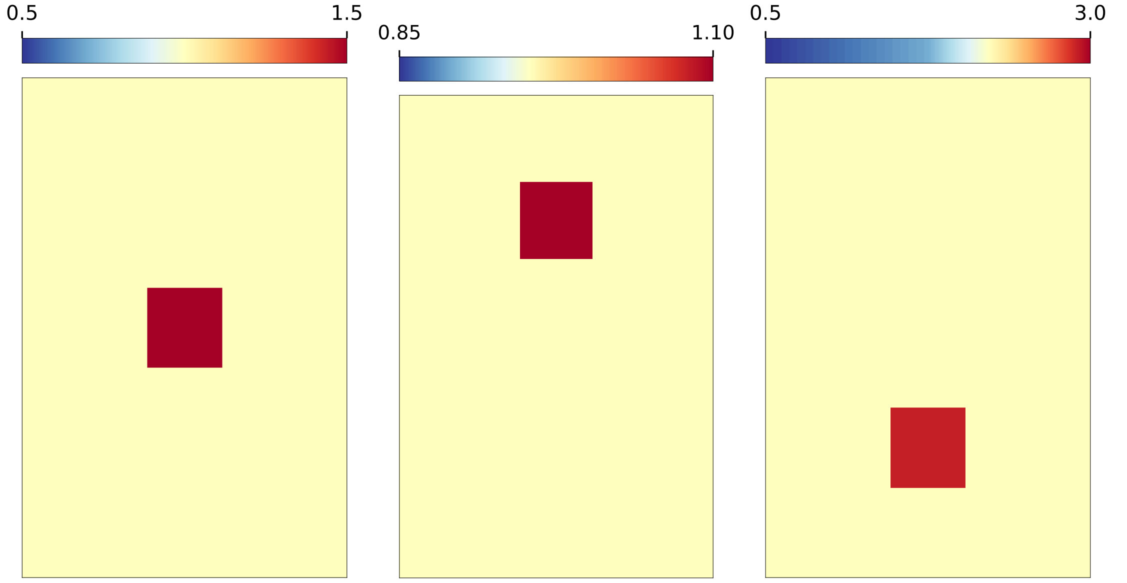
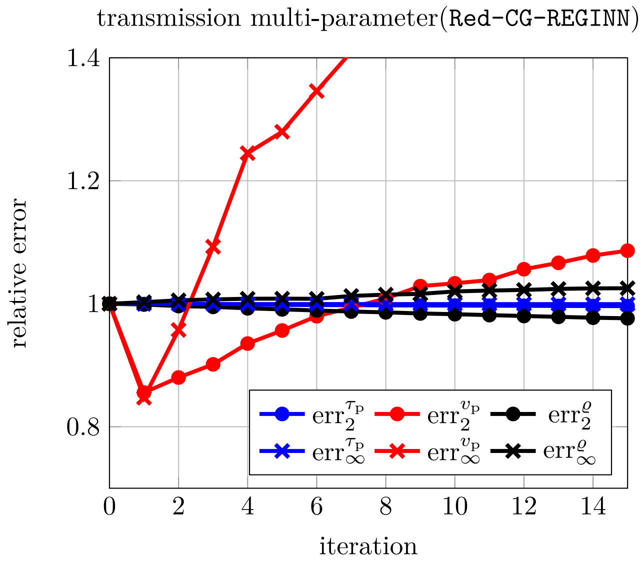


Figure 4.27: True values of the multi-parameter reconstruction. Left: $\mu^{\text{true}}_{v_{\text{p}}}$. Middle: $\mu^{\text{true}}_{v_{\text{p}}}$. Right: Dissipation $(Q^{-1}(\tau_{\text{p}}))$ $\mu^{\text{true}}_{1/\text{Q}}$ .To ensure good visibility, each image has its own color map where the maximal true values is set as the upper bound.

4.5 Multi-parameter reconstruction in FWI

Previously, in Section 4.4, we considered examples where only one of the parameters was sought for in each reconstruction. These were the mono-parameter reconstructions. The assumption that only one parameter (e.g. only v_{p}) is capable of explaining the measured data is a simplification. In reality, more than one material parameter is responsible for shape of the seismogram. Therefore, it is desirable to invert for more than one parameter at once. This is called multi-parameter inversion, and it is well-known to be a hard task. Typical difficulties are the coupling of the parameters in the updates (often called cross-talk) and that some parameters can be dominant in the inversion. This means in a joint reconstruction only one parameter gets updated significantly. These problems are described in many publications e.g. in [OGP+13].

Now, we apply `Red-CG-REGINN` to a multi-parameter problem, and apply the algorithm as before but calculate the updates for all parameters at the same time. Compared to the mono-parameter variant this adds almost no computational effort: exactly the same number of wave equations have to be solved numerically. Again, we consider the geometry from Fig. 4.1 and change one parameter in each Δ_i for $i = 1, 2, 3$ and leave the others constant. The true distributions of the parameters can be found in Fig. 4.27 and the numerical values of the parameters in Fig. 4.28. Results of this inversion are shown in Fig. 4.29. In the diagrams showing the errors in the reconstructions in Fig. 4.28, we can see that only the v_{p} component is updated significantly. From the results of the mono-parameter experiments we conducted before, we know that `Red-CG-REGINN` is capable of reconstructing parameter changes in τ_{p}. In the multi-parameter reconstruction, however, the changes of τ_{p} are very minor compared to the changes in the v_{p} component. This is due to the fact that the L^2-norm is more sensitive to phase difference of signals than to amplitude differences. Therefore, when the algorithm tries to minimize the residual, it



v_{bg}	3500 m/s	$\boldsymbol{\tau}$
v_{Δ_1}	3675 m/s	0.3207 s
ϱ_{bg}	2000 kg/m³	0.0748 s
ϱ_{Δ_2}	2500 kg/m³	0.0153 s
τ_{bg}	0.0767	0.0034 s
τ_{Δ_3}	0.3605	0.0013 s

Figure 4.28: Left: relative error as defined in (4.11) of the iterations for the reconstruction with Red-CG-REGINN in the multi-parameter case with inverse crime. Right: parameter values in the multi-parameter case.

is most 'greedy' in the v_p direction. This greediness makes the algorithm's updates scaled mainly to the v_p component and tends to suppress updates to parameters with lower sensitivity. Therefore, we aim to construct an algorithm that can scale the update differently for each component and increase the accuracy of the τ_p reconstruction in a simultaneous reconstruction with v_p. In the following, we explain our ansatz in a very general setting which is not exclusive to the problem (RED) and then apply it in our context of REGINN in FWI.

4.5.1 Multi Steepest Descent (mSD) - A subspace gradient descent ansatz

Let U, V be real Hilbert spaces with scalar products $(\cdot,\cdot)_U$ and $(\cdot,\cdot)_V$. For an inexact Newton method applied to a general nonlinear problem $\Gamma\colon U \longrightarrow V$, it is required to solve linear problems of the form $Ax = b,\ b \in V$, where $A \in \mathcal{L}(U,V)$ is the Fréchet derivative of Γ at the current iterate of the scheme. One typical way to approximate a solution is to minimize (in our case with regularization) a quadratic residual functional like

$$J\colon U \longrightarrow \mathbb{R},\ J(x) = \frac{1}{2}\|Ax - b\|_V^2\,.$$

If $Ax^* = b$, then x^* is also a critical point of J. The functional J is twice Fréchet differentiable for every $x \in U$ with

$$J'(x)[h] = (A^*(Ax-b), h)_U\,,\ \ J''(x)[h][\tilde{h}] = (A^*A\tilde{h}, h)_U\,,\ \text{ for } h, \tilde{h} \in U. \tag{4.20}$$

Given a descent direction $s \in U$ a common way to calculate the step size is

$$\varsigma = \underset{\tilde{\varsigma}\in\mathbb{R}}{\operatorname{argmin}}\, J\left(x + \tilde{\varsigma}s\right) = \frac{(-A^*(Ax-b), s)_U}{\|As\|_V^2}\,. \tag{4.21}$$

A well-known characterization of a descent direction for differentiable functions is that $J'(x)[s] < 0$ and therefore we know $-J'(x)[s] = (-A^*(Ax-b), s)_U > 0$ and ς is assured to be positive. Choosing the descent

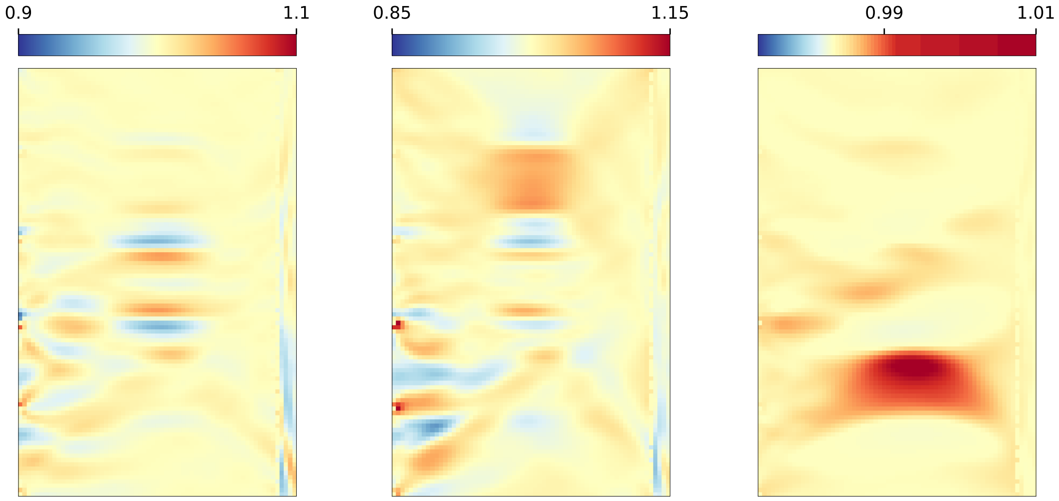


Figure 4.29: Results of the multi-parameter inversions with `Red-CG-REGINN`. Left: μ_{ϱ}^{15}. Middle: $\mu_{v_{\mathrm{p}}}^{15}$. Right: $\mu_{1/Q}^{15}$. The values of the iterates are divided by $\varrho_{\mathrm{bg}} = 2000\,{}^{\mathrm{kg}}/{}_{\mathrm{m}^3}$, $v_{\mathrm{p,bg}} = 3500\,{}^{\mathrm{m}}/{}_{\mathrm{s}}$, $1/Q_{\mathrm{bg}} = 1/15$. Note that only $\mu_{v_{\mathrm{p}}}^{15}$ is considerably different to the initial value, which is 1 for all pictures.

direction $s = -A^*(Ax - b)$, results in the standard steepest descent step size cf. [NW06, Chapter 3.3]. We mentioned at the beginning of the section that we want to find individual factors for each parameter component. Therefore, we choose some finite orthogonal decomposition $U = U_0 \oplus \cdots \oplus U_{M-1}$, $M \in \mathbb{N}$, and denote the corresponding orthogonal projections onto U_m by $\Pi_m \colon U \longrightarrow U_m \subset U$. Let $d \in U$ be an arbitrary vector and denote the component of d in U_m by $d_m := \Pi_m d \in U_m$. Now, we can write $d = \sum_{m=0}^{M-1} d_m$. Transferring the ansatz from (4.21) to our idea, we look for the M-tuple that solves the minimization problem

$$\alpha := (\alpha_0, \ldots, \alpha_{M-1}) = \operatorname*{argmin}_{(\tilde{\alpha}_0,\ldots,\tilde{\alpha}_{M-1})\in\mathbb{R}^M} J\left(x + \sum_{m=0}^{M-1} \tilde{\alpha}_m d_m\right) \in \mathbb{R}^M .$$

To find the minimizing $(\alpha_0, \ldots, \alpha_{M-1})$, we set the partial derivatives to zero:

$$\partial_{\tilde{\alpha}_n} J\left(x + \sum_{m=0}^{M-1} \tilde{\alpha}_m d_m\right) \overset{!}{=} 0 . \tag{4.22}$$

If we plug the derivative from (4.20) into (4.22) for all $n = 0, \ldots, M-1$ we get

$$J'\left(x + \sum_{m=0}^{M-1} \tilde{\alpha}_m d_m\right)[d_n] = \underbrace{(A^*(Ax-b), d_n)_U}_{=J'(x)[d_m]} + \sum_{m=0}^{M-1} \tilde{\alpha}_m \underbrace{(A^*Ad_m, d_i)_U}_{=J''(x)[d_m][d_n]} \overset{!}{=} 0 .$$

This results in M linear equations for α_m, $m = 0, \ldots, M-1$. Now, define the matrix

$$B := \begin{pmatrix} (Ad_0, Ad_0)_V & (Ad_1, Ad_0)_V & \cdots & (Ad_{M-1}, Ad_0)_V \\ \vdots & \vdots & \vdots & \vdots \\ (Ad_0, Ad_{M-1})_V & (Ad_1, Ad_{M-1})_V & \cdots & (Ad_{M-1}, Ad_{M-1})_V \end{pmatrix} \in \mathbb{R}^{M\times M} , \tag{4.23}$$

and the vectors

$$c := -\begin{pmatrix} (A^*(Ax-b), d_0)_U \\ \vdots \\ (A^*(Ax-b), d_{M-1})_U \end{pmatrix} = -\begin{pmatrix} J'(x)[d_0] \\ \vdots \\ J'(x)[d_{M-1}] \end{pmatrix} \in \mathbb{R}^M .$$

With these definitions we can express (4.22) as the linear problem

$$B\alpha = c\,. \tag{4.24}$$

The matrix B is a Gramian matrix and hence positive semi-definite. For any $\beta \in \mathbb{R}^M \setminus \{0\}$ it holds

$$\beta^\top B\beta = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1}(Ad_m, Ad_n)_V\beta_m\beta_n = \left(\sum_{m=0}^{M-1}\beta_m Ad_m, \sum_{n=0}^{M-1}\beta_n Ad_n\right)_V = \left\|A\sum_{m=0}^{M-1}\beta_m d_m\right\|_V^2 \geq 0\,. \tag{4.25}$$

Because of orthogonality, the d_m's are pairwise linearly independent, we can assure positive definiteness if A is an injective operator. Under this assumption existence and uniqueness of such a minimizing M-tuple α is ensured. Using this ansatz we get a descent direction of J which the following short calculation shows.

Lemma 3. *Let $d_m \in U_m$ for $m = 0,\dots M-1$ and $U = U_0 \oplus \cdots \oplus U_{M-1}$. Further, assume that $A \in \mathcal{L}(U,V)$ is injective. Let $\alpha = \left(\alpha_0,\dots,\alpha_{M-1}\right)^\top$ solve the linear system (4.24) with the d_m for $m = 0,\dots M-1$ plugged into B and c. Then, the vector $\sum_{m=0}^{M-1}\alpha_m d_m \in U$ is a descent direction of J.*

Proof. We show this with the characterization of descent directions for differentiable functions.

$$J'(x)\left[\sum_{m=0}^{M-1}\alpha_m d_m\right] = \sum_{m=0}^{M-1}\alpha_m J'(x)[d_m] = -(\alpha,c)_{\mathbb{R}^M} \overset{(4.24)}{=} -(\alpha,B\alpha)_{\mathbb{R}^M} \overset{(4.25)}{<} 0\,.$$

□

Note that, unlike in (4.21), is not ensured that $\alpha_m > 0$ for $m = 0,\dots,M-1$.

Remark 7. *The descent direction $s = \sum_{m=0}^{M-1}\alpha_m d_m$ is also optimal in the sense of the one-dimensional optimization in (4.21):*

$$\varsigma = \frac{-(A^*(Ax-b), \sum_{m=0}^{M-1}\alpha_m d_m)_U}{\|A\sum_{m=0}^{M-1}\alpha_m d_m\|_V^2} = \frac{(\alpha,c)_{\mathbb{R}^M}}{\|\sum_{m=0}^{M-1}\alpha_m Ad_m\|_V^2} \overset{(4.25)}{=} \frac{(\alpha,c)_{\mathbb{R}^M}}{\alpha^\top B\alpha} \overset{(4.24)}{=} 1\,.$$

So far, we have not described an algorithm to solve the linear problem $Ax = b$, but rather a way to calculate a descent direction on $J(x)$. The algorithm results from applying this procedure iteratively. We start with an initial guess x_0 and calculate the descent direction in $J(x_0)$, then we update $x^1 = x_0 + d$ and calculate the descent direction in $J(x_1)$ and so on until a stopping criterion is met. In our case, we use the same stopping criterion (4.6) as we used for the `CG` method in Section 4.3. We add the possibility to ensure that a minimal number of inner iterations $l_{\min}$ are done and add the possibility of adding a preconditioner. The whole algorithm is shown in Algorithm 4.4. The combination with `REGINN` is called `mSD-REGINN`. Subspace ansatzes like the one we presented were proposed in the seismic inversion literature before. For some general nonlinear methods in geophysics a similar ansatz was proposed in [KSW88]. In [STK91] the authors propose an algorithm which applies to a multi-parameter FWI without regularization. In [OME93] by Oldenburg, McGillivray and Ellis a subspace algorithm in the context of geophysics is discussed. This publication has influenced publications in FWI like [LAH12] where the authors use a sparse model representation in conjunction with Gauss-Newton method for inversion, whereas the authors in [MHMG10] use a sparse sampling of the model. In [MVdKG⁺10, XM14] the authors compute step sizes for every parameter separately without using additional information how they influence each

Algorithm 4.4 Preconditioned Multi Steepest Descent (PmSD) algorithm (inner loop)

```
Require: $A \in \mathcal{L}(U,V)$, Precon: $V \longrightarrow V$, $b \in V$, $\vartheta \in (0,1)$, $l_{\min}, l_{\max} \in \mathbb{N}$, $x^0 \in U$
 1: $l \leftarrow 0$,
 2: $r^0 \leftarrow b - Ax^0$
 3: repeat
 4:     $d^l \leftarrow A^* r^l$
 5:     $d_{\mathrm{PC}} \leftarrow \mathrm{Precon}(d)$
 6:     $B^l, c^l \leftarrow \mathrm{AssembleLinearSystem}(A, d_{\mathrm{PC}})$ % see (4.24).
 7:     $\alpha \leftarrow (B^l)^{-1} c^l$ % $\alpha \in \mathbb{R}^M$
 8:     $x^{l+1} \leftarrow x^l + \sum_{m=0}^{M-1} \alpha_m d^l_{\mathrm{PC},m}$
 9:     $r^{l+1} \leftarrow r^l - \sum_{m=0}^{M-1} \alpha_m A d^l_{\mathrm{PC},j}$ % $Ad^l_{\mathrm{PC},j}$ can be reused here if it is saved in line 6.
10:     $l \leftarrow l + 1$
11: until $(\|r^l\|_V \leq \vartheta \|b\|_V$ or $l \geq l_{\max})$ and $l \geq l_{\min}$
12: return $(x^l, l-1)$
```

other. Both approaches suffer from the need to manually adjust parameters, which can be very problem dependent. Additionally, in [XM14] the authors carry out an expensive line search. In the explanation of the algorithm we presented one step of the algorithm. In [GIP20] they use techniques similar to our ansatz but they build their coupling matrices (in our case called B) with Full-Hessian evaluations and use a more elaborate choice of descent directions. What makes mSD-REGINN unique among the algorithms above, is using it as a regularized iterative method for our inexact Newton scheme. The algorithm to solve the inner loop can also be interpreted in the light of projection optimization methods [Saa03]. With our choice of d as the gradient, we do not incorporate information about prior updates which could speed up convergence. Therefore, a possible and natural extension of mSD-REGINN could be a combination with the *Sequential Subspace Optimization Method* (SESOP) [NZ05] (or the corresponding regularized version thereof *Regularized SESOP* [WS17]). The idea of SESOP can be described with our algorithm. While we take different parts of our gradient into account in the optimization, they take multiple descent directions and optimize the same problem as we did. Therefore, if we had L descent directions and split them into M parts, the matrix B from (4.23) is in $\mathbb{R}^{LM \times LM}$.

4.5.2 Application of mSD-REGINN in the reduced case

Now, we want to apply mSD-REGINN to the reduced problem (RED). The functional arising from the Newton ansatz in the k-th iteration for shot ξ is

$$J^k_\xi \colon \boldsymbol{P} \longrightarrow \mathbb{R},\ J_{k,\xi}(\triangle\mu^k_\xi) = \frac{1}{2}\|\Phi'_{\mu^k,\xi}[\triangle\mu^k_\xi] - \boldsymbol{r}^k_\xi\|^2_{\boldsymbol{S}},$$

with $\boldsymbol{r}^k_\xi = \boldsymbol{s}_{\mathrm{obs},\xi} - \Phi_\xi(\mu^k_\xi)$. Further, we use the abbreviations $\Phi'_{\mu^k,\xi} = \Phi'_\xi(\mu^k)$ and $\Phi'^*_{\mu^k,\xi} := \Phi'^*_\xi(\mu^k)$. In section 4.5.1 the choice of $d \in U$ was left completely free. We use the standard gradient (see line 4 of Algorithm 4.4), but other choices would be possible, too.

Algorithm 4.5 `Red-mSD-REGINN` algorithm

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess; $\boldsymbol{s}_{\mathrm{obs}} \in \boldsymbol{S}^{\Xi}$ % seismograms
Ensure: $\mu^k \in \boldsymbol{P}_h$ % approximate solution of (RED)
$k \leftarrow 0$
while not termination do
    $\mu_0^k \leftarrow \mu^k$
    $\xi \leftarrow 0$
    repeat
        $\boldsymbol{r}_\xi^k \leftarrow \boldsymbol{s}_{\mathrm{obs},\xi} - \Phi_\xi(\mu_\xi^k)$
        determine $l_{\max,\xi}^k$ and $\vartheta_\xi^k$ % according to (4.7) and (4.6)
        $(\triangle\mu_\xi^k, l_\xi^k) \leftarrow$ PmSD($\Phi'_{\mu^k,\xi}, I, \boldsymbol{r}_\xi^k, \vartheta_\xi^k, l_{\max,\xi}^k$, 1,$\mathbf{0}$) % call of Algorithm 4.4
        $\mu_{\xi+1}^k \leftarrow \mu_\xi^k + \triangle\mu_\xi^k$
        $\xi \leftarrow \xi + 1$
    until $\xi > \Xi - 1$
    $\mu^{k+1} \leftarrow \mu_{\Xi-1}^k$
    $k \leftarrow k + 1$
end while
return $\mu^k$
```

Transmission Problem

We choose $M = 3$ and use the decomposition $\boldsymbol{P} = \boldsymbol{P}_\varrho \oplus \boldsymbol{P}_{v_{\mathrm{p}}} \oplus \boldsymbol{P}_{\tau_{\mathrm{p}}}$ with the projections $\Pi_\varrho, \Pi_{v_{\mathrm{p}}}, \Pi_{\tau_{\mathrm{p}}}$, which are the identity on the parameter in the subscript and set the other parameters to zero. Note that it is additionally possible to divide the space geometrically, which we do not employ in this experiment. As mentioned before we expect ϱ to be hard/almost impossible to reconstruct for this experimental setup, we include it in the method for now to put as little a priori information in the algorithm as possible. The matrix from (4.23) in this case then has the form

$$B^l = \begin{pmatrix} \|\Phi'_{\mu^k,\xi}[d_1^l]\|_{\boldsymbol{S}}^2 & \left(\Phi'_{\mu^k,\xi}[d_1^l], \Phi'_{\mu^k,\xi}[d_2^l]\right)_{\boldsymbol{S}} & \left(\Phi'_{\mu^k,\xi}[d_1^l], \Phi'_{\mu^k,\xi}[d_3^l]\right)_{\boldsymbol{S}} \\ \left(\Phi'_{\mu^k,\xi}[d_2^l], \Phi'_{\mu^k,\xi}[d_1^l]\right)_{\boldsymbol{S}} & \|\Phi'_{\mu^k,\xi}[d_2^l]\|_{\boldsymbol{S}}^2 & \left(\Phi'_{\mu^k,\xi}[d_2^l], \Phi'_{\mu^k,\xi}[d_3^l]\right)_{\boldsymbol{S}} \\ \left(\Phi'_{\mu^k,\xi}[d_3^l], \Phi'_{\mu^k,\xi}[d_1^l]\right)_{\boldsymbol{S}} & \left(\Phi'_{\mu^k,\xi}[d_3^l], \Phi'_{\mu^k,\xi}[d_2^l]\right)_{\boldsymbol{S}} & \|\Phi'_{\mu^k,\xi}[d_3^l]\|_{\boldsymbol{S}}^2 \end{pmatrix},$$

with d^l being the descent direction and $d_1^l = \Pi_\varrho d^l, d_2^l = \Pi_{v_{\mathrm{p}}} d^l, d_3^l = \Pi_{\tau_{\mathrm{p}}} d^l$. The right-hand side is

$$c^l = \begin{pmatrix} \left(\Phi'_{\mu^k,\xi}[\triangle\mu_\xi^k] - \boldsymbol{r}_\xi^l, \Phi'_{\mu^k,\xi}[d_1^l]\right)_{\boldsymbol{S}} \\ \left(\Phi'_{\mu^k,\xi}[\triangle\mu_\xi^k] - \boldsymbol{r}_\xi^l, \Phi'_{\mu^k,\xi}[d_2^l]\right)_{\boldsymbol{S}} \\ \left(\Phi'_{\mu^k,\xi}[\triangle\mu_\xi^k] - \boldsymbol{r}_\xi^l, \Phi'_{\mu^k,\xi}[d_3^l]\right)_{\boldsymbol{S}} \end{pmatrix}.$$

For each step in the inner loop it is necessary to solve $M + 1 = 4$ wave equations. One for evaluation of the adjoint $\Phi'^{*}_{\mu^k,\xi}$ and $M = 3$ for the linearized wave equations $\Phi'_{\mu^k,\xi}[d_m^l], m = 1, 2, 3$.

Multi-parameter inversion for ϱ, v_{p} and τ_{p}

The reconstruction result with `Red-mSD-REGINN` is shown in Fig. 4.30 and the convergence history of the reconstructions can be seen in Fig. 4.31. We can see that – in contrast to the inversion with

Red-CG-REGINN shown in Fig. 4.28 – the errors in v_p and the dissipation $Q^{-1}(\tau_\mathrm{p})$ both decrease in a monotone way. The error for the ϱ component is the same as for the mono-parameter inversion. This experiment shows that if we apply Red-mSD-REGINN in the way we presented it above, it is a clear improvement to the inversion with Red-CG-REGINN. Application of Red-mSD-REGINN can help for simultaneous inversion of v_p and τ_p. In the reconstructions in Fig. 4.30 we can see that especially the v_p part has fewer artifacts compared to the Red-CG-REGINN inversion in the region where τ_p is changed. The algorithm Red-CG-REGINN compensates changes in τ_p with changes in v_p since the τ_p updates are suppressed. In this experiment we commit inverse crime.

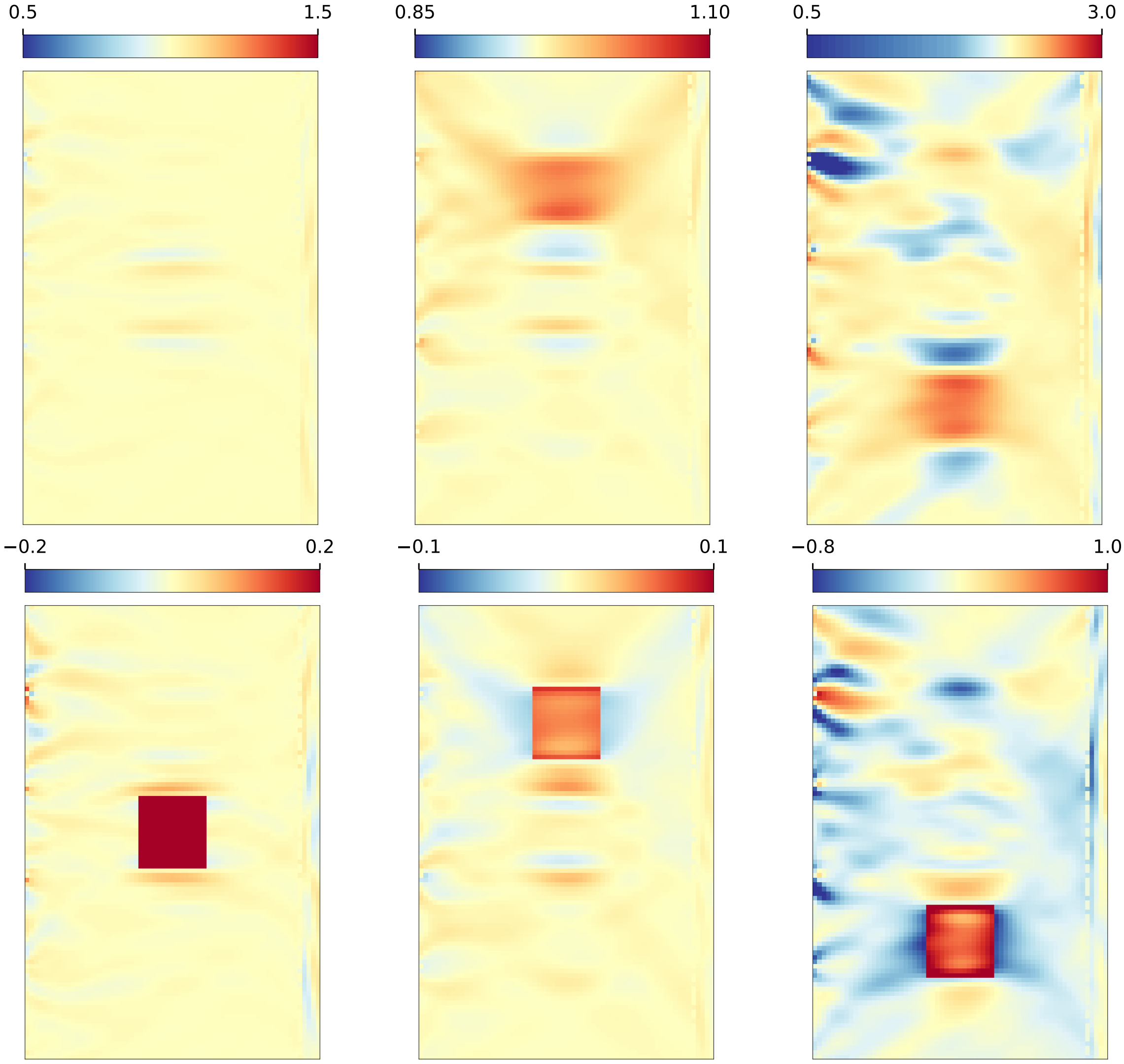


Figure 4.30: Results of the multi-parameter inversions with Red-mSD-REGINN. Top: final reconstructions μ_ϱ^{10}, $\mu_{v_\mathrm{p}}^{10}$, and $\mu_{\tau_\mathrm{p}}^{10}$, bottom: difference of the true value with the final reconstruction $\delta\mu_\varrho^{10}$, $\delta\mu_{v_\mathrm{p}}^{10}$, and $\delta\mu_{1/Q}^{10}$.The values of the iterates are divided by $\varrho_\mathrm{bg} = 2000\,\mathrm{kg/m^3}$, $v_\mathrm{p,bg} = 3500\,\mathrm{m/s}$, $1/Q_\mathrm{bg} = 1/15$.

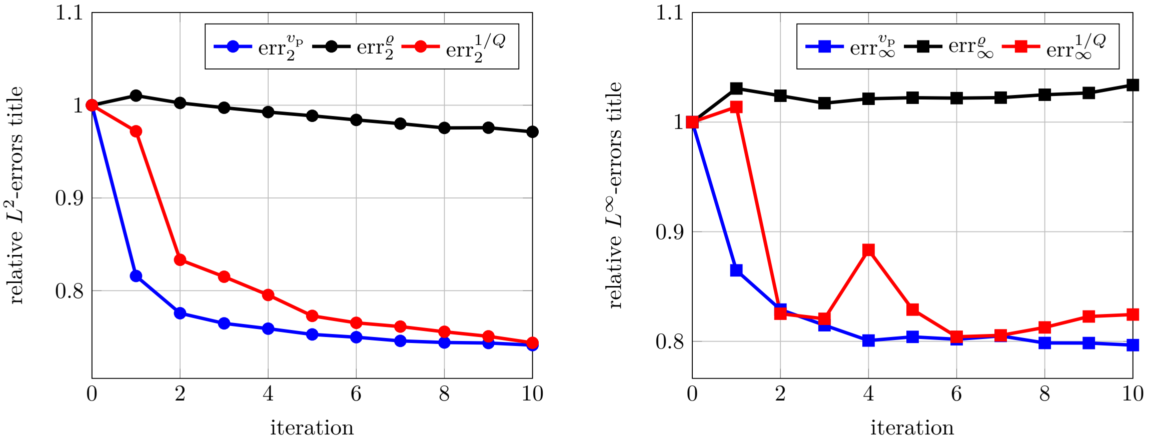


Figure 4.31: Relative errors, as defined in (4.11), of the multi-parameter inversions with `mSD-REGINN` from Section 4.5.2 Left: L^2-error. Right: L^∞-error.

4.5.3 Subspace Preconditioned `Red-CG-REGINN` (`Red-SPCG-REGINN`)

In this section we present a preconditioner, that is motivated by the results in the previous chapter. We want to take the information we gather in the matrix B^l and use it as a preconditioner to improve the CG algorithm. The first version we tested is the block-diagonal preconditioner

$$C_1^l = \operatorname{diag}(B^l, \dots, B^l) \in \mathbb{R}^{N_{\text{cells}} N_{\text{inv}} \times N_{\text{cells}} N_{\text{inv}}} .$$

where N_{cells} is the number of cells in the mesh and N_{inv} the number of parameters we want to invert at once. For this preconditioner, the matrix B^l has to be calculated in every inner loop, which makes this algorithm have the same cost as the Red-mSD-REGINN. We call the algorithm resulting from this choice SKPCG$_1$-REGINN. If we freeze the preconditioner we save the costly way of calculating B^l. This is the second preconditioner we want to investigate. This algorithm has an overhead of 3 wave equations to calculate B^0, but then it has the same cost as a normal CG algorithm. In this case we define $C_2^l = C_0^l$ and call it SKPCG$_2$-REGINN. Another preconditioner we tested was multiplying the B^l's with the background values of the parameters $D_{\text{bg}} \coloneqq \operatorname{diag}(\rho_{\text{bg}}^{-1}, v_{\text{bg}}^{-1}, \tau_{\text{bg}}^{-1})$ to additionally take the different scaling of the parameters into account. This variant is called SKPCG$_3$-REGINN. Here, we set

$$C_3^l \coloneqq C_1^l \cdot \operatorname{diag}(D_{\text{bg}}) \in \mathbb{R}^{N_{\text{cells}} N_{\text{inv}} \times N_{\text{cells}} N_{\text{inv}}} .$$

All these preconditioners can be seen as a crude approximation to the inverse $\Phi'_\mu \Phi'^*_\mu$ after discretization which of course is too expensive to compute in its entirety.

Results

We do not present reconstructions or error graphs here since this ansatz produced reconstructions that were significantly worse than the ones with Red-mSD-REGINN. The algorithm $\texttt{Red} - \texttt{SKPCG}_1$-REGINN produced more or less the same results as $\texttt{Red} - \texttt{SKPCG}_2$-REGINN, showing that the expensive recalculation each step does not bring any advantages. The rescaling in $\texttt{Red} - \texttt{SKPCG}_3$-REGINN also not did not improve the results. All-in-all the tests indicated that this is not a viable way to implement a preconditioner for the multi-parameter CG. For this reason, we did not pursue this ansatz further.

CHAPTER

FIVE

INVERSION WITH ALL-AT-ONCE FORMULATION

5.1 All-at-once formulation

In this chapter we present an approach to solve the inverse problem of FWI which differs from the reduced approach in Section 4.1.1. In Definition 3 of the parameter-to-solution map it is ensured that the seismogram is the point-wise evaluation of a wave field that solves the wave equation (2.8). However, we can relax this assumption and modify the problem (RED) to solve for $\boldsymbol{u}$ in the inversion process instead of only for μ. Before we give the formulation of the problem, we have to introduce some notation.

5.1.1 All-at-once operator

In this section we follow [Rie21] closely and apply the theorems to the visco-acoustic instead of the visco-elastic case, which are presented therein. We recall some definitions from Chapter 2. Two spaces we frequently use are

$$\boldsymbol{X} = L^2(\Omega, \mathbb{R}^{2+(1+L)}) \qquad \text{and} \qquad \boldsymbol{Y} = L^2([0,T], \boldsymbol{X})\,.$$

For $\mu = (\varrho, v_\mathrm{p}, \tau_\mathrm{p}) \in \boldsymbol{P}_\mathrm{adm} \subset \boldsymbol{P}$ and $\boldsymbol{u} = (\boldsymbol{v}, p_0, \dots, p_L) \in \boldsymbol{X}$ we defined

$$\begin{aligned} \boldsymbol{M}(\mu)\boldsymbol{u} &= (\varrho\boldsymbol{v}, \nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})p_0, \tau_\mathrm{p}^{-1}\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})p_1, \dots, \tau_\mathrm{p}^{-1}\nu(\varrho, v_\mathrm{p}, \tau_\mathrm{p})p_L)\,, \\ \boldsymbol{D}\boldsymbol{u} &= (\boldsymbol{0}, 0, \tau_1^{-1}p_1, \dots, \tau_L^{-1}p_L)\,, \\ \boldsymbol{A}\boldsymbol{u} &= -(\nabla(p_0 + \dots + p_L), \operatorname{div}\boldsymbol{v}, \dots, \operatorname{div}\boldsymbol{v})\,, \end{aligned}$$

with $\nu\colon \mathbb{R}^3 \longrightarrow \mathbb{R},\ (r, v, \tau) \longmapsto \frac{1+\alpha\tau}{v^2 r}$, where $\alpha = \alpha(\omega_0) = \sum_{l=1}^{L} \frac{\omega_0^2\tau_l^2}{1+\omega_0^2\tau_l^2}$. The constant is set to $\omega_0 = 2\pi f_c$, with f_c being the central frequency of the source signal. With this choice it is assured that ω_0 lies within the frequency band we want to model. In the proof of Theorem 2 we showed that $\boldsymbol{A}$ is the generator of a strongly continuous operator semigroup and therefore with Theorem 1 we get that for $\boldsymbol{f} \in L^1([0,T], \boldsymbol{X})$ and $\boldsymbol{u}_0 \in \mathcal{D}(\boldsymbol{A})$ there exists a unique mild solution $\boldsymbol{u} \in C([0,T], \boldsymbol{X})$ for

$$\boldsymbol{M}(\mu)\partial_t\boldsymbol{u}(t) + (\boldsymbol{A} + \boldsymbol{M}(\mu)\boldsymbol{D})\boldsymbol{u}(t) = \boldsymbol{f}(t)\,, \quad t \in [0,T]\,, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0\,.$$

If we turn to the integrated version of the equation

$$\boldsymbol{M}(\mu)\boldsymbol{u}(t) + (\boldsymbol{A} + \boldsymbol{M}(\mu)\boldsymbol{D})\int_0^t \boldsymbol{u}(s)\ \mathrm{d}s = \boldsymbol{M}(\mu)\boldsymbol{u}_0 + \int_0^t \boldsymbol{f}(s)\ \mathrm{d}s, \quad t \in [0,T]\,,$$

we get the additional notion of the *integrated solution* of the evolution equation. In our case the mild and the integrated solution are the same since $\boldsymbol{A}$ is the generator of a strongly continuous operator semigroup, cf. [Sch19, Proposition 2.15]. Due to its rather abstract representation via semigroups working with mild solutions explicitly is hard. On the other hand, the integrated solution has an explicit representation that simplifies its use and still enables us to require less regularity in $\boldsymbol{f}$ when formulating the inverse problem.

Remark 8. *In the next definition we apply a trick to achieve boundedness of an extension of the operator* $\boldsymbol{A}$*. For this, we introduce the norm* $\|\cdot\|_{-1} = \|(\boldsymbol{I}+\boldsymbol{A})^{-1}\cdot\|_{\boldsymbol{X}}$ *and consider* $\boldsymbol{X}_{-1}$ *as the completion of* $\boldsymbol{X}$ *with respect to this weaker norm. The norm is well-defined since* -1 *is in the resolvent set of* $\boldsymbol{A}$ *[Sch19]. We can extend* $\boldsymbol{A}$ *by* $\boldsymbol{A}_{-1}$ *which is a linear operator from* $\boldsymbol{X}_{-1}$ *to* $\boldsymbol{X}_{-1}$*. In [Sch19] it is shown that the graph norm of* $\boldsymbol{A}_{-1}$ *is equivalent to* $\|\cdot\|_X$*. Therefore we get that* $D(\boldsymbol{A}_{-1}) = \boldsymbol{X}_{-1}$ *and* $\boldsymbol{A}_{-1} \in \mathcal{L}(\boldsymbol{X}, \boldsymbol{X}_{-1})$*. With* $\boldsymbol{Y}_{-1} := L^2([0,T], \boldsymbol{X}_{-1})$ *we get* $\boldsymbol{A}_{-1} \in \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}_{-1})$*. We omit further details here and cite Schnaubelt:* "[...] one can quite often use $\boldsymbol{Y}_{-1}$[that is, spaces such as $\boldsymbol{Y}_{-1}$] to 'legalize illegal computations' without knowing a precise description of it" *[Sch19, p. 52]. To ensure mathematical correctness of the theorems we will use the notation above. For more detailed explanations we refer to [Rie21] or [Sch19].*

Now, we define the operators we need for the formulation of the all-at-once setting.

Definition 6. *Let* Ψ *be the measurement operator as defined in Section 2.4, and* $\boldsymbol{P}_{\text{adm}}$ *the set of admissible parameters introduced in Section 2.2 and* $\lambda \in \mathbb{R}_+$*. The* all-at-once operator *is defined by*

$$\tilde{\Phi}_\lambda \colon \boldsymbol{Y} \times \boldsymbol{P}_{\text{adm}} \longrightarrow \boldsymbol{Y}_{-1} \times \boldsymbol{S},\ (\boldsymbol{u}, \mu) \longmapsto \begin{pmatrix} \lambda\left(\tilde{\mathfrak{L}}(\mu)\boldsymbol{u} - \boldsymbol{M}(\mu)\boldsymbol{u}_0\right) \\ \Psi\boldsymbol{u} \end{pmatrix}, \tag{5.2}$$

with

$$\tilde{\mathfrak{L}} \colon \boldsymbol{P}_{\text{adm}} \longrightarrow \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}_{-1}),\ \mu \longmapsto \tilde{\mathfrak{L}}(\mu)\boldsymbol{u} := \boldsymbol{M}(\mu)\boldsymbol{u} + (\boldsymbol{M}(\mu)\boldsymbol{D} + \boldsymbol{A}_{-1})\boldsymbol{J}\boldsymbol{u}\,, \tag{5.3}$$

where

$$\boldsymbol{J} \colon \boldsymbol{Y} \longrightarrow \boldsymbol{Y},\ \boldsymbol{u} \longmapsto (\boldsymbol{J}\boldsymbol{u})(t) = \int_0^t \boldsymbol{u}(s)\,\mathrm{d}s\,.$$

Hence, for a fixed λ the nonlinear problem we aim to solve is:

$$\text{Given } (\boldsymbol{s}_{\text{obs}}, \boldsymbol{f})^\top \text{ find } (\boldsymbol{u}, \mu)^\top \in \boldsymbol{Y} \times \boldsymbol{P}_{\text{adm}} \text{ such that } \tilde{\Phi}_\lambda(\boldsymbol{u}, \mu) = \begin{pmatrix} \lambda(\boldsymbol{J}\boldsymbol{f} - \boldsymbol{M}(\mu)\boldsymbol{u}_0) \\ \boldsymbol{s}_{\text{obs}} \end{pmatrix}. \tag{AAO}$$

As we mentioned in Section 4.3.1 this can also be cast as an optimization problem:

$$\min_{(\boldsymbol{u},\mu)\in\boldsymbol{Y}\times\boldsymbol{P}_{\text{adm}}} \tilde{J}_\lambda(\boldsymbol{u}, \mu)\,,$$

with

$$\tilde{J}_\lambda \colon \boldsymbol{Y} \times \boldsymbol{P}_{\text{adm}} \longrightarrow \mathbb{R}\,, \quad \tilde{J}_\lambda(\boldsymbol{u}, \mu) = \frac{1}{2}\|\Psi\boldsymbol{u} - \boldsymbol{s}_{\text{obs}}\|_{\boldsymbol{S}}^2 + \frac{\lambda^2}{2}\|\tilde{\mathfrak{L}}(\mu)\boldsymbol{u} - \boldsymbol{J}\boldsymbol{f}\|_{0,[0,T]\times\Omega}^2, \quad \lambda \in \mathbb{R}_+\,. \tag{5.4}$$

In Section 4.1.1, we saw that the reduced formulation (RED) can be cast as an optimization problem with a PDE constraint. The functional in (5.4) is a relaxed version of that. The choice of the parameter λ controls the influence of the second term on the optimization. For increasing λ the relaxed version transitions to the reduced formulation. The potential merits of the all-at-once formulation in FWI were first shown by Herrmann and Leeuwen in frequency domain in [vLH13]. In a simple example the authors show that their version of the functional (5.4) yields a more convex minimization problem without local minima. Publications exploring this ansatz in FWI with various optimization strategies have become a field of interest in recent years, see [vLH15]. A review, also in frequency domain, about all-at-once methods in conjunction with different times of regularization was carried out in [GAO20]. More recent studies of algorithms that are closely related ansatzes in time domain can be found in [GAO21, YZ23]. All-at-once methods are also well-known in the mathematical inverse problem community. A general mathematic overview as well as convergence results of some typical iterative nonlinear inversion algorithms for problems with all-at-once formulation can be found in [Kal16]. A sound mathematical framework for operators involved in the all-at-once formulation for FWI in time domain was established in [Rie21]. There it the author showed that all derivatives of higher order than two vanish for materials that depend linearly on the parameters (in our notation this would mean that $\mu \longmapsto \boldsymbol{M}(\mu)$ is linear). This could be beneficial in the inversion process since the problem is less nonlinear.

Remark 9. *If we recall the definitions outlined at the beginning of Section 4.1.1, the objective is to simultaneously minimize both the data and the model error (in a slightly different version).*

Remark 10. *Note that in the geophysical literature, always the 'strong' version of the differential operator is considered without the integration operator. The functional considered there is*

$$\frac{1}{2}\|\Psi\boldsymbol{u} - \boldsymbol{s}_{\mathrm{obs}}\|_{\boldsymbol{S}}^2 + \frac{\lambda^2}{2}\|\boldsymbol{\mathfrak{L}}(\mu)\boldsymbol{u} - \boldsymbol{f}\|_{0,[0,T]\times\Omega}^2, \quad \lambda \in \mathbb{R}_+ .$$

For our inversions in this thesis we stick to the integrated version introduced in the beginning of this section.

5.1.2 Theory of all-at-once

We use results from [Rie21] which, for convenience of the reader, we summarize in the following.

Definition 7. *Let $\boldsymbol{Q} \in \mathcal{L}(X)$ and $\boldsymbol{A}_{-1}\boldsymbol{X} \longrightarrow \boldsymbol{X}_{-1}$ from Remark 8. Define*

$$\tilde{F}_\lambda \colon (\boldsymbol{Y} \times \mathcal{B}) \subset (\boldsymbol{Y} \times \mathcal{S}(\boldsymbol{X})) \longrightarrow \boldsymbol{Y}_{-1} \times \boldsymbol{S},\ (\boldsymbol{u}, \boldsymbol{\Pi}) \longmapsto \begin{pmatrix} \lambda(\boldsymbol{\Pi}\boldsymbol{u} + (\boldsymbol{\Pi}\boldsymbol{Q} + \boldsymbol{A}_{-1})\boldsymbol{J}\boldsymbol{u}) \\ \Psi\boldsymbol{u} \end{pmatrix},$$

where $\mathcal{S}(\boldsymbol{X}) = \{P \in \mathcal{L}(\boldsymbol{X}) \colon P^* = P\}$ and $\mathcal{B} = \{B \in \mathcal{S}(\boldsymbol{X}) \colon \beta_- \|x\|_X^2 \le (Bx, x)_{0,\Omega} \le \beta_+ \|x\|_X^2\}$ for given $0 < \beta_- < \beta_+ < \infty$.

Theorem 7. *The mapping $\tilde{F}_\lambda$ from Definition 7 is Fréchet differentiable at any $(\boldsymbol{u}, \boldsymbol{\Pi}) \in \mathrm{int}(\boldsymbol{Y} \times \mathcal{B})$ with*

$$\tilde{F}_\lambda'(\boldsymbol{u}, \boldsymbol{\Pi})[\boldsymbol{h}, \boldsymbol{H}] = \begin{pmatrix} \lambda(\boldsymbol{\Pi}(\boldsymbol{I} + \boldsymbol{Q}\boldsymbol{J})\boldsymbol{h} + \boldsymbol{H}(\boldsymbol{I} + \boldsymbol{Q}\boldsymbol{J})\boldsymbol{u} + \boldsymbol{A}_{-1}\boldsymbol{J}\boldsymbol{h} - \boldsymbol{H}\boldsymbol{u}_0) \\ \Psi\boldsymbol{h} \end{pmatrix}.$$

Proof. See [Rie21, Lemma 3.5]. □

Now we apply Theorem 7 to our setting. Note that, as before, we write $\tilde{\boldsymbol{\mathfrak{L}}}_\mu$ instead of $\tilde{\boldsymbol{\mathfrak{L}}}(\mu)$. We do this to save space and emphasize that in these cases μ is fixed.

Lemma 4. *Recall the definitions* $\boldsymbol{M}'_{\mu,\boldsymbol{u}}$ *from Lemma 2,* $\boldsymbol{P} = L^\infty(\boldsymbol{X})^3$ *and* $\tilde{\boldsymbol{\mathfrak{L}}}$ *is from (5.3). The all-at-once operator* $\tilde{\Phi}$ *is Fréchet differentiable at any* $(\boldsymbol{u}, \mu) \in \operatorname{int}(\boldsymbol{Y} \times \boldsymbol{P}_{\mathrm{adm}})$*. The Fréchet derivative is given by*

$$\tilde{\Phi}'_\lambda(\boldsymbol{u}, \mu) \colon \boldsymbol{Y} \times \boldsymbol{P} \longrightarrow \boldsymbol{Y}_{-1} \times \boldsymbol{S},\ (\boldsymbol{h}, \eta) \longmapsto \begin{pmatrix} \lambda \left(\tilde{\boldsymbol{\mathfrak{L}}}_\mu \boldsymbol{h} + \tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}[\eta] - \boldsymbol{M}'_{\mu,\boldsymbol{u}_0}[\eta] \right) \\ \Psi \boldsymbol{h} \end{pmatrix},$$

where we define

$$\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}[\eta] = \boldsymbol{M}'_{\mu,(\boldsymbol{I}+\boldsymbol{D}\boldsymbol{J})\boldsymbol{u}}[\eta]\,.$$

The explicit representation is

$$\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}[\eta] = \begin{pmatrix} \eta_\varrho \boldsymbol{u}_{\boldsymbol{v}} \\ \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2} \left(-\frac{1}{\varrho}\eta_\varrho - \frac{1}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} + \frac{\alpha}{1+\alpha\tau_{\mathrm{p}}}\eta_{\tau_{\mathrm{p}}} \right) \boldsymbol{u}_{p_0} \\ \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2 \tau_{\mathrm{p}}} \left(-\frac{1}{\varrho}\eta_\varrho - \frac{1}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} - \frac{1}{(1+\tau_{\mathrm{p}}\alpha)}\eta_{\tau_{\mathrm{p}}} \right) (\boldsymbol{u}_{p_1} + \tau_1^{-1}\boldsymbol{u}^\uparrow_{p_1}) \\ \vdots \\ \frac{1+\alpha\tau_{\mathrm{p}}}{\varrho v_{\mathrm{p}}^2 \tau_{\mathrm{p}}} \left(-\frac{1}{\varrho}\eta_\varrho - \frac{1}{v_{\mathrm{p}}}\eta_{v_{\mathrm{p}}} - \frac{1}{(1+\tau_{\mathrm{p}}\alpha)}\eta_{\tau_{\mathrm{p}}} \right) (\boldsymbol{u}_{p_L} + \tau_L^{-1}\boldsymbol{u}^\uparrow_{p_L}) \end{pmatrix}$$

with $\boldsymbol{u} = (\boldsymbol{u}_{\boldsymbol{v}}, \boldsymbol{u}_{p_0}, \dots, \boldsymbol{u}_{p_L}), \boldsymbol{h} = (\boldsymbol{h}_{\boldsymbol{v}}, \boldsymbol{h}_{p_0}, \dots, \boldsymbol{h}_{p_L}), \mu = (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}), \eta = (\eta_\varrho, \eta_{v_{\mathrm{p}}}, \eta_{\tau_{\mathrm{p}}})\,, \boldsymbol{u}^\uparrow(t) = (\boldsymbol{J}\boldsymbol{u})(t) = \int_0^t \boldsymbol{u}(s)\ \mathrm{d}s$*, and the* p_l *component of* $\boldsymbol{u}^\uparrow$ *is denoted by* $\boldsymbol{u}^\uparrow_{p_l}$*, for* $l = 0, \dots, L$*.*

Proof. Using the representation in Definition 6, we can write $\tilde{\Phi}_\lambda(\boldsymbol{u}, (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})) = \tilde{F}_\lambda(\boldsymbol{u}, \boldsymbol{M}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))$ and we apply Theorem 7 with $\boldsymbol{Q} = \boldsymbol{D} \in \mathcal{L}(X)$ and $\boldsymbol{M}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}) \in \mathcal{B}$ for $(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}) \in \boldsymbol{P}_{\mathrm{adm}}$. Using the chain rule for Fréchet derivatives we obtain

$$\tilde{\Phi}'_\lambda(\boldsymbol{u}, (\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))[\boldsymbol{h}, \eta] = \tilde{F}'_\lambda(\boldsymbol{u}, \boldsymbol{M}(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}}))[\boldsymbol{h}, \boldsymbol{M}'(\varrho, v_{\mathrm{p}}, \tau_{\mathrm{p}})[\eta_\varrho, \eta_v, \eta_{\tau_{\mathrm{p}}}]]\,.$$

Plugging the Fréchet derivative of $\boldsymbol{M}$ from Lemma 1 in the formula into Theorem 7 yields the representation. □

Before we calculate the adjoint of $\tilde{\Phi}'_\lambda$, we calculate $\boldsymbol{J}^*$.

Lemma 5. *Like before we use the notation* $\boldsymbol{Y} = L^2([0,T], \boldsymbol{X})$*. Consider the operator*

$$\boldsymbol{J} \colon \boldsymbol{Y} \longrightarrow \boldsymbol{Y},\ \boldsymbol{u} \longmapsto (\boldsymbol{J}\boldsymbol{u})(t) = \int_0^t \boldsymbol{u}(s)\ \mathrm{d}s\,.$$

This operator is bounded and its adjoint is

$$\boldsymbol{J}^* \colon \boldsymbol{Y} \longrightarrow \boldsymbol{Y},\ \boldsymbol{u} \longmapsto (\boldsymbol{J}^*\boldsymbol{u})(t) = \int_t^T \boldsymbol{u}(s)\ \mathrm{d}s\,.$$

Proof. With $\chi_I \colon \mathbb{R} \longrightarrow \mathbb{R},\ I \subset \mathbb{R}$, we denote the indicator function. Then, we get

$$\|\boldsymbol{J}\boldsymbol{u}\|^2_{0,[0,T]\times\Omega} = \int_0^T |\chi_{[0,t]}(s)|^2 \|\boldsymbol{u}(s)\|^2_{0,\Omega}\ \mathrm{d}s \le \int_0^T \|\boldsymbol{u}(s)\|^2_{0,\Omega}\ \mathrm{d}s = \|\boldsymbol{u}\|^2_{0,[0,T]\times\Omega}\,,$$

which shows boundedness. Further, we can calculate

$$\begin{aligned}(\boldsymbol{J}\boldsymbol{u},\boldsymbol{v})_{0,[0,T]\times\Omega} &= \int_0^T \int_\Omega \left(\int_0^T \chi_{[0,t]}(s)\boldsymbol{u}(s,\boldsymbol{x})\ \mathrm{d}s\right)\cdot \boldsymbol{v}(t,\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}t\\ &= \int_0^T \int_\Omega \boldsymbol{u}(s,\boldsymbol{x})\cdot\left(\int_0^T \chi_{[0,t]}(s)\boldsymbol{v}(t,\boldsymbol{x})\ \mathrm{d}t\right)\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}s\\ &= \int_0^T \int_\Omega \boldsymbol{u}(s,\boldsymbol{x})\cdot\left(\int_s^T \boldsymbol{v}(t,\boldsymbol{x})\ \mathrm{d}t\right)\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}s\\ &= (\boldsymbol{u},\boldsymbol{J}^*\boldsymbol{v})_{0,[0,T]\times\Omega}\end{aligned}$$

□

Analogous to before we define $\boldsymbol{u}^{\downarrow}(t) := (\boldsymbol{J}^*\boldsymbol{u})(t) = \int_t^T \boldsymbol{u}(s)\ \mathrm{d}s$. Although we could use the abstract representation from [Rie21] to derive the formula for the adjoint of $\tilde{\Phi}'$, we calculate the representation ourselves in a straightforward way.

Lemma 6. *With the definition of $\boldsymbol{M}'^*$ from Lemma 2 we define $\tilde{\mathfrak{L}}'^*_{\mu,\boldsymbol{u}}\colon \boldsymbol{Y}'_{-1} \longrightarrow L^1(\Omega) \subset \boldsymbol{P}'$*

$$\tilde{\mathfrak{L}}'^*_{\mu,\boldsymbol{u}}[\boldsymbol{y}] := \boldsymbol{M}'^*_{\mu,(\boldsymbol{I}+\boldsymbol{D}\boldsymbol{J})\boldsymbol{u}}[\boldsymbol{y}] = \begin{pmatrix} R - \frac{\nu(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})}{\varrho}\tilde{T}_0 - \frac{\tilde{\nu}(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})}{\varrho}\sum_{l=1}^L \tilde{T}_l\\ -\frac{2\nu(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})}{\upsilon_\mathrm{p}}\tilde{T}_0 - \frac{2\tilde{\nu}(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})}{\upsilon_\mathrm{p}}\sum_{l=1}^L \tilde{T}_l\\ \frac{\nu(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})\alpha}{1+\alpha\tau_\mathrm{p}}\tilde{T}_0 - \frac{\tilde{\nu}(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})}{(1+\alpha\tau_\mathrm{p})\tau_\mathrm{p}}\sum_{l=1}^L \tilde{T}_l\end{pmatrix}. \tag{5.5}$$

The adjoint of $\tilde{\Phi}'_\lambda(\boldsymbol{u},\mu)$ then can be expressed as

$$\tilde{\Phi}'^*_\lambda(\boldsymbol{u},\mu)\colon \boldsymbol{Y}'_{-1}\times\boldsymbol{S} \longrightarrow \boldsymbol{Y}\times\boldsymbol{P}',\ (\boldsymbol{y},\boldsymbol{s}) \longmapsto \begin{pmatrix}\lambda\tilde{\mathfrak{L}}^\star_\mu\boldsymbol{y} + \Psi^*\boldsymbol{s}\\ \lambda(\tilde{\mathfrak{L}}'^*_{\mu,\boldsymbol{u}} - \boldsymbol{M}'^*_{\mu,\boldsymbol{u}_0})[\boldsymbol{y}]\end{pmatrix}.$$

The notation is from Lemma 4 with $\boldsymbol{y} = (\boldsymbol{y}_{\boldsymbol{v}},\boldsymbol{y}_{p_0},\boldsymbol{y}_{p_1},\dots,\boldsymbol{y}_{p_L})$ and

$$R(\boldsymbol{x}) = (\boldsymbol{y}_{\boldsymbol{v}}(\cdot,\boldsymbol{x}),\boldsymbol{u}_{\boldsymbol{v}}(\cdot,\boldsymbol{x}))_{0,[0,T]}\,,\ \boldsymbol{x}\in\Omega, \tilde{T}_0 := \left(\boldsymbol{u}_{p_0},\boldsymbol{y}_{p_0}\right)_{0,[0,T]}\,,\ \tilde{T}_l := \left(\boldsymbol{u}_{p_l} + \tau_l^{-1}\boldsymbol{u}^{\uparrow}_{p_l},\boldsymbol{y}_{p_l}\right)_{0,[0,T]}\,.$$

The operator $\tilde{\mathfrak{L}}^\star_\mu\colon \boldsymbol{Y}'_{-1}\longrightarrow\boldsymbol{Y},\ \boldsymbol{y}\longmapsto \boldsymbol{M}(\mu)\boldsymbol{y} + (\boldsymbol{M}(\mu)\boldsymbol{D} - \boldsymbol{A}_{-1})\boldsymbol{J}^*\boldsymbol{y}$ has the representation

$$\tilde{\mathfrak{L}}^\star_\mu\boldsymbol{y} = \begin{pmatrix}\varrho\boldsymbol{y}_{\boldsymbol{v}} + \nabla_-\boldsymbol{y}^{\downarrow}_p\\ \nu(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})\boldsymbol{y}_{p_0} + \operatorname{div}_-\boldsymbol{y}^{\downarrow}_{\boldsymbol{v}}\\ \tilde{\nu}(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})(\boldsymbol{y}_{p_1} + \tau_1^{-1}\boldsymbol{y}^{\downarrow}_{p_1}) + \operatorname{div}_-\boldsymbol{y}^{\downarrow}_{\boldsymbol{v}}\\ \vdots\\ \tilde{\nu}(\varrho,\upsilon_\mathrm{p},\tau_\mathrm{p})(\boldsymbol{y}_{p_L} + \tau_L^{-1}\boldsymbol{y}^{\downarrow}_{p_L}) + \operatorname{div}_-\boldsymbol{y}^{\downarrow}_{\boldsymbol{v}}\end{pmatrix} \tag{5.6}$$

with $\boldsymbol{y}_p := \sum_{l=0}^L \boldsymbol{y}_{p_l}$. The differential div_- and ∇_- are the extensions of the operators that come with the extension of $\boldsymbol{A}_{-1}$. Here, we denote the dual spaces of Banach spaces with $'$.

Proof. We need to show

$$\left\langle \tilde{\Phi}'_\lambda(\boldsymbol{u},\mu)\begin{bmatrix}\boldsymbol{h}\\ \eta\end{bmatrix}, \begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{s}\end{pmatrix}\right\rangle_{(\boldsymbol{Y}_{-1}\times\boldsymbol{S})\times(\boldsymbol{Y}'_{-1}\times\boldsymbol{S})} = \left\langle \begin{pmatrix}\boldsymbol{h}\\ \eta\end{pmatrix}, \tilde{\Phi}'^*_\lambda(\boldsymbol{u},\mu)\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{s}\end{bmatrix}\right\rangle_{(\boldsymbol{Y}\times\boldsymbol{P})\times(\boldsymbol{Y}\times\boldsymbol{P}')}.$$

Since $\tilde{\Phi}'_\lambda(\boldsymbol{u}, \mu)$ is linear, we can compute the components separately. First, we compute

$$\left\langle \tilde{\Phi}'_\lambda(\boldsymbol{u}, \mu) \begin{bmatrix} \boldsymbol{h} \\ 0 \end{bmatrix}, \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{s} \end{pmatrix} \right\rangle_{(\boldsymbol{Y}_{-1} \times \boldsymbol{S}) \times (\boldsymbol{Y}'_{-1} \times \boldsymbol{S})} = \left\langle \lambda \tilde{\boldsymbol{\mathfrak{L}}}_\mu \boldsymbol{h}, \boldsymbol{y} \right\rangle_{\boldsymbol{Y}_{-1} \times \boldsymbol{Y}'_{-1}} + (\Psi \boldsymbol{h}, \boldsymbol{s})_{\boldsymbol{S}} \tag{5.7}$$

$$= \left(\boldsymbol{h}, \tilde{\boldsymbol{\mathfrak{L}}}^\star_\mu \boldsymbol{y} + \Psi^* \boldsymbol{s} \right)_{0,[0,T] \times \Omega} .$$

Plugging in the second component yields

$$\left\langle \tilde{\Phi}'_\lambda(\boldsymbol{u}, \mu) \begin{bmatrix} \boldsymbol{0} \\ \eta \end{bmatrix}, \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{s} \end{pmatrix} \right\rangle_{(\boldsymbol{Y}_{-1} \times \boldsymbol{S}) \times (\boldsymbol{Y}'_{-1} \times \boldsymbol{S})} = \left\langle \begin{pmatrix} \boldsymbol{0} \\ \eta \end{pmatrix}, \tilde{\Phi}'^*_\lambda(\boldsymbol{u}, \mu) \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{s} \end{bmatrix} \right\rangle_{(\boldsymbol{Y} \times \boldsymbol{P}) \times (\boldsymbol{Y} \times \boldsymbol{P}')} . \tag{5.8}$$

Since $\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}}[\eta] - \boldsymbol{M}'_{\mu,\boldsymbol{u}_0}[\eta] \in \boldsymbol{Y} \subset \boldsymbol{Y}_{-1}$ we can write

$$\left\langle \lambda(\tilde{\boldsymbol{\mathfrak{L}}}'_{\mu,\boldsymbol{u}} - \boldsymbol{M}'_{\mu,\boldsymbol{u}_0})[\eta], \boldsymbol{y} \right\rangle_{\boldsymbol{Y}_{-1} \times \boldsymbol{Y}'_{-1}} = \left\langle \eta, \lambda(\tilde{\boldsymbol{\mathfrak{L}}}'^*_{\mu,\boldsymbol{u}} - \boldsymbol{M}'^*_{\mu,\boldsymbol{u}_0})[\boldsymbol{y}] \right\rangle_{L^\infty(\Omega)^3 \times (L^\infty(\Omega)^3)'} .$$

Combining the two equations (5.7) and (5.8) yields the asserted representation. For the representation of (5.5) we use that $\boldsymbol{A}^*_{-1} = -\boldsymbol{A}_{-1}$, see the footnote on page 9 in [Rie21]. Note that we can continuously embed $\boldsymbol{Y}$ in $\boldsymbol{Y}_{-1}$. We get $\boldsymbol{Y}'_{-1} \subset \boldsymbol{Y}' \simeq \boldsymbol{Y}$ which makes the evaluation of $-\boldsymbol{A}_{-1}$ well-defined. Moreover, since in Lemma 2 we showed that $\boldsymbol{M}'^*_{\mu,\boldsymbol{z}}[\boldsymbol{y}] \in L^1(\Omega)$ for $\boldsymbol{z}, \boldsymbol{y} \in \boldsymbol{Y}$ we know that the range of $\tilde{\Phi}'^*_\lambda$ is indeed in $\boldsymbol{Y} \times \boldsymbol{P}'$. □

We can also show that – just like in Theorem 3 for the reduced formulation – the all-at-once formulation (AAO) is ill-posed, too.

Theorem 8. *The inverse problem $\tilde{\Phi}(\boldsymbol{u}, \mu) = \left(\boldsymbol{0}, \boldsymbol{s}_{\text{obs}}\right)^\top$ is locally ill-posed at any interior point $(\boldsymbol{u}, \mu) \in \mathcal{D}(\tilde{\Phi})$.*

Proof. See [Rie21, Proposition 4.2] with minor adaptations. □

Remark 11. *At the beginning of this section we stated that the all-at-once formulation is considered to be less nonlinear than the reduced formulation. The reasoning for this can be seen in Theorem 7, since $\boldsymbol{u}$ and $\boldsymbol{\Pi}$ only appear linearly. Therefore, all derivatives of $\tilde{F}_\lambda$ higher than two vanish. Note that if we use a parameterization $\mu \longrightarrow \boldsymbol{M}(\mu)$ that is not linear, the derivatives higher than two do not vanish completely. Even in the case of linear parameterizations higher derivatives in the reduced formulation will never vanish.*

From now on we assume $\boldsymbol{u}_0 = \boldsymbol{0}$, this makes calculations shorter and reflects our usual setups in inversions.

5.2 Discretization of the operators

In this section we briefly show how the operators $\tilde{\boldsymbol{\mathfrak{L}}}$, see (5.3), and $\tilde{\boldsymbol{\mathfrak{L}}}^\star$, see (5.6) are evaluated numerically. From now on, we do not denote the dependence of the operators on μ explicitly in most contexts. As before, operators with subscript h are representations of the operators on finite dimensional linear subspaces. An operator that is underlined stands for the representation of that operator with respect to our finite element basis. Recalling the definition in this section, we again denote our discontinuous Galerkin space by $\boldsymbol{X}_h$ from (3.1) and $\boldsymbol{Y}_h = \boldsymbol{X}_h^N$, where N is the number of time steps in our temporal discretization.

Evaluation of the operators $\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}$ and $\tilde{\boldsymbol{\mathfrak{L}}}^{\star}_{h,\mu}$

Unlike in Section 4.1.1 to evaluate the forward operators in time we do not solve a semi-discrete ordinary differential equation. Instead, we just have to evaluate an integral. Therefore, we define a discrete approximation of the integral operator $\boldsymbol{J}$. For that, we use a composite one-step integration scheme with two weights $(w_1, w_2) \in \{(1,0),(0,1),(0.5,0.5)\}$. Here, $(1,0)$ corresponds to the left rectangular rule, $(0,1)$ to the right rectangular rule, and $(0.5,0.5)$ to the trapezoidal rule. The temporal step size is called $\triangle t$, for a chosen value of N its value is $\triangle t = T/(N-1)$. We denote the operator that evaluates the integral at time $t_n = n \cdot \triangle t,\ n = 0,\dots,N-1$ by

$$\boldsymbol{J}^n_{h,\triangle t} \colon \boldsymbol{Y}_h \longrightarrow \boldsymbol{X}_h,\ \boldsymbol{u} \longmapsto \left(w_1 \boldsymbol{u}^0_h + \sum_{i=1}^{n} \boldsymbol{u}^i_h + w_2 \boldsymbol{u}^{n+1}_h \right) \triangle t\,.$$

For the full integral operator this yields the matrix representation

$$\underline{\boldsymbol{J}}_{h,\triangle t} = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ w_1 & w_2 & \ddots & 0 & 0 \\ w_1 & 1 & w_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & w_2 & 0 \\ w_1 & 1 & \cdots & 1 & w_2 \end{pmatrix} \in \mathbb{R}^{\dim \boldsymbol{Y}_h \times \dim \boldsymbol{Y}_h}$$

The numerical evaluation of the operator $\tilde{\boldsymbol{\mathfrak{L}}}$ at time t_n solves the variational problem

$$\text{find } \tilde{\boldsymbol{u}}_{h,n} \in \boldsymbol{X}_h \text{ s.t. } (\tilde{\boldsymbol{u}}_{h,n}, \boldsymbol{y}_h)_{0,\Omega} = \big(\boldsymbol{M}_h \boldsymbol{u}_{h,n} + (\boldsymbol{M}_h \boldsymbol{D}_h + \boldsymbol{M}_h \boldsymbol{A}_h) \boldsymbol{J}^n_{h,\triangle t} \boldsymbol{u}_h, \boldsymbol{y}_h\big)_{0,\Omega}\,, \forall \boldsymbol{y}_h \in \boldsymbol{X}_h\,. \quad (5.9)$$

With $\underline{\boldsymbol{A}}_h, \underline{\boldsymbol{M}}_h, \underline{\boldsymbol{D}}_h, \underline{\boldsymbol{G}}_h \in \mathbb{R}^{\dim \boldsymbol{X}_h \times \dim \boldsymbol{X}_h}$ we denote the matrices corresponding to the linear operators as defined in (3.3). Any element $\boldsymbol{x}_h \in \boldsymbol{X}_h$ can be represented via $\boldsymbol{x}_h = \sum_{i=1}^{\dim \boldsymbol{X}_h} \underline{\boldsymbol{x}}^i_h \boldsymbol{\varphi}_i \in \boldsymbol{X}_h$, and therefore interpreted as the coordinate vector $\underline{\boldsymbol{x}}_h = (\underline{\boldsymbol{x}}^1_h, \dots, \underline{\boldsymbol{x}}^{\dim \boldsymbol{X}_h}_h) \in \mathbb{R}^{\dim \boldsymbol{X}_h}$. Using this representation for $\tilde{\boldsymbol{u}}_{h,n}, \boldsymbol{u}_{h,n} \in \boldsymbol{X}_h$ and testing (5.9) with $\boldsymbol{y}_h = \boldsymbol{\varphi}_i$ we can write the fully discretized version of (5.9) as the linear system

$$\underline{\boldsymbol{G}}_h \underline{\tilde{\boldsymbol{u}}}_{h,n} = \underline{\boldsymbol{M}}_h \underline{\boldsymbol{u}}_{h,n} + (\underline{\boldsymbol{A}}_h + \underline{\boldsymbol{M}}_h \underline{\boldsymbol{D}}_h) \underline{\boldsymbol{J}}^n_{h,\triangle t} \underline{\boldsymbol{u}}_h\,.$$

Solving this system comes with little cost since $\underline{\boldsymbol{G}}_h$ is a block diagonal positive definite matrix. Define

$$\underline{\boldsymbol{T}} := (\underline{\boldsymbol{A}}_h + \underline{\boldsymbol{M}}_h \underline{\boldsymbol{D}}_h) \triangle t\,,\ \underline{\boldsymbol{B}}_{w_i} := \underline{\boldsymbol{M}}_h + w_i \underline{\boldsymbol{T}}\,,\quad \underline{\boldsymbol{C}}_{w_i} := w_i \underline{\boldsymbol{T}} - \underline{\boldsymbol{M}}_h \ \text{ for } i = 1,2.$$

The right-hand sides of (5.9) for all time steps $n = 0,\dots,N$ can be expressed as matrix

$$\widetilde{\underline{\boldsymbol{K}}}_h = \begin{pmatrix} \underline{\boldsymbol{M}}_h & 0 & \cdots & \cdots & \cdots & 0 \\ w_1 \underline{\boldsymbol{T}} & \underline{\boldsymbol{B}}_{w_2} & \ddots & \ddots & \ddots & \vdots \\ w_1 \underline{\boldsymbol{T}} & \underline{\boldsymbol{T}} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ w_1 \underline{\boldsymbol{T}} & \underline{\boldsymbol{T}} & \cdots & \cdots & \underline{\boldsymbol{T}} & \underline{\boldsymbol{B}}_{w_2} \end{pmatrix}$$

For the adjoint problem we use

$$\boldsymbol{J}^{*,n}_{\triangle t}\colon \boldsymbol{X}_h^N \longrightarrow \boldsymbol{X}_h,\ \boldsymbol{u} \longmapsto \left(w_2\boldsymbol{u}_{h,N-1} + \sum_{i=1}^{n-1}\boldsymbol{u}_{h,N-1-i} + w_1\boldsymbol{u}_{h,N-1-n}\right)\triangle t\,,$$

and we get

$$\widetilde{\underline{\boldsymbol{K}}}^*_h = \begin{pmatrix} \underline{\boldsymbol{B}}^*_{w_2} & \underline{\boldsymbol{T}}^* & \cdots & \cdots & \underline{\boldsymbol{T}}^* & w_2\underline{\boldsymbol{T}}^* \\ 0 & \underline{\boldsymbol{B}}^*_{w_2} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \underline{\boldsymbol{T}}^* & \vdots \\ \vdots & \ddots & \ddots & \ddots & \underline{\boldsymbol{B}}^*_{w_2} & w_2\underline{\boldsymbol{T}}^* \\ 0 & \cdots & \cdots & \cdots & 0 & \underline{\boldsymbol{M}}_h \end{pmatrix} \in \mathbb{R}^{\dim \boldsymbol{Y}_h \times \dim \boldsymbol{Y}_h}\,.$$

Note that $\underline{\boldsymbol{D}}^*_h = \underline{\boldsymbol{D}}_h$ and $\underline{\boldsymbol{A}}^*_h$ is defined as in Eq. (3.2). Further, define

$$\underline{\boldsymbol{G}}^N_h = \operatorname{diag}(\underline{\boldsymbol{G}}_h, \dots, \underline{\boldsymbol{G}}_h) \in \mathbb{R}^{\dim \boldsymbol{Y}_h \times \dim \boldsymbol{Y}_h}\,.$$

Then, we can define the discrete operators

$$\tilde{\mathfrak{L}}_{h,\mu} = (\underline{\boldsymbol{G}}^N_h)^{-1}\widetilde{\underline{\boldsymbol{K}}}_h, \quad \tilde{\mathfrak{L}}^\star_{h,\mu} = (\underline{\boldsymbol{G}}^N_h)^{-1}\widetilde{\underline{\boldsymbol{K}}}^*_h\,.$$

Later on we also want to invert the operators $\tilde{\mathfrak{L}}_{h,\mu}, \tilde{\mathfrak{L}}^\star_{h,\mu}$. Therefore, we want to rewrite them in a way that makes it easier to calculate the inverse. Modifying $\widetilde{\underline{\boldsymbol{K}}}_h, \widetilde{\underline{\boldsymbol{K}}}^*_h$ with the Gaussian elimination matrices $\underline{\boldsymbol{S}}^{-1}, \underline{\boldsymbol{S}}^{-\top}$ defined by

$$\underline{\boldsymbol{S}}^{-1} = \begin{pmatrix} \mathbf{1} & 0 & \cdots & \cdots & 0 \\ -\mathbf{1} & \mathbf{1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \mathbf{1} & \vdots \\ 0 & 0 & \cdots & -\mathbf{1} & \mathbf{1} \end{pmatrix} \text{ with inverse } \underline{\boldsymbol{S}} = \begin{pmatrix} \mathbf{1} & 0 & \cdots & \cdots & 0 \\ \mathbf{1} & \mathbf{1} & 0 & \cdots & 0 \\ \mathbf{1} & \mathbf{1} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \mathbf{1} & \vdots \\ \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} & \mathbf{1} \end{pmatrix}$$

we get the block banded matrices

$$\underline{\boldsymbol{K}}_h = \underline{\boldsymbol{S}}^{-1}\widetilde{\underline{\boldsymbol{K}}}_h = \begin{pmatrix} \underline{\boldsymbol{M}}_h & 0 & \cdots & \cdots & 0 \\ \underline{\boldsymbol{C}}_{w_1} & \underline{\boldsymbol{B}}_{w_2} & 0 & \cdots & 0 \\ 0 & \underline{\boldsymbol{C}}_{w_1} & \underline{\boldsymbol{B}}_{w_2} & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \underline{\boldsymbol{C}}_{w_1} & \underline{\boldsymbol{B}}_{w_2} \end{pmatrix}$$

and

$$\underline{\boldsymbol{K}}^*_h = \underline{\boldsymbol{S}}^{-\top}\widetilde{\underline{\boldsymbol{K}}}^*_h = \begin{pmatrix} \underline{\boldsymbol{B}}^*_{w_1} & \underline{\boldsymbol{C}}^*_{w_2} & 0 & \cdots & 0 \\ 0 & \underline{\boldsymbol{B}}^*_{w_1} & \ddots & \ddots & \vdots \\ 0 & \ddots & \underline{\boldsymbol{B}}^*_{w_1} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \underline{\boldsymbol{C}}^*_{w_2} \\ 0 & 0 & \cdots & 0 & \underline{\boldsymbol{M}}_h \end{pmatrix}.$$

We finally can define

$$\underline{\boldsymbol{\mathfrak{L}}}_h = (\underline{\boldsymbol{G}}_h^N)^{-1}\underline{\boldsymbol{K}}_h,\ \underline{\boldsymbol{\mathfrak{L}}}^*_h = (\underline{\boldsymbol{G}}_h^N)^{-1}\underline{\boldsymbol{K}^*}_h$$

yielding

$$\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} = (\underline{\boldsymbol{G}}_h^N)^{-1}\underline{\boldsymbol{S}}\,\underline{\boldsymbol{\mathfrak{L}}}_h,\ \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} = (\underline{\boldsymbol{G}}_h^N)^{-1}\underline{\boldsymbol{S}}^\top\underline{\boldsymbol{\mathfrak{L}}}^*_h\,.$$

Now, to invert $\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}$ we can invert $\underline{\boldsymbol{\mathfrak{L}}}_h$ which is the matrix representation of the Runge-Kutta methods applied to semi-discrete (that is, discretized with dG in space) version of the initial value problem (2.11). The Runge-Kutta method corresponds to the choice of the $w = (w_1, w_2)$, that is, explicit Euler for $w = (0, 1)$, implicit Euler for $w = (1, 0)$, and implicit trapezoid rule for $w = (0.5, 0.5)$. To invert $\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}$, for a given right-hand side $\underline{\boldsymbol{b}} \in \boldsymbol{Y}_h$, we can calculate $\underline{\boldsymbol{u}}_h = \underline{\boldsymbol{K}}_h^{-1}\underline{\boldsymbol{b}}$ by using the forward substitution. We can calculate the $n + 1$-th entry of $\underline{\boldsymbol{u}}_h$ with

$$\begin{aligned}
&\underline{\boldsymbol{C}}_{w_1}\underline{\boldsymbol{u}}_{h,n} + \underline{\boldsymbol{B}}_{w_2}\underline{\boldsymbol{u}}_{h,n+1} = \underline{\boldsymbol{b}}_n\\
\Leftrightarrow &(w_1\underline{\boldsymbol{T}} - \underline{\boldsymbol{M}}_h)\underline{\boldsymbol{u}}_{h,n} + (\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})\underline{\boldsymbol{u}}_{h,n+1} = \underline{\boldsymbol{b}}_n\\
\Leftrightarrow &(\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})\underline{\boldsymbol{u}}_{h,n+1} = -(w_1\underline{\boldsymbol{T}} - \underline{\boldsymbol{M}}_h)\underline{\boldsymbol{u}}_{h,n} + \underline{\boldsymbol{b}}_n\\
\Leftrightarrow &(\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})(\underline{\boldsymbol{u}}_{h,n+1} - \underline{\boldsymbol{u}}_{h,n}) = -(w_1\underline{\boldsymbol{T}} - \underline{\boldsymbol{M}}_h)\underline{\boldsymbol{u}}_{h,n} - (\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})\underline{\boldsymbol{u}}_{h,n} + \underline{\boldsymbol{b}}_n\\
\Leftrightarrow &(\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})(\underline{\boldsymbol{u}}_{h,n+1} - \underline{\boldsymbol{u}}_{h,n}) = -\underline{\boldsymbol{T}}\underline{\boldsymbol{u}}_{h,n} + \underline{\boldsymbol{b}}^n\\
\Leftrightarrow &\underline{\boldsymbol{u}}_{h,n+1} = \underline{\boldsymbol{u}}_{h,n} + (\underline{\boldsymbol{M}}_h + w_2\underline{\boldsymbol{T}})^{-1}(-\underline{\boldsymbol{T}}\underline{\boldsymbol{u}}_{h,n} + \underline{\boldsymbol{b}}_n)\,,
\end{aligned}$$

and then apply the rest of the operators. The same can be done for $\tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu}$, where the inversion of $\underline{\boldsymbol{\mathfrak{L}}}^*_h$ results in a backward substitution and is equivalent to solve the adjoint wave equation (2.23) with a Runge-Kutta method. Due to what we just calculated, in this section we will always use the implicit trapezoidal rule. Now, we turn to the algorithms to invert the all-at-once problem (AAO).

5.3 Algorithms for the all-at-once formulation

In this section we will present algorithms that we have tested for the time domain formulation of the all-at-once problem. While we consider only the last one to be three successful, we still describe the others and additionally the reasons why they failed. In this way we also show how we arrived at to the final algorithm.

5.3.1 Sequential inversion

Now we discuss the inversion procedure that was presented in [vLH13] in the frequency domain. We want to adapt it to our time domain and formulation. Recall the misfit functional from (5.4):

$$J_\lambda \colon \boldsymbol{Y} \times \boldsymbol{P}_{\mathrm{adm}} \longrightarrow \mathbb{R}, \quad J_\lambda(\mu, \boldsymbol{u}) = \frac{1}{2}\|\Psi\boldsymbol{u} - \boldsymbol{s}_{\mathrm{obs}}\|^2 + \frac{\lambda^2}{2}\|\tilde{\boldsymbol{\mathfrak{L}}}(\mu)\boldsymbol{u} - \boldsymbol{J}\boldsymbol{f}\|^2$$

and the associated optimization problem

$$\min_{(\boldsymbol{u},\mu)\in\boldsymbol{Y}\times\boldsymbol{P}_{\mathrm{adm}}} J_\lambda(\boldsymbol{u}, \mu).$$

The authors in [vLH13] suggest an iterative alternating minimizing algorithm. In every nonlinear step they first solve the minimization problem in the $\boldsymbol{u}$ direction, which in our formulation is

$$\boldsymbol{u}^* = \mathrm{argmin}_{\boldsymbol{u}\in\boldsymbol{Y}}\, J_\lambda(\boldsymbol{u}, \mu) \Leftrightarrow \boldsymbol{u}^* \text{ solves } \left(\lambda^2\tilde{\boldsymbol{\mathfrak{L}}}^\star_\mu\tilde{\boldsymbol{\mathfrak{L}}}_\mu + \Psi^*\Psi\right)\boldsymbol{u}^* = \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}^\star_\mu\boldsymbol{J}\boldsymbol{f} + \Psi^*\boldsymbol{s}_{\mathrm{obs}}\,.$$

Then, a descent direction is calculated by plugging in $\boldsymbol{u}^*$ in the functional J_λ and differentiate in μ direction

$$\partial_\mu J_\lambda(\boldsymbol{u}^*, \mu)[\eta] = \lambda^2 \left(\tilde{\boldsymbol{\mathfrak{L}}}'^*_{\mu,\boldsymbol{u}^*} [\tilde{\boldsymbol{\mathfrak{L}}}_\mu \boldsymbol{u}^* - \boldsymbol{J}\boldsymbol{f}], \eta \right)_{0,\Omega} .$$

Solving the *augmented wave equation*

$$\left(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_\mu \tilde{\boldsymbol{\mathfrak{L}}}_\mu + \Psi^* \Psi \right) \boldsymbol{u}^* = \lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_\mu \boldsymbol{J}\boldsymbol{f} + \Psi^* \boldsymbol{s}_{\mathrm{obs}} \tag{5.10}$$

is very challenging in time domain. Due to the high dimensionality of $\boldsymbol{Y}_h \times \boldsymbol{Y}_h$, direct solvers are not viable. So, we turn to an iterative scheme. However, just applying, for example, the `CG` algorithm to (5.10) does not converge in a meaningful way in the sense that the residual is not reduced more than 0.01% after 1000 iterations. In our numerical experiments we saw faster convergence by applying the `CG` algorithm together with the preconditioner $(1/\lambda^2)(\tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1}$. To apply this preconditioner, we need to solve one forward and one adjoint wave equation together with integration, as described at the end of Section 5.2.

Remark 12. *In our search for a suitable method to approximate the solution of the augmented wave equation, we first implemented the equation without the integral operators just like it is done in geophysics like mentioned in Remark 10. We switched $\tilde{\boldsymbol{\mathfrak{L}}}$ for $\boldsymbol{\mathfrak{L}}$ and $\tilde{\boldsymbol{\mathfrak{L}}}^\star$ for $\boldsymbol{\mathfrak{L}}^\star$ because then, the system $\tilde{\boldsymbol{\mathfrak{L}}}^\star \tilde{\boldsymbol{\mathfrak{L}}} + \Psi^* \Psi$ is a banded block matrix with width three and hence easier to handle. For this system we implemented two preconditioners. One block Jacobi preconditioner, where we invert the diagonal blocks. And one* block symmetric successive over relaxation *(SSOR) where we first do a block forward substitution on the lower part of the matrix and then do a block backward substitution on the upper part of the matrix. These preconditioners improved the convergence of the algorithms, but the implementation was rather complicated in our setting and the convergence was still slow. In [YZ23] a similar procedure to the SSOR is done to solve a problem related to the augmented wave equation. In [GAO21] they tackle the same problem but introduce some simplifications and therefore solve a simplified version of the equation.*

Remark 13. *We introduced the expression $(1/\lambda^2)(\tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1}$ as preconditioner above. We use this preconditioner on arbitrary wave fields in the `CG` iteration. This 'destroys' the adjointness of the operators as we defined in Section 2.3. We allow the application of the forward operators on wave fields $\boldsymbol{u}_h, \boldsymbol{y}_h \in \boldsymbol{Y}_h$ that do not fulfill $\boldsymbol{u}_{h,0} = \boldsymbol{0}$ and the backward operator on wave fields that do not fulfill $\boldsymbol{y}_{h,N} = \boldsymbol{0}$. Therefore, the adjointness is violated by the difference*

$$\left(\boldsymbol{M}_h \boldsymbol{u}_{h,0}, \boldsymbol{M}_h \boldsymbol{y}_{h,0} \right)_{0,\Omega} - \left(\boldsymbol{M}_h \boldsymbol{u}_{h,N}, \boldsymbol{M}_h \boldsymbol{y}_{h,N} \right)_{0,\Omega} .$$

We saw no numerical instabilities arising from this.

Remark 14. *We can reduce the dimension by using the Woodbury formula. It states that for $T \in \mathbb{R}^{n\times n}, W \in \mathbb{R}^{r\times r}, U \in \mathbb{R}^{n\times r}, V \in \mathbb{R}^{r\times n}$, the following identity holds*

$$(T - UW^{-1}V)^{-1} = T^{-1} + T^{-1}U(W - VT^{-1}U)^{-1}VT^{-1} .$$

In the discrete version of (5.10) our goal is to find $\left(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} + \underline{\Psi^*}_h \Psi \right)^{-1}$ which is a challenge due to its very high dimension. Applying the identity to $T = \lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}, W = -\underline{\boldsymbol{I}}_h \colon \boldsymbol{S} \longrightarrow \boldsymbol{S},\ \boldsymbol{s} \longrightarrow -\boldsymbol{s},$

$U = \underline{\Psi^*{}_h}$, $V = \Psi$ yields

$$\left(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} + \underline{\Psi^*{}_h}\Psi\right)^{-1} = (\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} \\ - (\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} \underline{\Psi^*{}_h} (\underline{\boldsymbol{I}}_h + \Psi(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} \underline{\Psi^*{}_h})^{-1} \Psi (\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} .$$

With this formula we can exploit that we can apply $(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1}$. We still have to solve an equation involving $(\underline{\boldsymbol{I}}_h + \Psi(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} \underline{\Psi^*{}_h})^{-1}$. This formula reduces the problem to solving a linear equation in the seismogram space

$$(\underline{\boldsymbol{I}}_h + \Psi(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} \underline{\Psi^*{}_h}) \boldsymbol{s} = \Psi(\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1} (\lambda^2 \tilde{\boldsymbol{\mathfrak{L}}}^\star \boldsymbol{J} \boldsymbol{f} + \Psi^* \boldsymbol{s}_{\text{obs}}) .$$

Although this problem is of lower dimensionality it is unclear whether it behaves better or worse than the full equation. One iteration costs still two solutions of wave equations which is the same as the preconditioner we used for `CG`.

Algorithm 5.1 Sequential method

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess; $\boldsymbol{s}_{\text{obs}} \in \boldsymbol{S}^\Xi$ % seismograms
Ensure: $\mu^k \in \boldsymbol{P}_h$ % approximate solution of (5.2)
 1: $\mu^k \leftarrow \mu^0$
 2: while not termination do
 3:     $\mu^k_0 \leftarrow \mu^k$
 4:     $\xi \leftarrow 0$
 5:     repeat
 6:         $\xi \leftarrow \xi + 1$
 7:         $\boldsymbol{b} \leftarrow \boldsymbol{J}_{h,\Delta t} \boldsymbol{f}_\xi + \Psi^* \boldsymbol{s}_{\text{obs},\xi}$
 8:         $\mathbf{u}^{\text{init}} \leftarrow \tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu} \boldsymbol{J}_{h,\Delta t} \boldsymbol{f}_{\boldsymbol{\xi}}$
 9:         $\boldsymbol{u} \leftarrow$ CG$(\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} + \Psi^*\Psi, \boldsymbol{b}, \mathbf{u}^{\text{init}})$ % with preconditioner $(1/\lambda^2)(\tilde{\boldsymbol{\mathfrak{L}}}^\star_{h,\mu} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu})^{-1}$ which means solving one forward and one adjoint wave equation per CG iteration.
10:         $\Delta\mu^k_\xi \leftarrow \tilde{\boldsymbol{\mathfrak{L}}}'^*_{h,\mu}[\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} \boldsymbol{u} - \boldsymbol{J}_{h,\Delta t} \boldsymbol{f}_\xi]$ % search direction with the formula see Lemma 6
11:         $\alpha \leftarrow$ line search$(p)$ % Armijo rule.
12:         $\mu^k_{\xi+1} \leftarrow \mu^k_\xi + \alpha \Delta\mu^k_\xi$
13:     until $\xi > \Xi - 1$
14:     $\mu^{k+1} \leftarrow \mu^k_{\Xi-1}$
15:     $k \leftarrow k + 1$
16: end while
17: return $\mu^k$
```

We tested the algorithm on the transmission problem from Fig. 4.1 for the mono-parameter experiment for v_{p} with $L = 0$. As initial value for the `CG` algorithm in line 8 of Algorithm 5.1 we choose $\mathbf{u}^{\text{init}} = \tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu} \boldsymbol{J}_{h,\Delta t} \boldsymbol{f}$. With $\boldsymbol{u}^{\text{init}} = 0$, we saw no convergence of the material. We break the `CG` algorithm when only little in the search direction changes. After 5 `CG` iterations each, we calculate the search direction which comes with little numerical cost. If the relative change is below 1.01^5, we stop the iteration. This condition says that in average over the last 5 iterations each iteration only changed by 1%. This condition offers us a possibility to gauge when to stop the CG and not just base it on the reduction of the residual. This can be an unreliable indicator especially in systems with high condition numbers. In most iterations, the reduction of the residual was about two orders of magnitude. In this experiment we needed between 15 and 20 CG iterations for each shot. In the end we had double the cost compared to Red-CG-REGINN and more or less the same result and slower convergence. We do not show these results here, since we only want to consider this geometry in the presence of damping and this first test was for without damping ($L = 0$).

These first results show that this method is capable of finding the parameter inclusions, too. When we applied the sequential algorithm in the case of $L = 5$ with attenuation, we observed that the method chooses step sizes that are small and almost no change of the parameter values is seen in the iterative process. We tried to fix this with a line search method such as the Armijo rule (which is a line search rule, where the step size gets multiplied by the same factor after each failed line search) that allows the line search factor to get bigger in each line search iteration. However, this did not yield stable results. Thus, this ansatz was not pursued further since we wanted to find a stable algorithm even if attenuation is involved. It did show, however, that the algorithm can produce results in time domain and maybe is capable of doing so with some modifications. We turn to the algorithm we applied with success in the reduced case.

5.3.2 AAO-CG-REGINN

The first algorithm we inspect is CG-REGINN. For the first variant, we just use the formulas obtained in Lemma 6, Lemma 4 and apply the algorithm “out of the box”. The main numerical cost of this operation is evaluating $\underline{\tilde{\boldsymbol{\mathfrak{L}}}}'_{h,\mu}$ and $\underline{\tilde{\boldsymbol{\mathfrak{L}}}}'^{*}_{h,\mu}$. This makes single iterations of the algorithm cheap compared to the reduced version. In [Rie21] it was emphasized that this could be a possible advantage over other methods, since solving partial differential equations contributes the most to the numerical cost. However, just like with the augmented wave equation, no convergence is seen without preconditioning. We look at the normal equation of the Newton equation as a whole, that is

$$\tilde{\Phi}'^{*}(\boldsymbol{u},\mu)\tilde{\Phi}'(\boldsymbol{u},\mu)[\triangle\boldsymbol{u},\triangle\mu] = \tilde{\Phi}'^{*}(\mu,\boldsymbol{u})[\tilde{\Phi}(\mu,\boldsymbol{u}) - (\boldsymbol{J}\boldsymbol{f},\boldsymbol{s}_{\mathrm{obs}})]\,.$$

The system then has the form

$$\begin{pmatrix} \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}^{\star}\tilde{\boldsymbol{\mathfrak{L}}} + \Psi^*\Psi & \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}^{\star}\tilde{\boldsymbol{\mathfrak{L}}}'_{\boldsymbol{u}} \\ \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}'^{*}_{\boldsymbol{u}}\tilde{\boldsymbol{\mathfrak{L}}} & \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}'^{*}_{\mu,\boldsymbol{u}}\tilde{\boldsymbol{\mathfrak{L}}}'_{\boldsymbol{u}} \end{pmatrix} \begin{pmatrix} \triangle\boldsymbol{u} \\ \triangle\mu \end{pmatrix} = \begin{pmatrix} \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}^{\star}\left(\tilde{\boldsymbol{\mathfrak{L}}}\boldsymbol{u} - \boldsymbol{J}\boldsymbol{f}\right) + \Psi^*(\Psi\boldsymbol{u} - \boldsymbol{s}_{\mathrm{obs}}) \\ \lambda^2\tilde{\boldsymbol{\mathfrak{L}}}'^{*}_{\mu,\boldsymbol{u}}[\tilde{\boldsymbol{\mathfrak{L}}}\boldsymbol{u} - \boldsymbol{J}\boldsymbol{f}] \end{pmatrix}\,. \tag{5.11}$$

Looking at the upper part left of the system we can see that it is similar to (5.10). The methods we developed in the last section to solve the problem will be viable here, too. We use a preconditioner of the form

$$P = \begin{pmatrix} \frac{1}{\lambda^2}(\tilde{\boldsymbol{\mathfrak{L}}}^{\star}\tilde{\boldsymbol{\mathfrak{L}}})^{-1} & 0 \\ 0 & \boldsymbol{I}\beta \end{pmatrix} \tag{5.12}$$

with λ being the already known penalty factor and $\beta \in \mathbb{R}^+$ is an additional scaling parameter. The parameter β is necessary to account for the different scaling of the rows. In the reconstructions, we saw that the algorithm prioritized updates of the wave field component and did not update the material. Note that the operator P is positive definite for $\boldsymbol{u}(0) = \boldsymbol{u}(T) = \boldsymbol{0}$ because of the existence and uniqueness of

Algorithm 5.2 `CG-REGINN` method for all-at-once

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess; $\boldsymbol{s}_{\text{obs}} \in \boldsymbol{S}^{\Xi}$ % seismograms
Ensure: $\mu^k \in \boldsymbol{P}_h$ % approximate solution of (5.2)
$\mu^k \leftarrow \mu^0$
while not termination do
    $\mu_0^k \leftarrow \mu^k$
    $\xi \leftarrow 0$
    repeat
        $\xi \leftarrow \xi + 1$
        $\boldsymbol{u} \leftarrow \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \boldsymbol{J}_{h,\triangle t} \boldsymbol{f}_{\boldsymbol{\xi}}$ % other initial values are possible
        $b_{\boldsymbol{u}} \leftarrow \boldsymbol{J} \boldsymbol{f}_{\xi} + \Psi^* \boldsymbol{s}_{\text{obs},\xi}$
        $b_{\mu} \leftarrow \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu,\boldsymbol{u}}^{\prime *} [\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu} \boldsymbol{u} - \boldsymbol{J} \boldsymbol{f}^n]$
        $(\triangle \boldsymbol{u}, \triangle \mu_\xi^k) \leftarrow$ apply CG to the system in (5.11), with preconditioner from (5.12) and right-hand
            side $(b_{\boldsymbol{u}}, b_{\mu})$
        $\mu_{\xi+1}^k \leftarrow \mu_\xi^k + \triangle \mu_\xi^k$
    until $\xi > \Xi - 1$
    $\mu^{k+1} \leftarrow \mu_{\Xi-1}^k$
    $k \leftarrow k + 1$
end while
```

the underlying initial value problem

$$\Big(\boldsymbol{u}, (\tilde{\boldsymbol{\mathfrak{L}}}_{\mu_\xi^k}^{\star} \tilde{\boldsymbol{\mathfrak{L}}}_{\mu_\xi^k})^{-1} \boldsymbol{u}\Big)_{0,[0,T]\times\Omega} = \Big(\tilde{\boldsymbol{\mathfrak{L}}}_{\mu_\xi^k}^{-\star} \boldsymbol{u}, \tilde{\boldsymbol{\mathfrak{L}}}_{\mu_\xi^k}^{-\star} \boldsymbol{u}\Big)_{0,[0,T]\times\Omega} = \|\tilde{\boldsymbol{\mathfrak{L}}}_{\mu_\xi^k}^{-\star} \boldsymbol{u}\|_{0,[0,T]\times\Omega}^2 > 0\,. \tag{5.13}$$

This property makes it a suitable preconditioner for the `CG` algorithm. Although the equalities from (5.13) above do not hold in the discrete case as we mentioned in Remark 13, this did not cause problems in the algorithm. The preconditioner is similar to the forward backwards scheme in [AGO20], where it is used to solve the augmented wave equation (5.10). With the preconditioned algorithm we were able to reconstruct a simple 1D problem with this algorithm, but a careful hand-tuning of β was necessary. Therefore, we did not pursue this approach for 2D problems. It may be possible to develop a heuristic rule for the choice of β, but it is not clear how. Since we wanted to avoid to hand-tune β, we investigated another, more successful ansatz in the next section.

5.3.3 `AAO-PmSD-REGINN`

From Section 4.5.1 we know the `mSD-REGINN` algorithm and want to apply it to the all-at-once formulation giving rise to what we will call `AAO-PmSD-REGINN`. The goal is to alleviate the problem of the different scalings of the components and to avoid the need to hand-tune hyper parameters like in the `Red-CG-REGINN` above. In the notation of Section 4.5.1 we choose $U = \boldsymbol{Y}_h \times \boldsymbol{P}_h$, $V = \boldsymbol{Y}_h \times \boldsymbol{S}$. For the following however we swap the order of the wave field space and the parameters. We split the parameter space in M_1 parts and split the wave field space in M_2 subspaces. Hence, the linear system to calculate

Algorithm 5.3 `AAO-PmSD-REGINN`

```
Require: $\mu^0 \in \boldsymbol{P}_h$ % starting guess; $\boldsymbol{s}_{\text{obs}} \in \boldsymbol{S}^\Xi$ % seismograms
Ensure: $\mu^k \in \boldsymbol{P}_h$ % approximate solution of (RED)
 1: $\mu^k \leftarrow \mu^0$
 2: while not termination do
 3:     $\mu_0^k \leftarrow \mu^k$
 4:     $\xi \leftarrow 0$
 5:     repeat
 6:         $\boldsymbol{u}_\xi^k \leftarrow \tilde{\mathfrak{L}}^{-1}_{h,\mu_\xi^k} \boldsymbol{J}_{h,\triangle t} \boldsymbol{f}_\xi$ % one solution of the integrated forward wave equation
 7:         $\boldsymbol{r}_\xi^k \leftarrow (\boldsymbol{0}, \Psi \boldsymbol{u}_\xi^k - \boldsymbol{s}_{\text{obs},\xi})$ % wave residual is zero, since $\tilde{\mathfrak{L}}_{\mu_\xi^k} \boldsymbol{u} - \boldsymbol{J} \boldsymbol{f}_\xi = \boldsymbol{0}$
 8:         determine $l^k_{\max,\xi}$ and $\vartheta_\xi^k$ according to (4.7) and (4.6) to break the iterative process in line 9
 9:         $(\triangle\mu_\xi^k, -, l_\xi^k) \leftarrow$ PmSD$(\tilde{\Phi}'(\boldsymbol{u}_\xi^k, \mu_\xi^k), \boldsymbol{r}_\xi^k, \mu_\xi^k)$ % Algorithm 4.4 the update in the wave field is discarded

10:         $\mu_{\xi+1}^k \leftarrow \mu_\xi^k + \triangle\mu_\xi^k$
11:         $\xi \leftarrow \xi + 1$
12:     until $\xi > \Xi - 1$
13:     $\mu^{k+1} \leftarrow \mu_{\Xi-1}^k$
14:     $k \leftarrow k + 1$
15: end while
16: return $\mu^k$
```

the weighting factors $\alpha = (\alpha_1, \alpha_2)^\top$ like in (4.24) is of the form

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{12}^\top & B_{22} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \tag{5.14}$$

where $B_{11} \in \mathbb{R}^{M_1 \times M_1}, B_{12} \in \mathbb{R}^{M_1 \times M_2}, B_{22} \in \mathbb{R}^{M_2 \times M_2}$ and $c_1 \in \mathbb{R}^{M_1}, c_2 \in \mathbb{R}^{M_2}$. For our inversions we consider the parameter spaces on each cell separately: $U_{i+(j-1)\cdot N_{\text{cells}}} = \boldsymbol{e}_i^j, \forall i = 1, \ldots, N_{\text{cells}}$ and $j = 1, \ldots, N_{\text{inv}}$. With $\boldsymbol{e}_i^j \in \boldsymbol{P}_h$ we denote the ansatz function that is constant with value one in the i-th cell and the j-th parameter. The dimension of all those spaces is $M_1 = N_{\text{inv}} \cdot N_{\text{cells}}$. The values for all other cells and parameters are zero. For the wave field space we choose $U = \boldsymbol{Y}_h$ with $M_2 = 1$. By $(\boldsymbol{h}, \sum_{j=1}^{N_{\text{inv}}} \sum_{i=1}^{N_{\text{cells}}} d_i^j)$ we denote the decomposition of the descent direction that results from the choices above. The matrix B_{11} is composed of N_{inv}^2 blocks and reads

$$B_{11} = \begin{pmatrix} D_{11} & D_{12} & \cdots & D_{1N_{\text{inv}}} \\ D_{12} & D_{22} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ D_{1N_{\text{inv}}} & D_{2N_{\text{inv}}} & \cdots & D_{N_{\text{inv}}N_{\text{inv}}} \end{pmatrix}, \; D_{j_1 j_2} \in \mathbb{R}^{N_{\text{cells}} \times N_{\text{cells}}}, j_1, j_2 = 1, \ldots, N_{\text{inv}} \tag{5.15}$$

The block matrices are diagonal for our space discretization, since the Fréchet derivative only acts locally on the cells, so the scalar products are zero, if the cells are not the same. Assume we linearize at the point $(\boldsymbol{u}_\xi^k, \mu_\xi^k)$. Then, the entries of the block matrices in (5.15) are

$$\begin{aligned}(D_{j_1,j_2})_{i,i} &= \Big(\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\mathbf{0}, d_i^{j_1}], \tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\mathbf{0}, d_i^{j_2}]\Big)_{0,\boldsymbol{Y}_h\times\boldsymbol{S}} \\ &= \Big(\lambda\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^{j_1}], \lambda\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^{j_2}]\Big)_{0,[0,T]\times\Omega},\end{aligned}$$

for all $j_1, j_2 = 1, \dots, N_{\text{inv}}$ and $i = 1, \dots, N_{\text{cells}}$. Since $M_2 = 1$, B_{22} is a scalar:

$$B_{22} = \|\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\boldsymbol{h}, 0]\|_{0,\boldsymbol{Y}_h\times\boldsymbol{S}}^2 = \|\lambda\tilde{\mathfrak{L}}_{h,\mu}\boldsymbol{h}\|_{0,[0,T]\times\Omega}^2 + \|\Psi\boldsymbol{h}\|_{\boldsymbol{S}}^2.$$

With the choices above we further have $B_{12} \in \mathbb{R}^{(N_{\text{inv}}\cdot N_{\text{cells}})\times 1}$ with the entries

$$(B_{12})_{i,j} = \Big(\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\boldsymbol{h}, 0], \tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\mathbf{0}, d_i^j]\Big)_{0,\boldsymbol{Y}_h\times\boldsymbol{S}} = \Big(\lambda\tilde{\mathfrak{L}}_{h,\mu}\boldsymbol{h}, \lambda\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^j]\Big)_{0,[0,T]\times\Omega},$$

for $j = 1, \dots, N_{\text{inv}}$ and $i = 1, \dots, N_{\text{cells}}$. The Matrix B_{11} describes interaction of the parameters with each other, B_{12} the interaction between parameters and elements from wave field space and B_{22} the interaction of the wave field space, with itself. Let $(\triangle\boldsymbol{u}^l, \triangle\mu^l) \in \boldsymbol{Y}_h \times \boldsymbol{P}_h$ be the l-th iterate of the inner iteration and $(\boldsymbol{r}_{\boldsymbol{u}}^l, \boldsymbol{r}_{\boldsymbol{s}}^l) \in \boldsymbol{Y}_h \times \boldsymbol{S}$ the l-th residual of the linearized problem. Then, the upper part in the right-hand side vector for the linear system (5.14) is

$$\begin{aligned}(c_1)_{jN_{\text{cells}}+i} &= \Big(\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\triangle\boldsymbol{u}^l, \triangle\mu^l] - (\boldsymbol{r}_{\boldsymbol{u}}^l, \boldsymbol{r}_{\boldsymbol{s}}^l), \tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\mathbf{0}, d_i^j]\Big)_{0,\boldsymbol{Y}_h\times\boldsymbol{S}} \\ &= \Big(\lambda\tilde{\mathfrak{L}}_{h,\mu}\triangle\boldsymbol{u}^l + \lambda\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\triangle\mu^l] - \boldsymbol{r}_{\boldsymbol{u}}^l, \lambda\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^j]\Big)_{0,[0,T]\times\Omega}\end{aligned}$$

for $j = 1, \dots, N_{\text{inv}}$ and $i = 1, \dots, N_{\text{cells}}$. The lower part of the right-hand side is

$$\begin{aligned}c_2 &= \Big(\tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\triangle\boldsymbol{u}^l, \mu^l] - (\boldsymbol{r}_{\boldsymbol{u}}^l, \boldsymbol{r}_{\boldsymbol{s}}^l), \tilde{\Phi}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\boldsymbol{h}, 0]\Big)_{0,\boldsymbol{Y}_h\times\boldsymbol{S}} \\ &= \Big(\lambda\tilde{\mathfrak{L}}_{h,\mu}\triangle\boldsymbol{u}^l + \lambda\tilde{\mathfrak{L}}'_{h,\mu}[\triangle\mu^l] - \boldsymbol{r}_{\boldsymbol{u}}^l, \lambda\tilde{\mathfrak{L}}_{h,\mu}\boldsymbol{h}\Big)_{0,[0,T]\times\Omega} + \big(\Psi\triangle\boldsymbol{u}^l - \boldsymbol{r}_{\boldsymbol{s}}^l, \Psi\boldsymbol{h}\big)_{\boldsymbol{S}}.\end{aligned}$$

We solve the linear system (5.14) with the use of its Schur complement

$$S = B_{22} - B_{12}^\top B_{22}^{-1} B_{12}$$

which in our case is only a number. Then, we can compute α_2 by a simple division

$$\alpha_2 = S^{-1}(c_2 - B_{12}^\top B_{11}^{-1} c_1)\,. \tag{5.16}$$

Then, we also can calculate α_1:

$$B_{11}\alpha_1 = c_1 - B_{22}\alpha_2$$

which does not pose a problem since we already calculated B_{11}^{-1} for S. For $N_{\text{inv}} = 1$ the Matrix B_{11} is easy to invert since we just have to invert the diagonal elements. For $N_{\text{inv}} = 2$ the Schur complement for B_{22} is $S_{B_{11},ii} = (D_{22})_{ii} - (D_{12})_{ii}^2/(D_{11})_{ii}, i = 1, \dots, N_{\text{cells}}$. With $S_{B_{11},ii}$ we can express the inverse as

$$B_1^{-1} = \begin{pmatrix} D_{22}^{-1} + D_{22}^{-1} D_{12} S^{-1} D_{12} D_{22}^{-1} & -D_{11}^{-1} D_{12} S^{-1} \\ -S^{-1} D_{12} D_{11}^{-1} & S^{-1} \end{pmatrix}.$$

Since all these matrices are diagonal, all the inverses are easy to calculate. For higher values of N_{inv} we can use the same inversion formula with the Schur complement, but we have to do it in a recursive manner.

Remark 15. *In theory, the algorithm would also allow to split the wave field space. A canonic choice would be splitting the wave field for every time step. This, however, proved to be unstable in small numerical examples. Further, it poses a problem with regard to computational cost. With this choice the matrix B_3 is a dense symmetric matrix. For every entry one full space-time scalar product would have to be calculated. This means calculating about $\frac{1}{2}N^2$ space-time scalar products which is a considerable numerical effort that is comparable to solving the wave equation N times.*

We know from Section 5.3.2 that simply applying the algorithm without a preconditioning step is unlikely to lead to convergence of the method. Therefore, we use the preconditioner from (5.12) with $\beta = 1$:

$$P_\xi^k = \begin{pmatrix} \frac{1}{\lambda^2}(\tilde{\mathfrak{L}}^\star_{\mu_\xi^k}\tilde{\mathfrak{L}}_{\mu_\xi^k})^{-1} & 0 \\ 0 & \boldsymbol{I} \end{pmatrix} .$$

The positive definiteness from (5.13) guarantees that the negative gradient remains a descent direction after application of the preconditioner. The combination of algorithms can be found in Algorithm 5.3.

Remark 16. *Since we assume constant material values on each cell, we can also write the $d_i^j = \underline{d_i^j}\boldsymbol{e}_i^j$ and therefore get for the diagonal entries*

$$(D_{j_1,j_2})_{i,i} = \left(\lambda\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^{j_1}], \lambda\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[d_i^{j_2}]\right)_{0,[0,T]\times\Omega} \simeq \left(\tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\boldsymbol{e}_i^{j_1}], \tilde{\mathfrak{L}}'_{h,\mu_\xi^k,\boldsymbol{u}_\xi^k}[\boldsymbol{e}_i^{j_2}]\right)_{0,[0,T]\times\Omega} .$$

These values are similar to the values used in 'Pseudo-Hessian' preconditioner from (4.15), although they appear in a slightly different location. It is interesting that this type of values appear naturally in our inversion scheme which is not physically motivated.

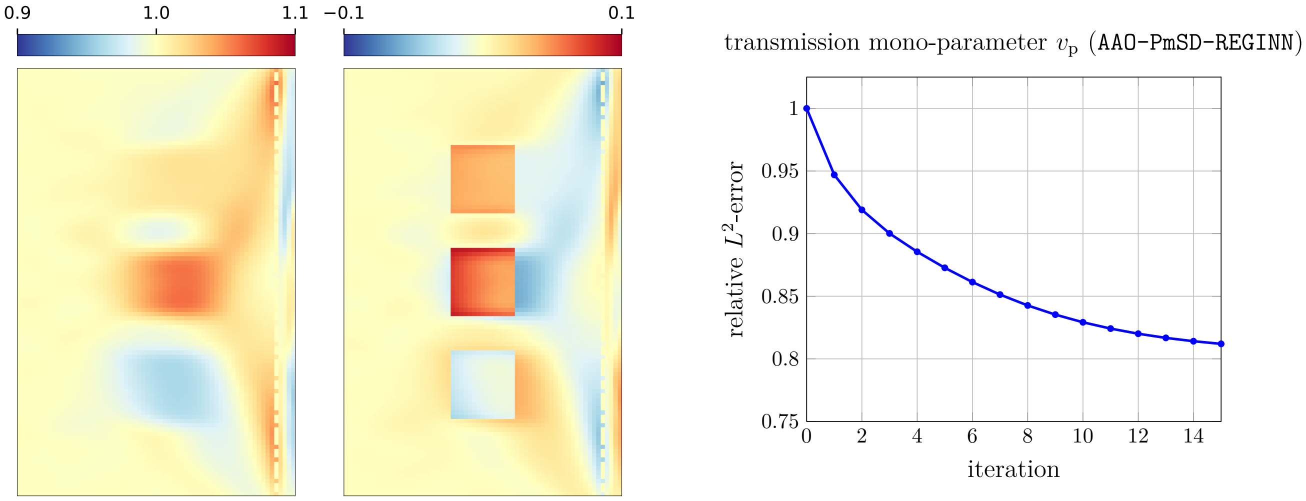


Figure 5.1: Left: final iterate $\mu_{v_\mathrm{p}}^{15}$ and difference to true values $\delta\mu_{v_\mathrm{p}}^{10}$ in the mono-parameter v_p inversion with `AAO-PmSD-REGINN`. Right: relative L^2-error as defined in (4.11). No inverse crime is committed in the reconstruction.

Numerical Cost

Before we present results, we want to clarify some aspects with regard to `AAO-PmSD-REGINN`. Although we call it an all-at-once algorithm, it is in a way an "all-at-once algorithm in the linearization". Only

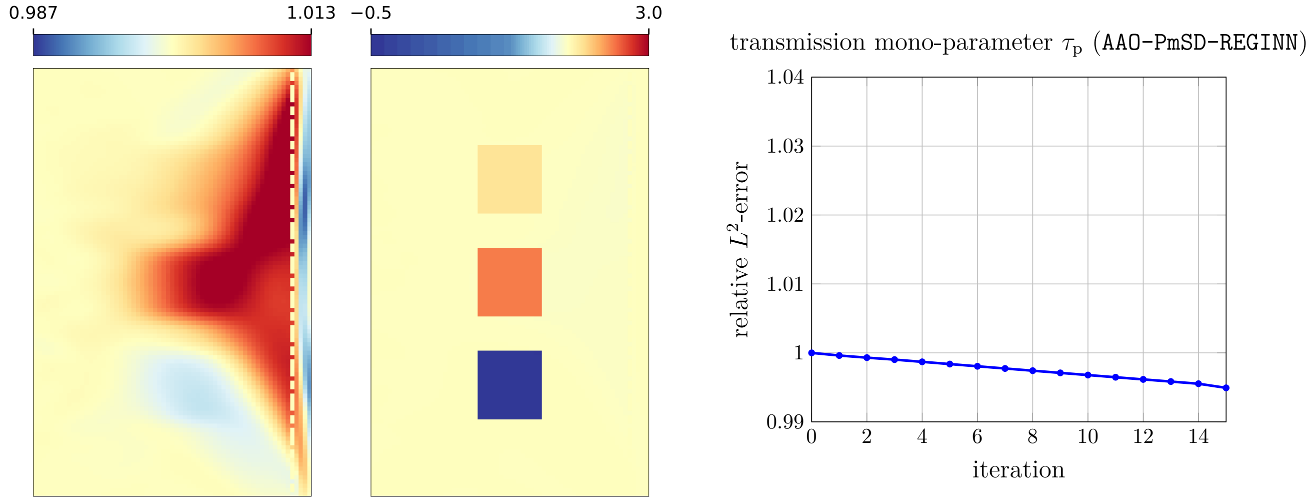


Figure 5.2: Mono-parameter τ_p inversion with `AAO-PmSD-REGINN`. Left: dissipation values (see (4.13)) final iterate $\mu^{10}_{1/Q}$, difference of the final iterate to the ground truth $\delta\mu^{\mathrm{true}}_{1/\mathrm{Q}}$. Right: relative L^2-error as defined in (4.11). No inverse crime is committed.

the material parameter part carries over from one shot to the next. In the k-th iteration and the ξ-th shot, we take the current iterate μ^k_ξ and consider the linearization in $(\mu^k_\xi, \tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu^k_\xi}\boldsymbol{J}_{h,\Delta t}\boldsymbol{f}_\xi)^\top$ (see line 6 of Algorithm 5.3,where $\boldsymbol{u}^k_\xi = \tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu^k_\xi}\boldsymbol{J}_{h,\Delta t}\boldsymbol{f}_\xi$ is calculated). There are several reasons for this: for each shot in our inversion we would have to save a full wave field, since they differ with regard to the source location. Hence we are not able to just pass one wave field for all shots as we do it for the material. Having one wave field for each shot would, however, increase the memory requirement too much. Our aim is, ultimately to avoid saving wave fields due to high memory consumption (we have to save degrees of freedom in space times number of time steps per wave field). Moreover, we tested saving the wave fields in a simple 1D example and found that they stagnate at some point. Then, we had to reset them to $\tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu^k_\xi}\boldsymbol{J}_{h,\Delta t}\boldsymbol{f}_\xi$ to achieve convergence. For these reasons we do not carry iterates of the wave fields in our algorithm. The inner loop of `PmSD-REGINN` therefore can be seen as an expansion away from the solution of the wave fields. We set the minimal number of inner iterations $l_{\min}$ for `AAO-PmSD-REGINN` to 3. This is a value we decided on by experience. Note that the value of $l_{\min}$ has to be at least 2, since the first inner iteration does not change the material. This can be seen with the following calculation. The residual in the first iteration of the inner loop is $(0, \Psi\boldsymbol{u}^k_\xi - \boldsymbol{s}_{\mathrm{obs},\xi})$. Set $\boldsymbol{s}_\triangle = \Psi\boldsymbol{u}^k_\xi - \boldsymbol{s}_{\mathrm{obs},\xi}$. Application of $\tilde{\Phi}'^*$ and the preconditioner yields for first update

$$d_{\mathrm{PC}} = \begin{pmatrix} (1/\lambda)^2 \tilde{\boldsymbol{\mathfrak{L}}}^{-1}_{h,\mu}\tilde{\boldsymbol{\mathfrak{L}}}^{-\star}_{h,\mu}\Psi^*\boldsymbol{s}_\triangle \\ 0 \end{pmatrix},$$

(see Algorithm 5.3). The main numerical cost of one inner iteration is solving two wave equations, one forward and one adjoint in the preconditioner. So one iteration of the inner loop of `AAO-PmSD-REGINN` needs the same number of solved wave equations as one iteration of the inner loop in `Red-CG-REGINN`. However, together with the calculation of $\boldsymbol{u}^k_\xi$, we have to solve a minimal number of $7 = 1 + l_{\min} \cdot 2$ wave equations per inner loop. If `Red-CG-REGINN` needs less than $l_{\min}$ to break, it is always cheaper than `AAO-PmSD-REGINN` per nonlinear iteration. Otherwise, both algorithms cost the same, if they need the same number of inner iterations. We apply the same stopping rule as we did in `Red-CG-REGINN`.

Transmission Reconstruction

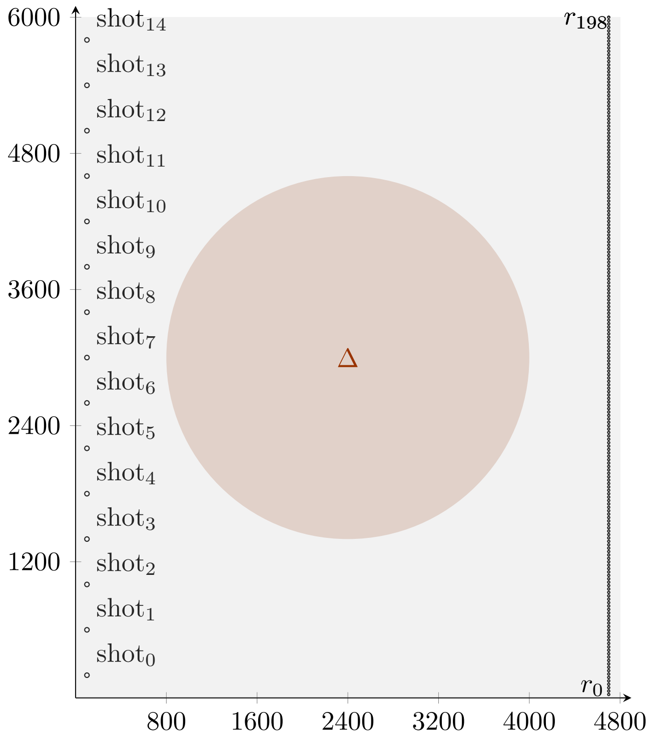
Again, we start by applying the method to the example of Fig. 4.1 for the mono-parameter problem for v_p (no inverse crime) for $L = 5$. In this experiment we set $\lambda = 1$. In Fig. 5.1 we can see that we can partly reconstruct the geometry, but we do not attain the same level of error as with the `Red-CG-REGINN` method. There are no source artifacts present as we saw in the reduced reconstruction in Fig. 4.4. This could be a result of the observation we made in Remark 16. These encouraging results prompted us to test `AAO-PmSD-REGINN` further. We applied the algorithm with $\lambda = 1$ to the τ_p mono-parameter experiment. The algorithm does not converge, the L^2-error increases from the first iteration. The result can be seen in Fig. 5.2. It has a lot of artifacts toward the receivers. With prior knowledge where the inclusions are one is also able to identify the middle and the bottom inclusion in the result, but they are not distinguishable against the artifacts without that knowledge. We tried the same inversion for various values of λ in the interval $[10^{-6}, 10^{20}]$, but it barely changed the outcome of the inversion. Also increasing l_min did not improve the results. Since the mono-parameter for τ_p failed, we do not apply multi-parameter reconstructions for `AAO-PmSD-REGINN` and focus on the v_p reconstructions from now on.

5.4 Camembert Model

In this section we present a situation, where the reduced method fails but the all-at-once method succeeds. This can be illustrated with a variant of the Camembert model, which was first introduced in [GVT86] as an example of a highly nonlinear problem. We consider a rectangle $\Omega = (0, 6000)\mathrm{m} \times (0, 4800)\mathrm{m}$ with homogeneous Neumann boundary. The background velocity is $v_\mathrm{p,bg} = 4000$ m/s. Within a circle centered around $x_\mathrm{mid} = (2400, 3000)^\top$ with radius $r = 1600\,\mathrm{m}$, the velocity differs from the background by a constant factor of $1 + p$. Therefore, the velocity is $v_{\Delta p} = (1 + p) \cdot v_\mathrm{p,bg}$ $p > -1$. We choose a mesh width $h = 37.5$, which yields 245760 degrees of freedom in space. We set $T = 2.1$ and $\triangle t = 0.0005$. Most of the calculations in this section are done on the HoreKa (Hochleistungsrechner Karlsruhe). One wave equation on 256 Intel Xeon Platinum 8368 cores with 2.4 GHz clock frequency takes 5 s to solve. A full inversion takes about 10 hours. On the left side of the domain are $\Xi = 15$ equally spaced sources $\boldsymbol{x}_{\mathrm{src},i} = (100, 200 + 400 \cdot i)^\top, i = 0, \dots, 14$. The source signal we use is the integrated Ricker wavelet $\tilde{r}$ from (4.14) with central frequency $f_c = 10\,\mathrm{Hz}$. On the right side of the domain, there are $R = 200$ equally spaced receivers $\boldsymbol{r}_i = (4700, 60 + 30 \cdot i)^\top, i = 0, \dots, 199$. Similar configurations of this problem are used in [EY20] and [GAO21]. More details can be taken from Fig. 5.3. According to [GVT86] the nonlinearity stems from the large diameter of the inclusion. This makes the initial model more important due to the local nature of inversion schemes such as Newton-type methods. In the following section, we will use a small example to illustrate the specific problem of the inversion of v_p.

Cycle-Skipping

In this section we visualize the phenomenon called *cycle-skipping*, which can occur in any seismic inversion for v_p and can be explained using the Camembert geometry. In this section we assume that there is only one source at $\boldsymbol{x}_{\mathrm{src},7} = (100, 3000)^\top$ but the rest is the same as above. In Fig. 5.4 the whole measurements



T	2.1
$\triangle t$	0.0005
L	0
k	1
Δh	37.5
dof_h	245760

Figure 5.3: Left: Geometry, sources, inclusion, and receiver for the Camembert model. Right: parameters of the numerical calculation.

for $p = \{0.00, 0.08, 0.15\}$ are shown. When we increase p the wave arrives earlier at the receivers the dotted line shows the approximate first arrival time for $p = 0.00$. Let $\boldsymbol{s}_{80}^{p}$ be the measurement of the 80-th receiver for the shot mentioned above for a simulation with value p for the Camembert. In Fig. 5.5 on the left we can see $\boldsymbol{s}_{80}^{p}$ for the values $p \in \{0.00, 0.08, 0.14, 0.15\}$ on the interval $[1.0, 1.7]$. We can further see that the change of v_{p} affects mostly the phase, and has little effect on the amplitude or the form of the wave. Now, we want to look at a simplified version of the inversion in the whole domain. Instead of optimizing over v_{p} we optimize over phase shifts for the measurements. In order to do this, define $\boldsymbol{m}^p(t) := \chi_{[1.0,1.7]}\boldsymbol{s}_{80}^{p}(t)$ and consider the one dimensional optimization problem

$$\min_{s\in[-0.15,0.3]} j_p(s) := \|\boldsymbol{m}^{0.00} - \boldsymbol{m}^p(\cdot - s)\|_{\boldsymbol{S}}^2.$$

This is a simplified model of how, in general, the residual is sensitive to the change of v_{p} for a constant initial value ($p = 0.00$). In Fig. 5.5 on the right j_p is depicted. We can see, that the function is non-convex for any p, and potentially has more than one local minimum. From the tangents in the three graphs at $s = 0.0$, we see that for $p = 0.08, 0.14$ the direction of steepest descent leads into the direction of the minimum of the function. Therefore, a local gradient method is likely to converge. For $p = 0.15$ the direction of steepest descent leads away from the minimum. For this case, if we only minimize the norm of the data residual in the L^2-norm, the algorithm cannot get around the local maximum and therefore cannot converge. This simplified example highlights the strong dependence on the initial value for inversions of v_{p} when only focusing on the data error. The term cycle-skipping is derived from the fact that one cycle is skipped, which can be seen in Fig. 5.5 for $p = 0.15$ (red): to reduce the data-error

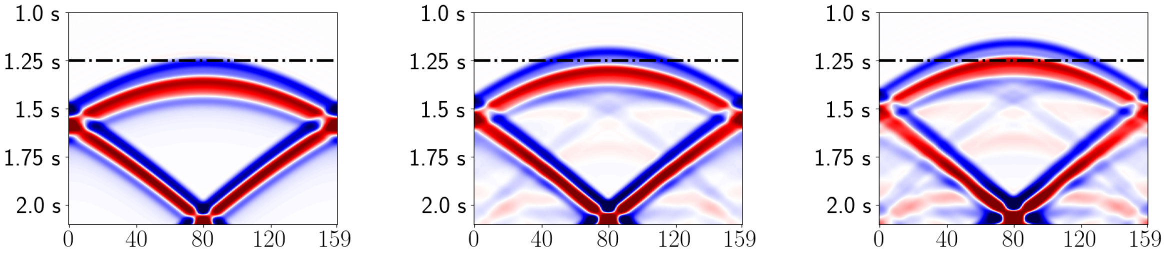


Figure 5.4: Observed data in Camembert geometry with $p = 0.00, 0.08, 0.15$ (from left to right). Each column in each plot corresponds to one receiver. Negative values are blue, positive values red, and zero is white

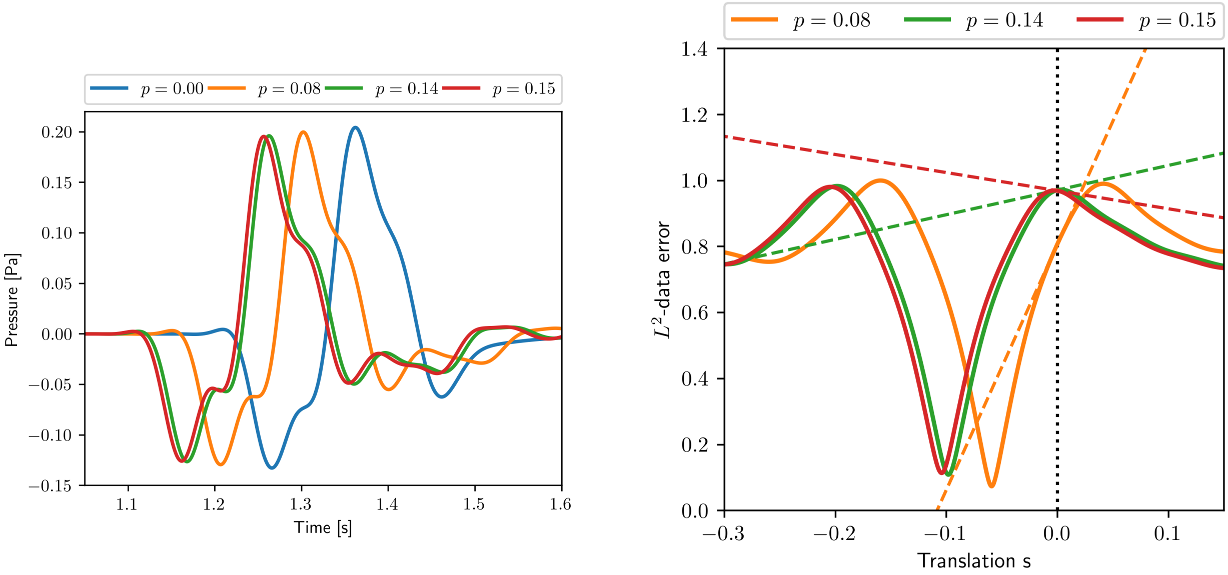


Figure 5.5: Left: measurements of the receiver at $\boldsymbol{r}_{80}$ for $p = 0.00, 0.08, 0.14, 0.15$ in the Camembert model with source at $\boldsymbol{x}_{\mathrm{src},7} = (100, 3000)^\top$. Right: plot of $j_p(s)$ for $p = 0.08, 0.14, 0.15$.

the initial (blue) seismogram moves to the right instead of the left, thereby skipping the full first cycle of the true seismogram. This phenomenon occurs commonly with real data sets as good initial values are not attainable most of the time. In this paragraph we showed why only focusing on the data error can be problematic and since the all-at-once formulation does additionally contain the model error (see the beginning of Section 4.1.1), all-at-once is likely to be more robust to this problem.

Remark 17. *Note that the frequency filtering from Section 4.4.2 is also a tool to reduce cycle-skipping since cycles with lower frequency have larger cycles, making it harder to skip them. We do not include frequency filtering in this section to have a more direct comparison of the methods. It is easy to implement it to* `AAO-PmSD-REGINN`.

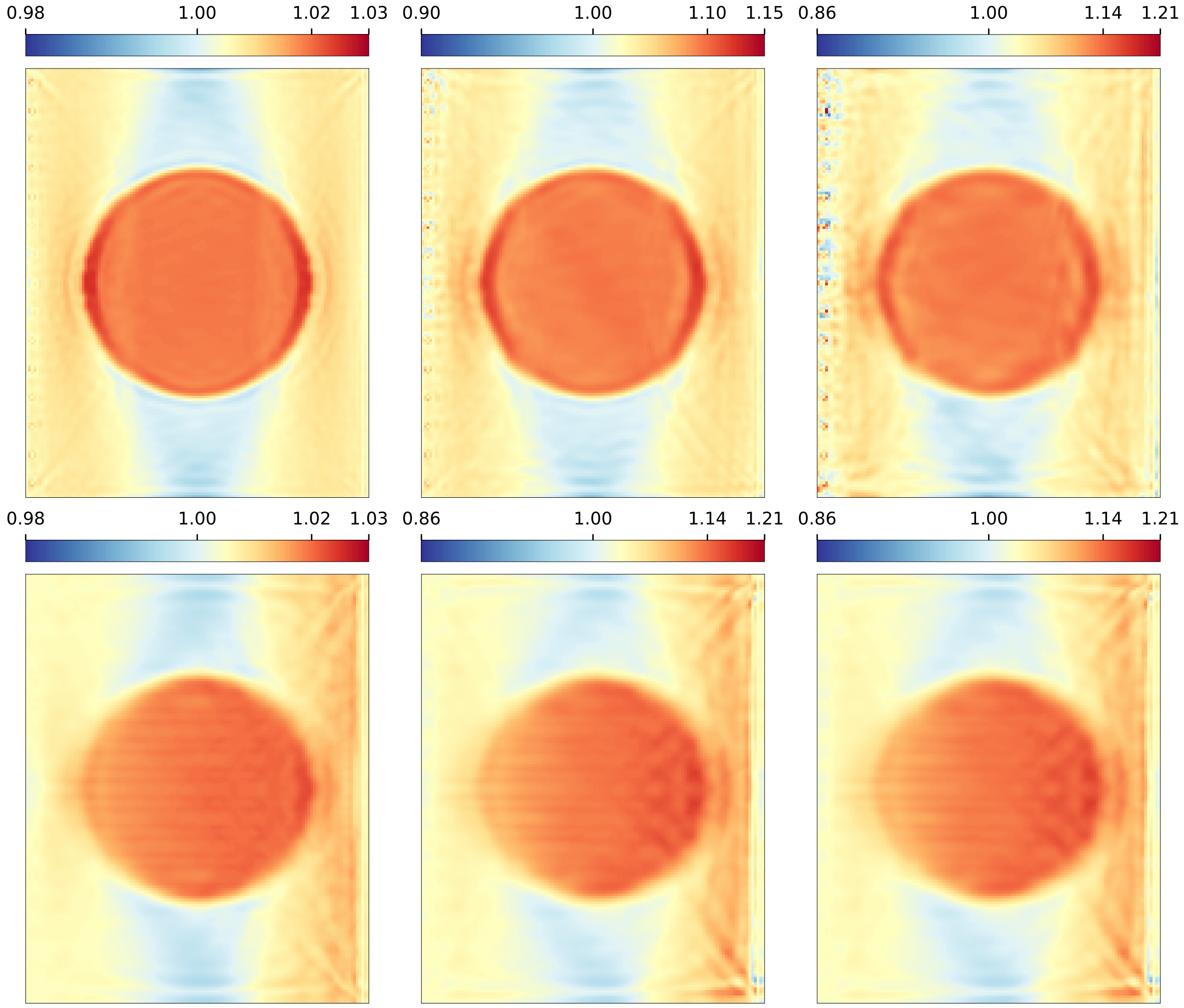


Figure 5.6: Results $\mu_{v_\mathrm{p}}^{15}$ for the Camembert model for $p \in \{0.02, 0.10, 0.14\}$ Top row: `Red-CG-REGINN`, bottom row: `AAO-PmSD-REGINN`. For both we can see the quality of the reconstruction deteriorates with increasing p.

Results with `Red-CG-REGINN`

Now, we turn to the inversion process across the entire domain. Our aim is to demonstrate that the difficulty of the inversion or the failure of `Red-CG-REGINN` is not determined solely by the size of the circle. To achieve this objective, we first start with lower p values, gradually increasing until the inversion fails. In total, we perform 15 iterations in each inversion. Looking at the final iterate in Fig. 5.6 and the errors in Fig. 5.8, we observe good results up to a percentage of $p = 0.14$. However, as p increases, reconstruction quality declines and the convergence rate slows down. Finally, for $p = 0.15$ `Red-CG-REGINN` fails. Some iterates of this failed inversion can be seen in Fig. 5.7. In the failed inversion, a large blue area is visible, indicating a decrease in velocity instead of an increase. This is the cycle-skipping effect: the algorithm uses a local descent in the wrong direction.

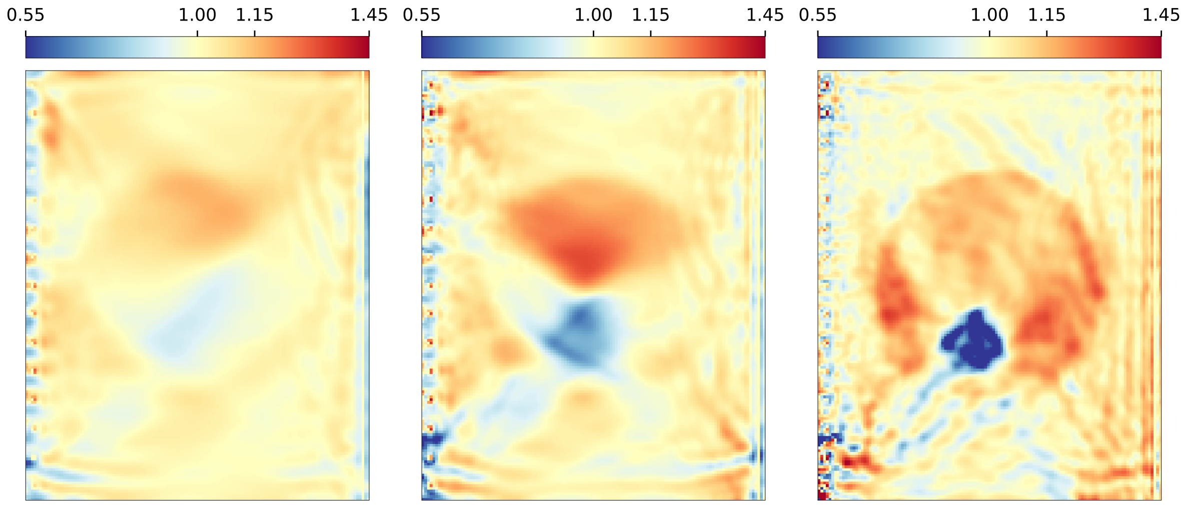


Figure 5.7: Iterates $\mu^k_{v_p}$ for $k \in \{1, 3, 15\}$ of `Red-CG-REGINN` with $p = 0.15$.

Results with `AAO-PmSD-REGINN`

The results of `AAO-PmSD-REGINN` for $p < 0.14$ can be seen in Fig. 5.6 and Fig. 5.8. They show that `AAO-PmSD-REGINN` is capable of inverting the structure of the Camembert model clearly. In all these reconstructions we set $l_{\min} = 3$. With regard to the relative error, `Red-CG-REGINN` is better (40% vs 50%). One major motivation of `AAO-PmSD-REGINN` is to find an algorithm that is more reliable in the sense that it has a larger radius of convergence. Therefore, we will now examine inversions that lie outside the convergence radius ($p < 0.14$) of `Red-CG-REGINN` as discussed in the preceding section. In Fig. 5.11 we can see that for $p = 0.15, l_{\min} = 3$ `AAO-PmSD-REGINN` is able to reduce the error considerably and therefore shows better convergence results than `Red-CG-REGINN`. Some iterates and the final result of this inversion can be seen in Fig. 5.9. We increase the percentage to $p = 0.16$ and are now comparing the computational efficiency of two different values of $l_{\min}$, namely 3 and 5. In order to have similar run times, we set the outer iterations of $l_{\min} = 3$ to 30, whereas for $l_{\min} = 5$ we keep it at 15. For both choices we see convergence in Fig. 5.11. Note that the curve for the $l_{\min} = 3$ has twice as many data points as the other curves. Even though we set increased the number of outer iterations, the algorithm with $l_{\min} = 3$ takes less time. Therefore, considering runtime only the number of forced inner iterations should be kept as low as possible. In the $p = 0.17$ case, the algorithm does not converge for $l_{\min} \in \{5, 7\}$ but it converges for $l_{\min} = 15$. This indicates that higher values of $l_{\min}$ can lead to convergence when the algorithm fails for lower values. Increasing $l_{\min}$ increases the numerical cost, so ideally one would choose it high enough for the algorithm to converge and as low as possible. In practice this value cannot be calculated exactly. Since convergence can critically depend on the value one should not be too eager to choose a value that is the lowest possible. For $p = 0.18$ and $p = 0.19$ we set values as high as 50 of $l_{\min}$ and no convergence is seen. This leads to the assumption that without further modification of the algorithm convergence cannot be attained for the Camembert model with `AAO-PmSD-REGINN`. In the failed reconstructions (see Fig. 5.10), we can see that `AAO-PmSD-REGINN` is not immune to cycle-skipping, as it also inserts negative values in the update. That the all-at-once ansatz can also suffer from cycle-skipping and being trapped in local minima is shown in [Sym20] and in [YZ23].

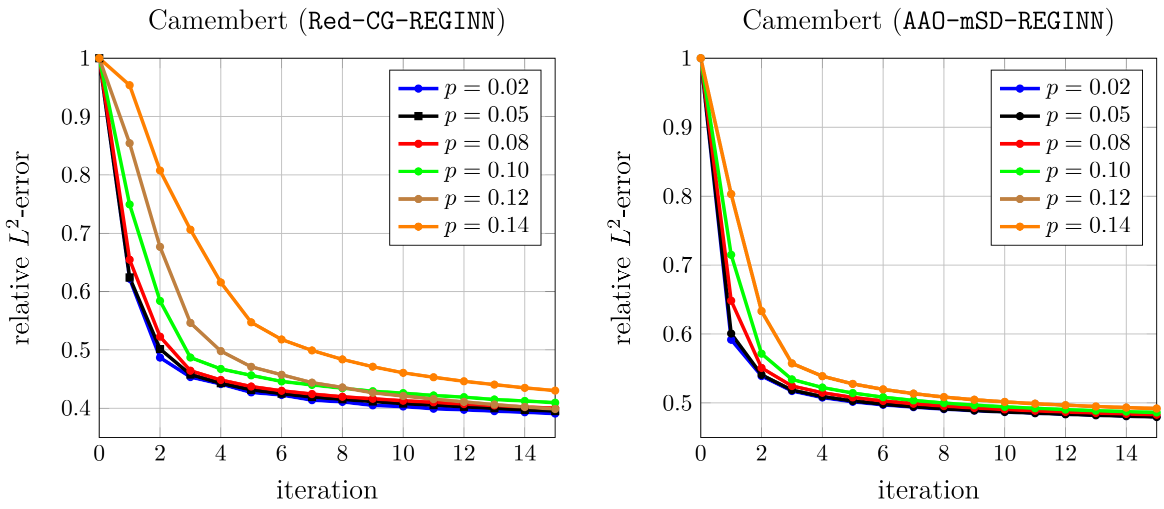


Figure 5.8: Relative L^2-errors as defined in (4.11) of the Camembert model. Left: `Red-CG-REGINN`, Right: `AAO-mSD-REGINN`. Some of the final iterates are shown in Fig. 5.6.

Combination of `Red-CG-REGINN` and `AAO-PmSD-REGINN`

In the sections before we saw convergence of `AAO-PmSD-REGINN` for this problem for higher values of p whereas `Red-CG-REGINN` tended to converge to lower values of the L^2 error, if it converged. Therefore, we want to combine the two methods to benefit from "best of both worlds". To this end we start `Red-CG-REGINN` with iterates of `AAO-PmSD-REGINN` for $p = 0.16$. If these starting values are close enough to the real value `Red-CG-REGINN` should converge in a case where we see divergence for the constant starting value. Four different iterates of `AAO-PmSD-REGINN` are taken as starting values; $\mu^k_{v_\mathrm{p},\mathrm{A}}, k = 1, 3, 5, 15$. In Fig. 5.11 we can see that the convergence of `Red-CG-REGINN` is dependent on the initial value. For $k = 1, 3$ the reconstruction of `AAO-PmSD-REGINN` is still not close enough to the true value to see convergence in `Red-CG-REGINN`. However, for $k = 5, 15$ the algorithm converges and `CG-REGINN` seems to be able to further reduce the error even after `AAO-PmSD-REGINN` stagnates. In Fig. 5.12 we can see that the boundary of the circle is clearer and `Red-CG-REGINN` is able to reduce some of the artifacts while at the same time introducing some of its own at the source positions. This also worked for $p = 0.17$ and $k = 10$, as also is shown in Fig. 5.12. This demonstrates that we can profit from a combination of the algorithms.

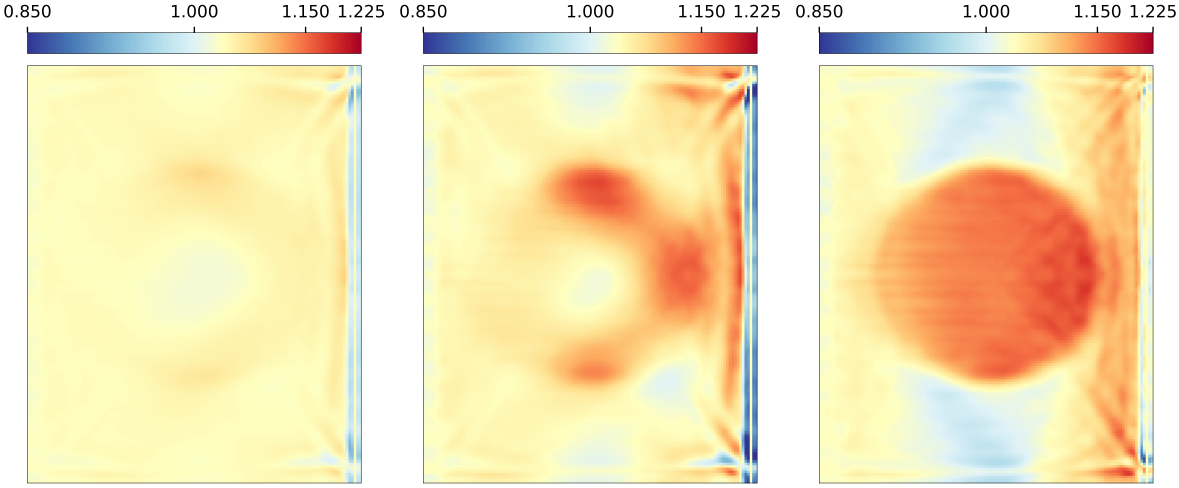


Figure 5.9: Iterates $\mu_{v_{\mathrm{p}}}^{k}$ for $k \in \{1, 3, 14\}$ of `AAO-PmSD-REGINN` for the Camembert model for $p = 0.15$.

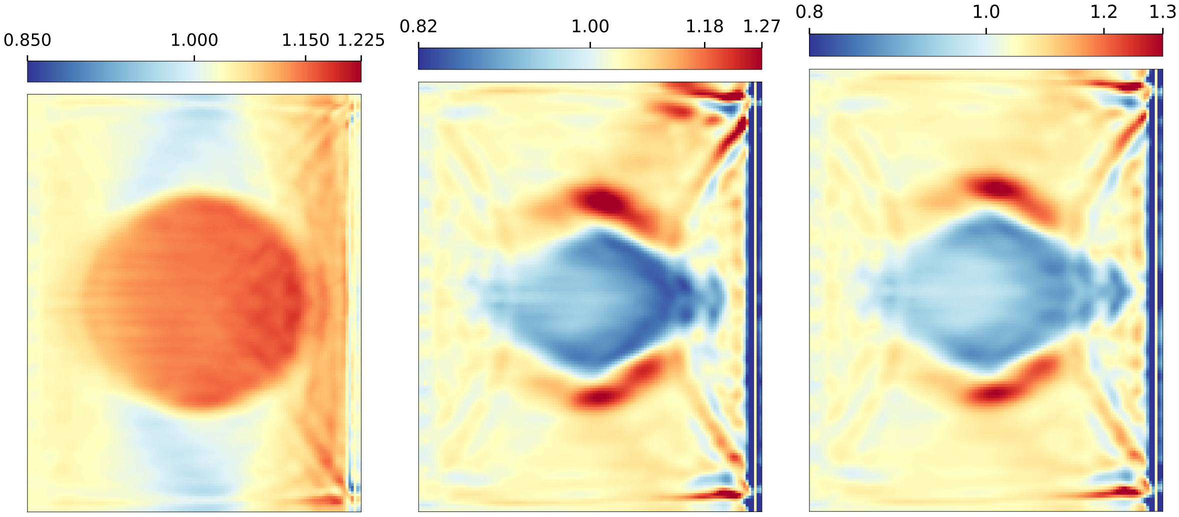


Figure 5.10: Final iterate $\mu_{v_{\mathrm{p}}}^{15}$ of `AAO-PmSD-REGINN` for the Camembert model for $p \in \{0.15, 0.18, 0.20\}$(from left to right), the two right ones are failed inversions.

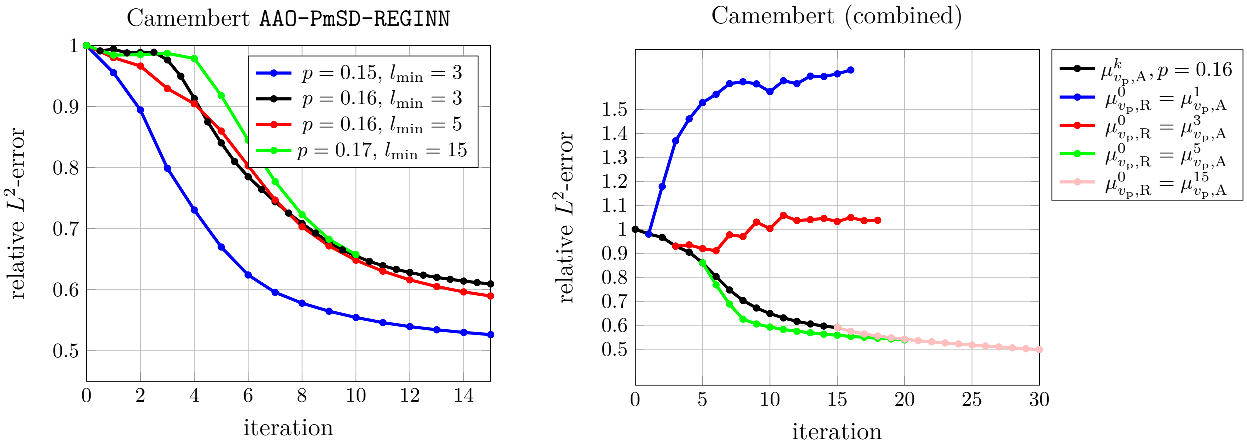


Figure 5.11: Left: relative L^2-errors as defined in (4.11) of the Camembert model with AAO-PmSD-REGINN for $p \in \{0.15, 0.16, 0.17\}$. Red-CG-REGINN fails for these values. Notice, that the black line has 30 iterations, instead of 15 as the others and which is indicated at the bottom. Right: relative L^2-errors of Red-CG-REGINN with initial value $\mu^0_{v_p,\mathrm{R}}$. As initial values, we choose iterates of AAO-PmSD-REGINN $\mu^k_{v_p,\mathrm{A}}$ for $k \in \{1, 3, 5, 15\}$, only for $k = 5, 15$ the inversion succeeds to decrease the error.

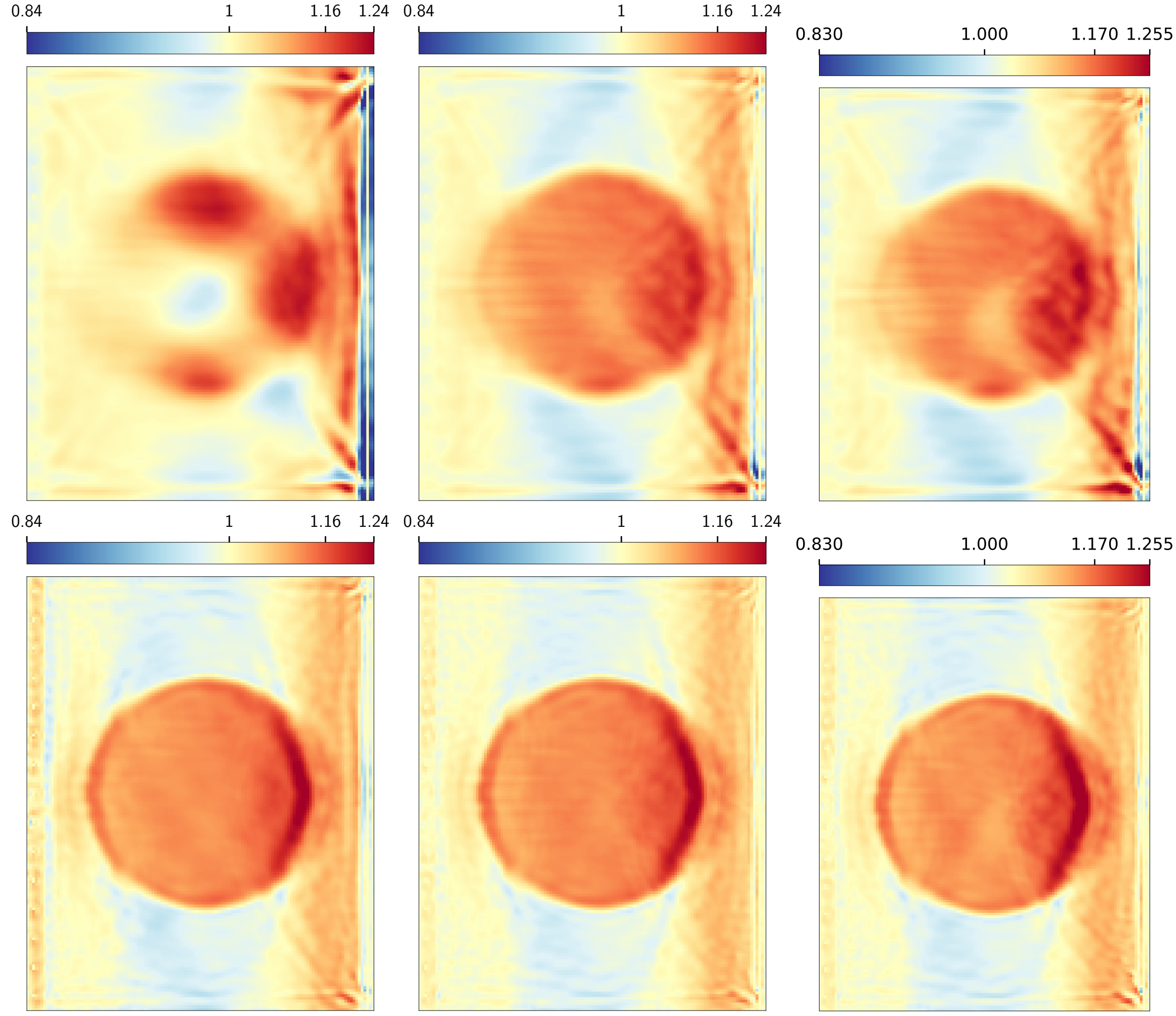


Figure 5.12: Top: from left to right: initial values $\mu^0_{v_\mathrm{p},\mathrm{R}} = \mu^k_{v_\mathrm{p},\mathrm{A}}$ for $k \in \{5, 15\}$ and $p = 0.16$. The picture on the right uses and $k = 10$ for $p = 0.17$. Both are `Red-CG-REGINN` continuation experiments. Bottom: results $\mu^{15}_{v_\mathrm{p},\mathrm{R}}$ of the inversions.

5.5 Marmousi

5.5.1 Application of `AAO-PmSD-REGINN`

We apply `AAO-PmSD-REGINN` to the Marmousi setup from Section 4.4.2. We want to further demonstrate that the larger convergence radius also carries over to other, more complex, and layered inversions. To illustrate this, three initial values that lie beyond the convergence radius of `Red-CG-REGINN`: the 'blurred' model from Fig. 4.13, the 'averaged' model (which is the blurred model averaged over each row), and the 'linear' model (which assumes a linear increase of the initial model depending on the depth).The values start at $1500\,{}^{\mathrm{m}}/_{\mathrm{s}}$ at the seabed and increase to $4000\,{}^{\mathrm{m}}/_{\mathrm{s}}$ at the bottom of the model. The latter initial values can be seen in Fig. 5.13. We set the minimal number of inner iterations to $l_{\mathrm{min}} = 3$ and apply no frequency filtering and do not enforce lower and upper bounds on v_{p}. The iteration numbers are fixed to 20 (blurred,which is stopped earlier because it attains good results earlier) and 25 (averaged, linear) outer iterations. In order to distinguish between the reconstructions from different initial values, we refer to them as $\mu_{v_{\mathrm{p}},\mathrm{blur}}, \mu_{v_{\mathrm{p}},\mathrm{avg}}$ and $\mu_{v_{\mathrm{p}},\mathrm{lin}}$. The results in Fig. 5.15 demonstrate that `AAO-PmSD-REGINN` is able to produce reasonable results, where `Red-CG-REGINN` fails. The quality of the reconstruction deteriorates as the distance from the original value increases, with the blurred initial value being closest and the linear one farthest away. The reconstruction of the shallow parts is reconstructed reasonably well, but the deeper parts do not reflect the layered structure well. This is especially true for the *averaged* and *linear* initial value. The L^2-errors in Fig. 5.16 show that the error decreases for all algorithms. They are normalized to the initial error of the linear reconstruction. This is done to show a quantitative comparison: even the last iteration of the linear reconstruction has a higher error than the initial value of the averaged reconstruction and the last averaged reconstruction still has a higher error than the initial *blurred* reconstruction. This shows that the L^2-error can be misleading, because the final reconstructions arguably carry more information about the subsurface than the initial values of the other algorithms. We should note that adjusting the frequency filter or modifying the parameter l_{min} may improve reconstruction results, but it is not the focus of this study.

Remark 18. *There also was one case where we did not achieve convergence. We tried to push the algorithm to its limit and started with a constant initial value with* $2750\,{}^{\mathrm{m}}/_{\mathrm{s}}$. *It did not produce meaningful results and produced singularities in the model. This reconstruction can be seen in Fig. 5.14. For this initial value we actually did some tinkering. We applied frequency filtering of Order 2 and 4 to the frequency* $4.5\,\mathrm{Hz}$ *and additionally increased the number of forced minimal iterations* l_{min} *to 7. None of these measures substantially improved the results, however.*

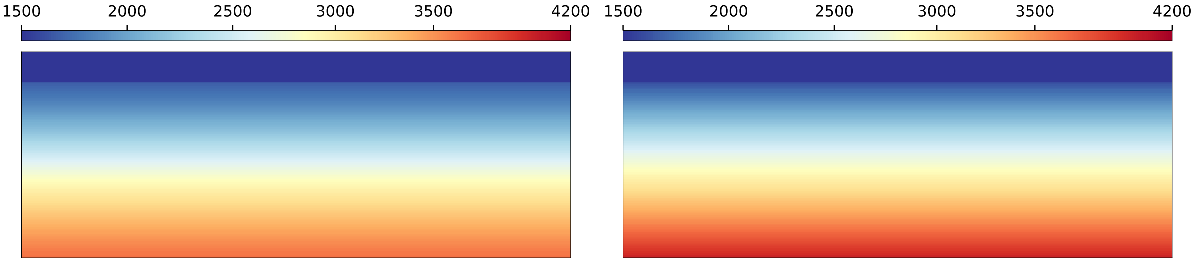


Figure 5.13: Initial values of the `PmSD-REGINN` inversion. Left: $\mu_{v_{\mathrm{p}},\mathrm{avg}}$, Right: $\mu_{v_{\mathrm{p}},\mathrm{lin}}$.

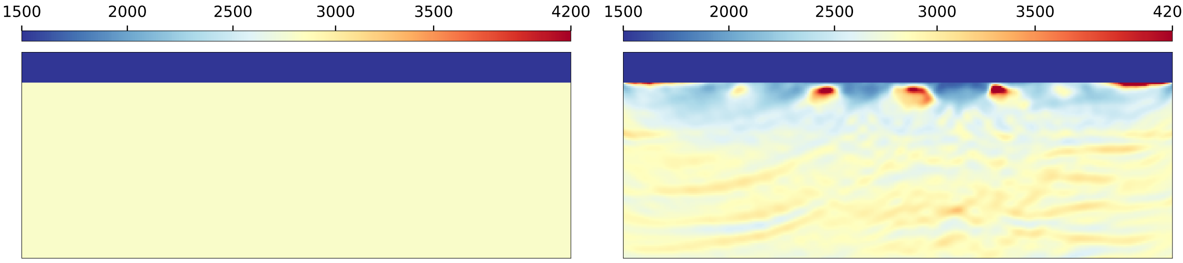


Figure 5.14: Left: constant initial value, Right: result after 15 iterations.

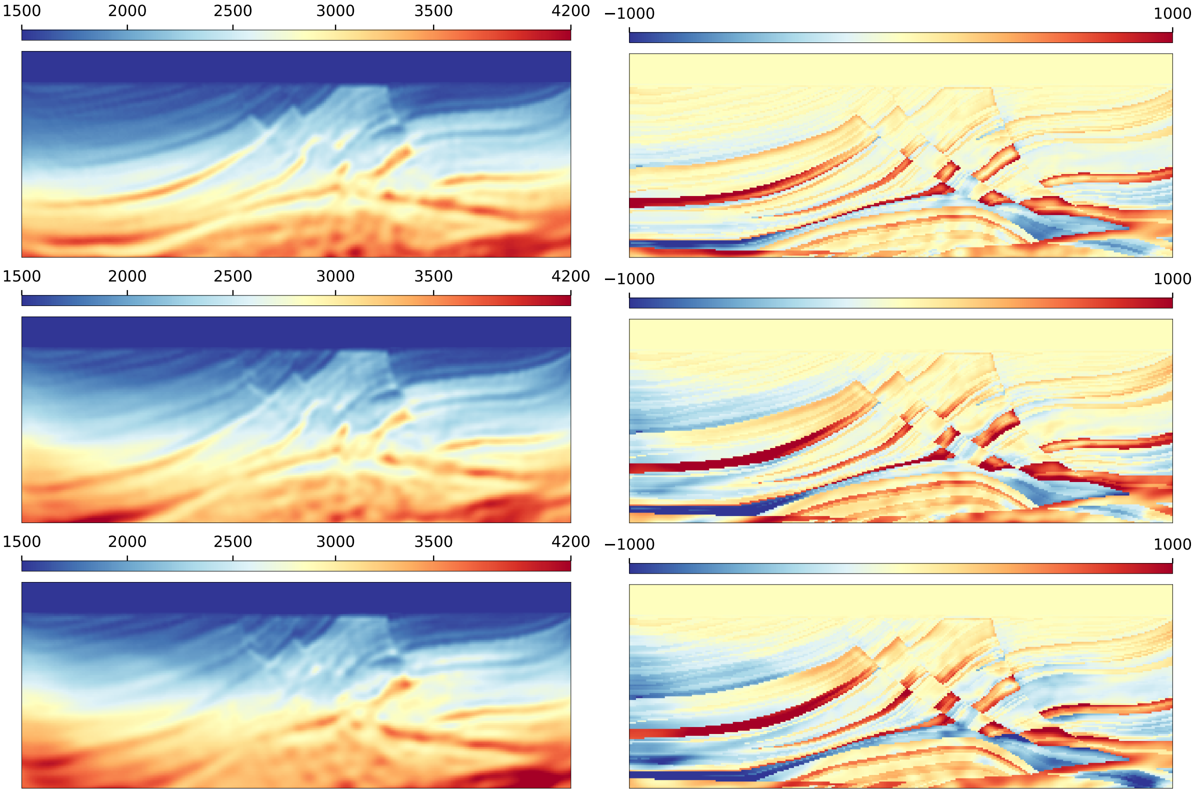
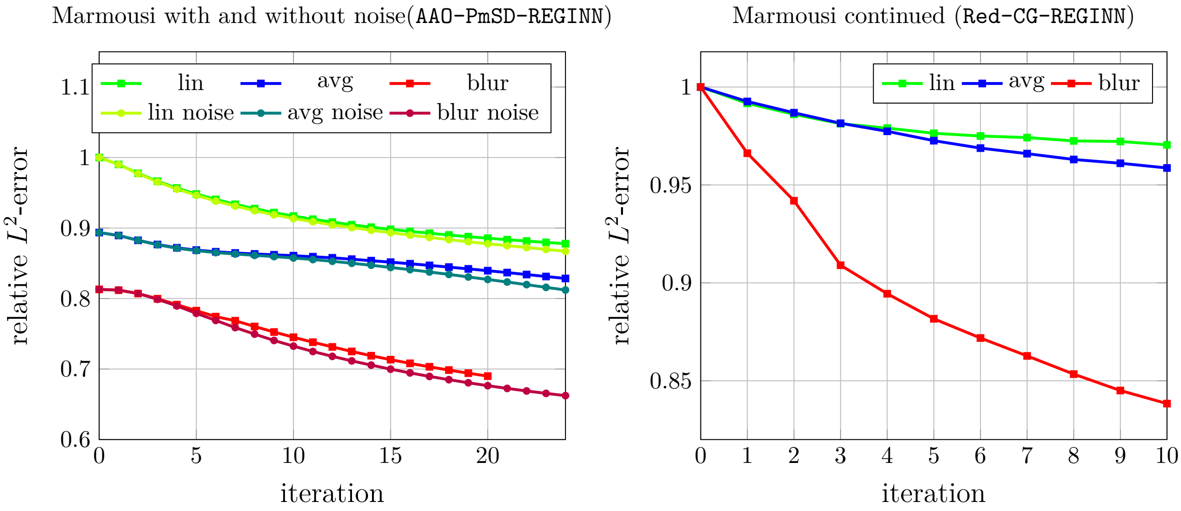


Figure 5.15: Left: final reconstructions $\mu^{20}_{v_p,\mathrm{blur}}, \mu^{25}_{v_p,\mathrm{avg}}, \mu^{25}_{v_p,\mathrm{lin}}$ (top to bottom) of `AAO-PmSD-REGINN` and differences to the true model $\delta\mu^{20}_{v_p,\mathrm{blur}}, \delta\mu^{25}_{v_p,\mathrm{avg}}, \delta\mu^{25}_{v_p,\mathrm{lin}}$

5.5.2 Continuation

While it was not our goal to attain the best error in the reconstruction, we still want to show that combining the `AAO-PmSD-REGINN` method with `Red-CG-REGINN` is able to produce results that are better than just applying `AAO-PmSD-REGINN`. In this reconstruction we just added one reconstruction with `Red-CG-REGINN` without frequency filtering or the Pseudo-Hessian preconditioner. The final iterates are shown in Fig. 5.17, and the relative error reduction with respect to the initial value produced by `AAO-PmSD-REGINN` is shown in Fig. 5.16. For the linear and the averaged initial, the additional error reduction is moderate (3% and 4%), while for the blurred starting model there is a more significant reduction of 15%. Note that the final error of the blurred inversion `AAO-PmSD-REGINN` is comparable (28.9% relative reduction with respect to the initial value) to the error at the end of the three stages of `Red-CG-REGINN` from (30.5% relative


Marmousi with and without noise(`AAO-PmSD-REGINN`)

Marmousi continued (`Red-CG-REGINN`)

Figure 5.16: Left: errors for the inversions from Fig. 5.15, Fig. 5.18, and Fig. 5.17 relative to the initial error of the all-at-once Marmousi inversion with the linear initial value. Right: relative improvement of the final iterate for the continuation with `Red-CG-REGINN` of $\mu^{20}_{v_\mathrm{p},\mathrm{blur}}, \mu^{25}_{v_\mathrm{p},\mathrm{avg}}, \mu^{25}_{v_\mathrm{p},\mathrm{lin}}$.

reduction with respect initial value, see Fig. 4.17).

5.5.3 Marmousi with noise

Now we run the Marmousi experiment `AAO-PmSD-REGINN` with strong noise, that is, SNR $\in \{0.1, 1\}$, is SNR the signal-to-noise ratio from (4.17). The error of this inversion is shown in Fig. 5.16. For SNR $= 1$, the final error even drops below the error of the data without noise. This reconstruction can be taken as evidence that `AAO-PmSD-REGINN` is stable even for noisy data. The algorithm does not converge for SNR $= 0.1$. In Fig. 5.18 we can see that although the reconstruction is heavily riddled with noise artifacts, we can still see the structure of the upper part of the Marmousi model.

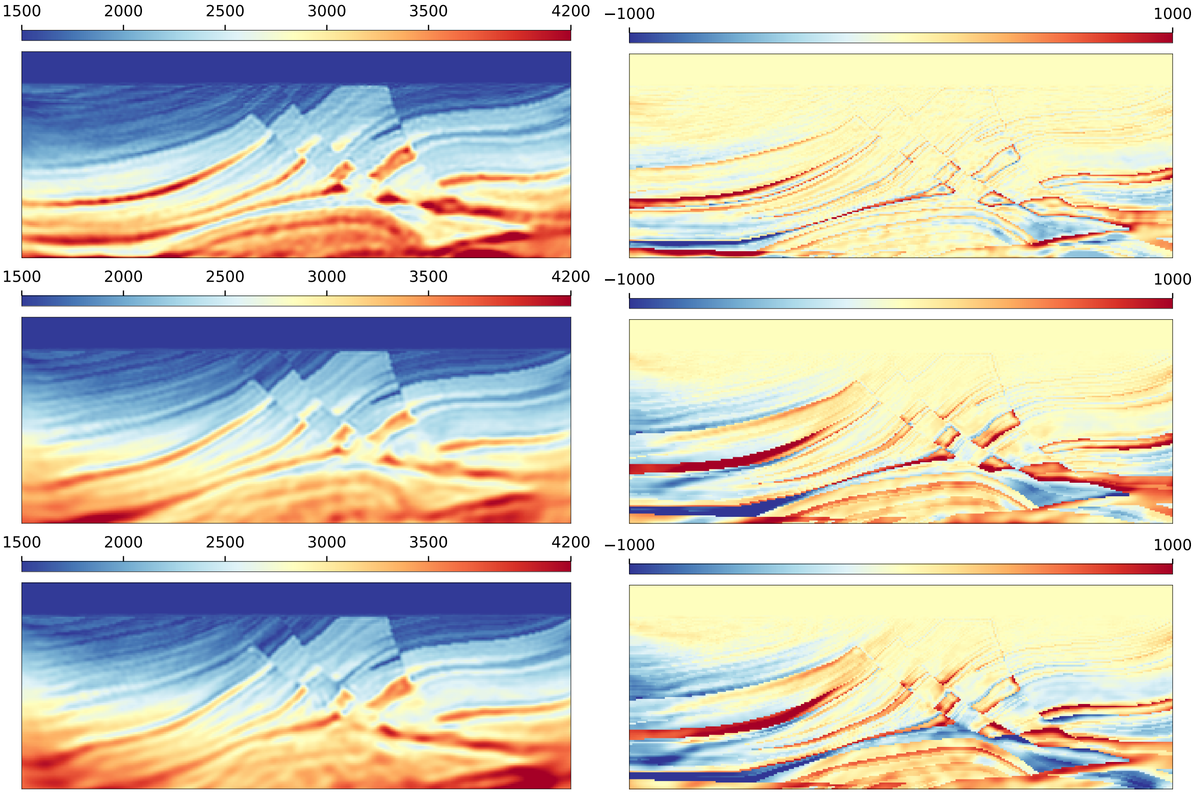


Figure 5.17: Final reconstruction $\mu^{10}_{v_\mathrm{p},\mathrm{blur}}, \mu^{10}_{v_\mathrm{p},\mathrm{avg}}, \mu^{10}_{v_\mathrm{p},\mathrm{lin}}$ of the combination of `AAO-PmSD-REGINN` and `Red-CG-REGINN` and difference to the true model $\delta\mu^{10}_{v_\mathrm{p},\mathrm{blur}}, \delta\mu^{10}_{v_\mathrm{p},\mathrm{avg}}, \delta\mu^{10}_{v_\mathrm{p},\mathrm{lin}}$ for the blurred, averaged and linear initial value from top to bottom.

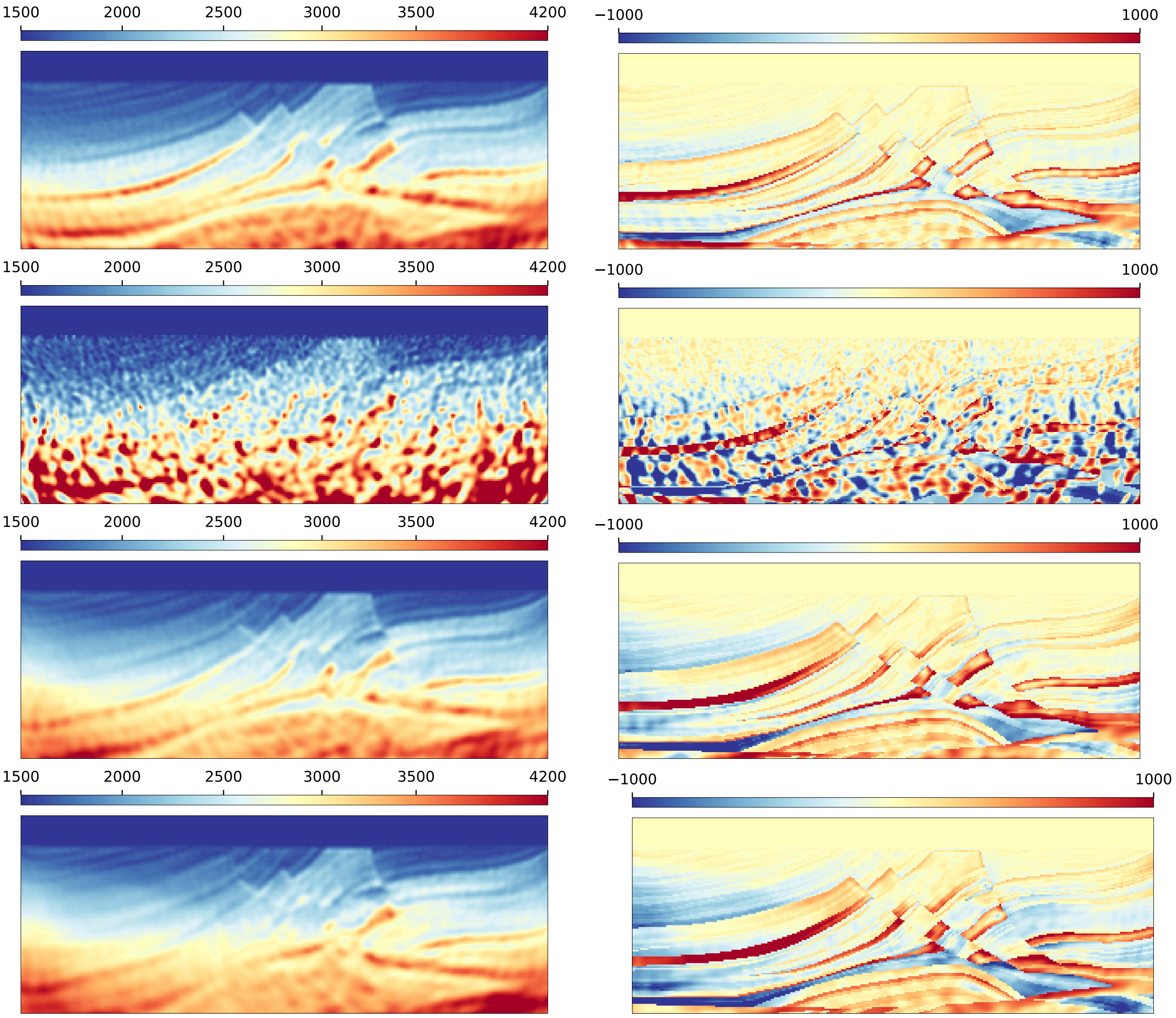


Figure 5.18: Reconstructions of Marmousi with noisy data with `AAO-PmSD-REGINN`. Left: final result of the inversion, Right: difference to the true model. The initial value and SNR from top to bottom are: blurred with SNR = 1, blurred with SNR = 0.1, averaged with SNR = 1, linear with SNR = 1.

5.5.4 On the choice of the penalty factor λ in (AAO)

So far, not much consideration was put into the choice of λ. We simply set it to 1 in most inversions and saw good results with this choice. Now, we want to expand on this aspect and see if we can influence the result or the speed of convergence of the inversions by changing λ. Before we start with numerical results of the study, we calculate the first update of the inner loop (see Algorithm 4.4) of `AAO-PmSD-REGINN`. Let $\boldsymbol{s}_{\mathrm{obs}}$ be the observed data and define $\boldsymbol{s}_{\triangle} := \Psi \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \boldsymbol{J}_{h,\triangle t} \boldsymbol{f} - \boldsymbol{s}_{\mathrm{obs}}$. Then, the starting inner residual is $r^0 = (0, \boldsymbol{s}_{\triangle})^{\top}$. Applying $\tilde{\Phi}'_{\mu}$ and the preconditioner we get $d^0_{\mathrm{PC}} = ((1/\lambda)^2 \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}, 0)^{\top}$. Since there is no update in the material, we can ignore all blocks in the linear system (5.14) that are related to the material. The whole block matrix therefore reduces to the scalar $B_{22} = \|\tilde{\Phi}'_{\mu}[d^0_{\mathrm{PC}}]\|^2_{0,\boldsymbol{Y}_h \times \boldsymbol{S}}$. We only have to consider the formula from (5.16) to calculate the factor for the wave update directly:

$$\alpha_2^0 = \frac{\left(\tilde{\Phi}' \left[d^0_{\mathrm{PC}}\right], \begin{pmatrix} 0 \\ \boldsymbol{s}_{\triangle} \end{pmatrix}\right)_{0,\boldsymbol{Y}_h \times \boldsymbol{S}}}{\|\tilde{\Phi}'_{\mu}[d^0_{\mathrm{PC}}]\|^2_{0,\boldsymbol{Y}_h \times \boldsymbol{S}}}$$

which yields

$$\begin{aligned} \alpha_2^0 &= \frac{\left(\Psi \lambda^{-2} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}, \boldsymbol{s}_{\triangle}\right)_{\boldsymbol{S}}}{\|\lambda^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{0,[0,T]\times\Omega} + \|\lambda^{-2} \Psi \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{\boldsymbol{S}}} \\ &= \left(1 + \lambda^{-2} \frac{\|\Psi \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{\boldsymbol{S}}}{\|\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{0,[0,T]\times\Omega}}\right)^{-1} . \end{aligned}$$

Therefore, the first iterate of the inner loop is

$$x^1 = \alpha_2^0/\lambda^2 (\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}, 0) = \tilde{\alpha}_2^0 (\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}, 0) \text{ with } \tilde{\alpha}_0^0 := \left(\lambda^2 + \frac{\|\Psi \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{\boldsymbol{S}}}{\|\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|^2_{0,[0,T]\times\Omega}}\right)^{-1} .$$

From this representation we can see that

$$\tilde{\lambda} := \frac{\|\Psi \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-1} \tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|_{\boldsymbol{S}}}{\|\tilde{\boldsymbol{\mathfrak{L}}}_{h,\mu}^{-\star} \Psi^* \boldsymbol{s}_{\triangle}\|_{0,[0,T]\times\Omega}} \tag{5.17}$$

is a pivotal point in this update. While for $\lambda \ll \tilde{\lambda}$ the factor $\tilde{\alpha}_2^0$ stays approximately constant, around $\lambda \approx \tilde{\lambda}$ the behavior changes into a transitional phase, and for $\lambda \gg \tilde{\lambda}$ it decreases with the speed of λ^{-2}. We conduct two studies: we start `AAO-PmSD-REGINN` for the failed τ_{p} mono-parameter example from from Fig. 5.2 and for the Marmousi geometry, together with the blurred initial value with a wide range of values for λ, and let it run for one iteration. We denote the result of the inner loop (which runs for 3 and 7 iterations for the τ_{p} experiment and the Marmousi experiment, respectively) by $(\Delta \boldsymbol{h}_{\lambda}, \Delta \mu_{\lambda})^{\top}$ and the final residual by $(\boldsymbol{h}_{\mathrm{res},\lambda}, \boldsymbol{s}_{\mathrm{res},\lambda})^{\top}$. The subscript λ emphasizes the dependence of these quantities in terms of the penalty factor. We want to look at the evolution of $\|\Delta \boldsymbol{h}_{\lambda}\|_{0,[0,T]\times\Omega}$, $\|\Delta \mu_{\lambda}\|_{0,\Omega}$, $\|\boldsymbol{h}_{\mathrm{res},\lambda}\|_{0,[0,T]\times\Omega}$, and $\|\boldsymbol{s}_{\mathrm{res},\lambda}\|_{\boldsymbol{S}}$ over λ. We do this for the first shot each, but the results apply – for these two examples – to all shots. In Fig. 5.19 and Fig. 5.20 we can see the plot of these quantities over λ. The dotted line shows the discrete value of $\tilde{\lambda} = 5.311 \cdot 10^8$, $\tilde{\lambda} = 1.318 \cdot 10^5$ for the Marmousi and the τ_{p} experiment, respectively. The results in Fig. 5.19 illustrate that when λ is less than $\tilde{\lambda}$, the norms $\|\Delta \boldsymbol{h}_{\lambda}\|_{0,[0,T]\times\Omega}$, $\|\Delta \mu_{\lambda}\|_{0,\Omega}$ remain constant. As λ approaches $\tilde{\lambda}$ from below, the norms show a quadratic decrease. This is evident as the norms parallel the dashed line. Not only the norm of the updates stays more or less the same, the updates themselves are very similar, too. For the Marmousi experiment in the constant section of the curve in Fig. 5.19 (left) we calculate

$$\max_{\lambda_i,\lambda_j\in\{10^{-8},10^{-7},\ldots,10^{4}\}} \left|1-\frac{\|\Delta\mu_{\lambda_i}-\Delta\mu_{\lambda_j}\|_{0,\Omega}}{\|\Delta\mu_{\lambda_i}\|_{0,\Omega}}\right| = 3.16\cdot 10^{-3}.$$

The maximal relative difference between the updates in the flat part is less than $3.16\cdot 10^{-3}$. This shows that changing λ does not change the updates in a significant way. The behavior of $\|\boldsymbol{h}_{\mathrm{res},\lambda}\|_{0,[0,T]\times\Omega}$, $\|\boldsymbol{s}_{\mathrm{res},\lambda}\|_{\boldsymbol{S}}$ over λ is shown in Fig. 5.20 in the right column. When the value of λ is less than $\tilde{\lambda}$, the residual norm $\boldsymbol{s}_{\mathrm{res},\lambda}$ decreases in relation to the initial residual $\boldsymbol{s}_{\Delta}$. From $\lambda\approx\tilde{\lambda}$ on, $\|\boldsymbol{s}_{\mathrm{res},\lambda}\|_{\boldsymbol{S}}\approx\|\boldsymbol{s}_{\Delta}\|_{\boldsymbol{S}}$. The wave field residual norm seemingly does not influence the updates and peaks at $\tilde{\lambda}$. This residual does not seem to give any information about the success of the inversion, while the reduction of the data residual seems to carry crucial information. This shows, as long as $\lambda\ll\tilde{\lambda}$, the algorithm is not very sensitive to λ. Note that the behavior does not change exactly at $\tilde{\lambda}$ but can also start earlier as we approach this point (we called this 'transitional phase' above). Moreover, numerical errors can change the tipping point in these numerical calculations. Since we primarily want to avoid very small updates, we want to stay in the 'flat' part in the plots. Therefore, we suggest choosing $\lambda = s\tilde{\lambda}$ with $s<10^{-3}$. Note that calculating $\tilde{\lambda}$ does not add any numerical cost. We can calculate $\|\tilde{\mathfrak{L}}_{h,\mu}^{-\star}\Psi^*\boldsymbol{s}_{\Delta}\|_{0,[0,T]\times\Omega}$ and $\|\Psi\tilde{\mathfrak{L}}_{h,\mu}^{-1}\tilde{\mathfrak{L}}_{h,\mu}^{-\star}\Psi^*\boldsymbol{s}_{\Delta}\|_{\boldsymbol{S}}$ while applying the preconditioner and save it for later for the calculation of α_2^0. In our experiments the value of $\tilde{\lambda}$ almost did not change over the course of the inversion (for $\lambda=1$). Also, it was the same order of magnitude for all shots. Nevertheless, we suggest calculating $\tilde{\lambda}$ and adjusting λ accordingly for each shot individually. This is especially true for cases where we may have different source signals or apply source inversion in real measurements since this could lead to different orders of magnitude in $\tilde{\lambda}$. To conclude this consideration, we could say that in the interval where we want λ to be in, the algorithm is not very sensitive to it. This result can be seen as an advantage since we do not have to calibrate the factor and do not have to think of it all that much after choosing it once/calculating it on the fly. Therefore, we have one parameter less to tune. However, we cannot control how much influence is put on the wave equation part in the inversion which was one of the motivations for the optimization ansatz (5.4). Moreover, the algorithm does not revert to a reduced algorithm as we increase λ.

Remark 19. *The result from our calculations mirrors an observation in [vLH15] for the parameter choice. They identify the operator norm of $\Psi\tilde{\mathfrak{L}}_{h,\mu}^{-1}$ as an interesting value and say anything smaller or larger than the operator norm can be considered a small respectively large value of λ. Although their algorithm is not connected to ours, they draw this conclusion from an analysis of an expansion of the wave field in their inner loop for different values of λ. A small calculation shows the link from our suggestion to their observation:*

$$\|\Psi\tilde{\mathfrak{L}}_{h,\mu}^{-1}\|_{\boldsymbol{Y}_h\longrightarrow\boldsymbol{S}} = \sup_{\|\boldsymbol{f}\|_{0,[0,T]\times\Omega}\neq 0}\frac{\|\Psi\tilde{\mathfrak{L}}_{h,\mu}^{-1}\boldsymbol{f}\|_{\boldsymbol{S}}}{\|\boldsymbol{f}\|_{0,[0,T]\times\Omega}} \geq \frac{\|\Psi\tilde{\mathfrak{L}}_{h,\mu}^{-1}\tilde{\mathfrak{L}}_{h,\mu}^{-\star}\Psi^*\boldsymbol{s}_{\Delta}\|_{\boldsymbol{S}}}{\|\tilde{\mathfrak{L}}_{h,\mu}^{-\star}\Psi^*\boldsymbol{s}_{\Delta}\|_{0,[0,T]\times\Omega}}.$$

So, in their words, our algorithm only works for small values of λ.

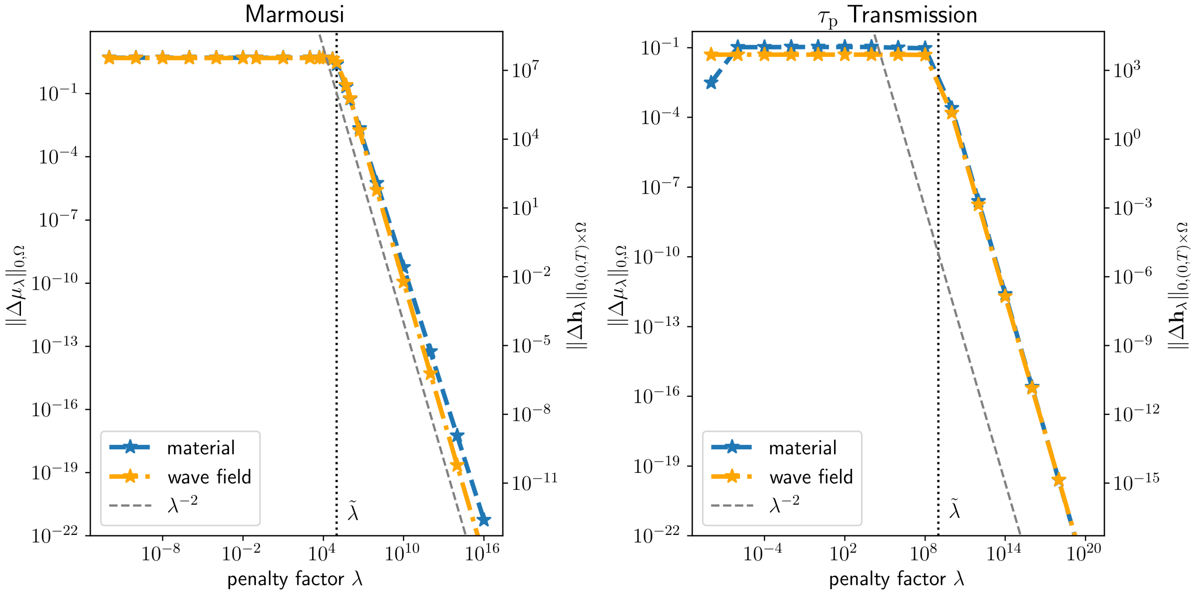


Figure 5.19: Evolution of the norms of the components $\Delta\boldsymbol{h}_\lambda, \Delta\mu_\lambda$ of the update (result of the inner loop) in the first nonlinear iteration over λ. The dotted line marks the value of $\tilde{\lambda}$ from (5.17). Left: values of the v_p inversion for the Marmousi geometry. Right: values of the τ_p inversion for the Transmission geometry.

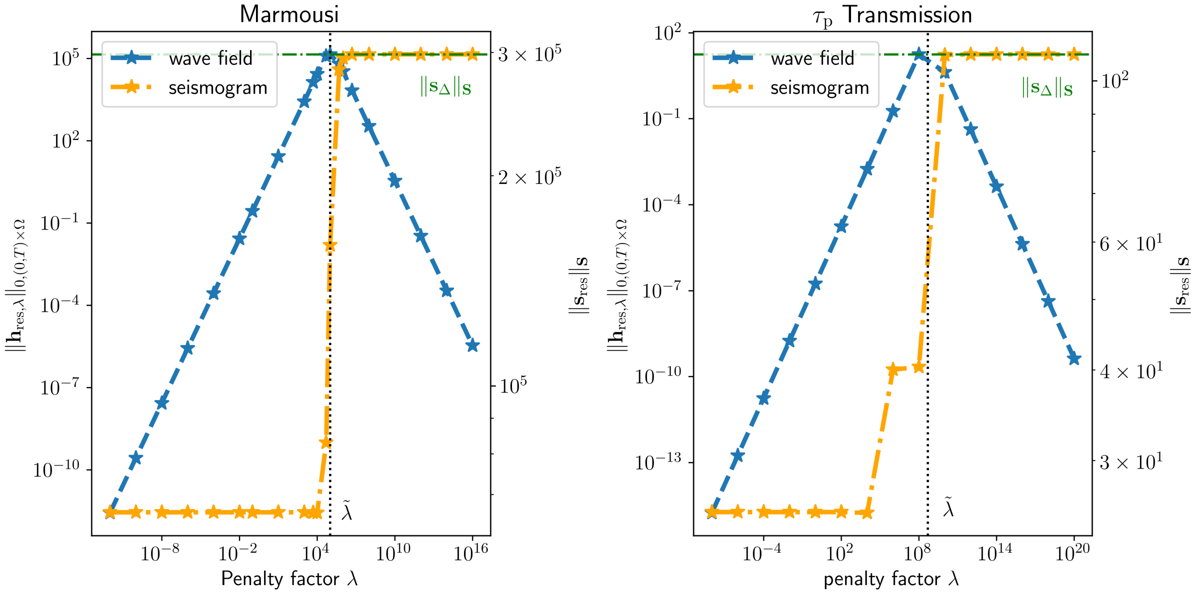


Figure 5.20: Evolution of residual component norms $\boldsymbol{h}_{\mathrm{res},\lambda}, \boldsymbol{s}_{\mathrm{res},\lambda}$ in the inner loop. The dotted line indicates the numerical value of $\tilde{\lambda}$ from (5.17), while the green line signifies the residual in the seismogram at the start of the inner loop.

Left: values of the v_p inversion for the Marmousi geometry. Right: values of the τ_p inversion for the Transmission geometry.

CHAPTER SIX

CONCLUSION

Even almost 40 years after its inception, Full Waveform Inversion remains an actively researched topic. Applications in global scale models allow for new insights into the structure of the earth and local scale models allow for a more efficient way to find and exploit natural resources. Due to its ill-posedness, a careful approach to FWI is necessary. A large amount of data and high numerical cost for each iteration make regularized methods necessary that converge in a fast and stable manner. With increasing calculation power, new possibilities arise to exploit the method to its full potential. One way to exploit the potential is putting research into new formulations of the underlying problem of FWI that were not computationally viable before but have merits with regard to the inversion. In this thesis we did both. We applied an inexact Newton method that proved to be flexible and is known to be able to stably solve ill-posed problems (for examples in Electrical Impedance Tomography, see [WR15, MRL14]). Moreover, we developed and applied a new type of `REGINN` algorithm in the all-at-once formulation in time domain.

6.1 Contribution

We started with the reduced formulation and applied the algorithm we called `Red-CG-REGINN`. The combination of tools (formulation of the wave equation, discretization, and inversion algorithm) was used before in [BFE$^+$21] for a transmission example only. After adding some techniques known from the literature (frequency filtering, Pseudo-Hessian preconditioner) we expanded on the usefulness of this combination by successfully applying `Red-CG-REGINN` to the Marmousi geometry. In this geometry we showed stability of the v_p inversion with `Red-CG-REGINN` in a viscous medium. Even for noisy data with a SNR up to 1. After that we turned to the problem of multi-parameter inversion, which means that we want to invert for multiple parameters at once. In Section 4.5 we presented and applied a novel and flexible ansatz (we called `mSD`) in the `REGINN` context that allows to balance different parts of gradient updates in a way that is well-suited for the multi-parameter inversion. In the reduced context the algorithm was called `Red-mSD-REGINN`, and we applied it to a multi-parameter problem in a transmission geometry. We saw monotone error decay for $\tau_\mathrm{p}, v_\mathrm{p}$, where `Red-CG-REGINN` was unable to reconstruct values in τ_p. This result demonstrated the potential of `Red-mSD-REGINN` to improve multi-parameter inversions in the reduced formulation.

In Chapter 5 we took the theoretical results for the all-at-once formulation from [Rie21] and presented discretizations of the operators therein. We implemented and investigated an algorithm from the literature in time domain that was previously considered in frequency domain. Moreover, we developed and presented a combination of the previously presented `mSD` ansatz and `REGINN` called `AAO-PmSD-REGINN` with a preconditioner that is adapted to the all-at-once formulation. For the same geometries we used for `Red-CG-REGINN`, we conducted and presented numerical experiments for `AAO-PmSD-REGINN` that show viability for v_{p} inversions and robustness under addition of artificial noise. Additionally, we showed that `AAO-PmSD-REGINN` can be used to avoid cycle-skipping to some extent: In examples where the reduced ansatz failed, we saw convergence of `AAO-PmSD-REGINN`. This highlights that the algorithm is less dependent on the choice of the initial value. In the same geometries we used for `Red-CG-REGINN`, we conducted and presented numerical experiments for `AAO-PmSD-REGINN` that confirmed the viability for v_{p} inversions and robustness under addition of artificial noise. We demonstrated that a combination of `AAO-PmSD-REGINN` and `Red-CG-REGINN` can be advantageous, that is, starting `Red-CG-REGINN` with the result of `AAO-PmSD-REGINN`. The algorithm `AAO-PmSD-REGINN` is more robust with regard to the initial value, and `Red-CG-REGINN` converges to better results as soon as it converges. With this combination we get the benefits of both strengths of the algorithms. Further, we suggested an easy-to-realize choice of the penalty parameter λ due to a brief analysis which was backed up by numerical experiments. Finally, we also want to include a minor point: We developed a brief analysis on the origin of artifacts around sources and receivers that are inherent to reduced methods. We also provided complimentary numerical calculations to show the behavior of material updates in a very simple geometry.

6.2 Outlook

The final goal of most seismic inversion techniques is to be feasible and performant to real-world applications. That means inverting real measured data in a three-dimensional setting. After some additional features (such as source inversion) are implemented, the algorithms `Red-CG-REGINN`, `Red-mSD-REGINN`, and `AAO-PmSD-REGINN` can be applied to real measured data. However, for `AAO-PmSD-REGINN` it poses an additional challenge to do three-dimensional inversions. The all-at-once algorithm we presented needs to save at least four wave fields simultaneously. One each for the iterate, the update, the residual, and one temporary while applying the preconditioner. This is a lot and can only be executed on very large modern calculation clusters: Typical applications have at least $\approx 10^9$ grid points (see [Ige16, vLH15]). For a simulation with $L = 3$ we have 3 velocity components and 4 pressure components. Therefore, one time step in double precision consumes $\approx 1\,\mathrm{GB}$ RAM, making a full wave field easily consume more than 1 TB. Hence, before the algorithm can be applied to three-dimensional problems, it should be altered in a way that is more memory-friendly. A standard technique in reduced methods is saving snapshots and recalculating parts of the wave field when needed. In `AAO-PmSD-REGINN` this is not possible, since the wave fields are not just solutions of the wave equation. Also, we cannot store the right-hand sides that produce the wave field. They are results of previous calculations and not known as in reduced applications. Furthermore, other preconditioners could be explored that do not rely on saving an additional wave field. One possible way to save memory could be hp-adaptive space-time methods (cf. [DWZ20] for a p-adaptive variant or [CDW22] for a variant with an error indicator). The space-time ansatz offers more flexibility to adapt both h and p in space and time. In a certain sense, space-time methods are a perfect candidate for all-at-once methods: Solving (5.10) comes much more natural in a space-time setting, since we are forced to consider the whole system anyway and cannot just use time-stepping. A small study concerning space-time methods in the context of reduced FWI was done in [Ern18] and could be extended to all-at-once methods. This space-time approach could also open up a new class of preconditioners such as wave adapted space-time multi-grid methods.

Note that a natural extension of all algorithms in this thesis would be including elastic or visco-elastic modeling (including shear waves), which would further increase the memory needs of `AAO-PmSD-REGINN`. An optimal choice of the minimal number of inner iterations l_{min} also remains an open question. In the Camembert example, we saw that the convergence of `AAO-PmSD-REGINN` can depend on l_{min}. Other non-systematic testing indicates that l_{min} should be chosen as small as possible: The speed of the inversions did not improve with regard to calculation time when we increased it over the threshold of convergence. However, we did not present a heuristic to choose a good value for l_{min}. This should be researched further. Also, the reconstruction of τ_{p} with `AAO-PmSD-REGINN` warrants further research. It is an open question whether `AAO-PmSD-REGINN` is capable of reconstructing attenuation. Although our first experiments indicate that in its current version it is not, there may be some modification that improves the sensitivity to attenuation. Finding a way around this restriction could lead to a viable algorithm for multi-parameter problems, because the structure of the B_{11} block (see (5.15)) provides a very natural way to implement multiple-parameters into the linear system (5.14). Perhaps it may even reduce cross-talk due to its correlating nature.

In the context of the reduced FWI formulation other `REGINN` variants could be explored. Variants that include additional variational regularization or other measures of distance than the L^2-norm. One such variant is a L^∞ approach presented in [PR23], which was not yet implemented for two-dimensional problems. Methods that include evaluations of the second Fréchet-derivative are often considered in geophysics and are a promising way to improve reconstructions, especially for multi-parameter inversions. Therefore, a combination of `REGINN` and methods from the inverse problems community that include a second derivative (such as [HR99]) could be explored.

For `Red-mSD-REGINN` more complex geometries for multi-parameter inversions could be tested. Additionally, the flexibility of the decomposition of the update could be further explored. For `Red-mSD-REGINN` we could not only split the different material parameters but could also separate the updates spatially. For example with prior information about a region of interest, or by splitting the update depending on the depths and realizing a better scaling of deeper layers that way.

Acknowledgements

I gratefully acknowledges the support of the German Research Foundation (DFG) through an association with the CRC 1173 "Wavephenomena: Analysis and Numerics". A large part of the calculations in Chapter 5 were performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

APPENDIX

A

GREEN'S FUNCTIONS

In general a Green's function G is a function (or in some cases distribution) that solves

$$LG = \delta_0,$$

if L is a linear operator given by a linear differential equation with constant coefficients and δ_0 is Dirac's delta distribution i.e. $\delta_0(f) = f(0), \forall f \in C_c^\infty(\mathbb{R}^d)$. With this knowledge we can solve the differential equation

$$Ly = f$$

for any compatible right-hand side f. The solution then reads

$$y = G * f.$$

For the second order wave equation

$$\frac{1}{v_\mathrm{p}^2}\,\partial_t^2 G(\boldsymbol{x},t) - \Delta G(\boldsymbol{x},t) = \delta(\boldsymbol{x})\delta(t), \quad p(\boldsymbol{x},0) = 0\,, \quad \partial_t p(\boldsymbol{x},0) = 0\,,$$

we can easily find Green's functions in the literature [Ige16]. The Green's function for the wave equation is dependent on the dimension which will be be denoted as subindex. For an arbitrary source position $\boldsymbol{x}_{\mathrm{src},0}$ they are

$$\begin{aligned} G_1(\boldsymbol{x},t) &= \frac{1}{2v_\mathrm{p}} H\left(t - \frac{|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}|}{v_\mathrm{p}}\right), \\ G_2(\boldsymbol{x},t) &= \frac{1}{2v_\mathrm{p}^2\pi} \frac{H\left(t - \frac{||\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}||_2}{v_\mathrm{p}}\right)}{\sqrt{t^2 - \frac{||\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}||_2^2}{v_\mathrm{p}^2}}}, \\ G_3(\boldsymbol{x},t) &= \frac{1}{4\pi v_\mathrm{p}^2 ||\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}||_2} \delta\left(t - \frac{||\boldsymbol{x} - \boldsymbol{x}_{\mathrm{src},0}||_2}{v_\mathrm{p}}\right) \end{aligned}$$

With $H\colon \mathbb{R} \longrightarrow \mathbb{R},\ H(x) = 0$ if $x < 0, 1$ else. Assuming constant material parameters, we can transform our first order formulation into the equation above. Recall the equation from (2.8) with $L = 0$:

$$\begin{aligned} \varrho\,\partial_t v(\boldsymbol{x},t) &= \nabla p(\boldsymbol{x},t) \\ \frac{1}{\varrho v_\mathrm{p}^2}\,\partial_t p(\boldsymbol{x},t) &= \operatorname{div} v(\boldsymbol{x},t) + f(\boldsymbol{x},t) \end{aligned} \tag{A.1}$$

with $v(\boldsymbol{x},0) = \mathbf{0},\ p(\boldsymbol{x},0) = 0$ on the whole $\mathbb{R}^d, d \in \{2,3\}$ and $t \in [0,T]$. Note that for $f(\boldsymbol{x},0) = 0$, we also have that $\partial_t p(\boldsymbol{x},0) = 0$. Deriving the second equation in time yields

$$\frac{1}{\varrho v_{\mathrm{p}}^2}\,\partial_t^2 p(\boldsymbol{x},t) = \operatorname{div}\partial_t v(\boldsymbol{x},t) + \partial_t f(\boldsymbol{x},t)$$

plugging in the first equation we get

$$\frac{1}{v_{\mathrm{p}}^2}\,\partial_t^2 p(\boldsymbol{x},t) - \Delta p(\boldsymbol{x},t) = \varrho \cdot \partial_t f(\boldsymbol{x},t)$$

with $\Delta = \operatorname{div}(\nabla\cdot)$ being the Laplacian. This problem is equivalent to the system (A.1) for constant material parameters and well-defined with $p(\boldsymbol{x},0) = \partial_t p(\boldsymbol{x},0) = \mathbf{0}$. Likewise, we can do the same for the adjoint equation and get the same differential equation, but with terminal conditions

$$\frac{1}{v_{\mathrm{p}}^2}\,\partial_t^2 q(\boldsymbol{x},t) - \Delta q(\boldsymbol{x},t) = \varrho\partial_t g(\boldsymbol{x},t) \quad \boldsymbol{w}(\boldsymbol{x},T) = \mathbf{0}\ , \partial_t q(\boldsymbol{x},T) = q(\boldsymbol{x},T) = 0\,.$$

Therefore, the solution of the adjoint equation is given by

$$q(\boldsymbol{x},t) = G(\cdot, T-\cdot) * (-\partial_t g(\cdot, T-\cdot)\,.$$

APPENDIX

B

CALCULATION TIMES

We want to give a short overview over the different computing clusters we used. Some information on HoreKa, HORST and PDE (not all nodes have the same architecture, we only list the two we used which have the same architecture), the latter two are local nodes exclusive for the mathematics department is compiled in the table Fig. B.1. For typical calculation times see Fig. B.3

Name	# procs per node	RAM per node	nodes	proc name	clock frequency
PDE	128	1 TB	2	AMD EPYC 7713 64-Core	2.0 GHz
HoreKa	76	256 GB	570	Intel Xeon Platinum 8368	2.4 GHz
HORST	20	90 GB	12	Intel Xeon E5-2609 v2	2.5 GHz

Figure B.1: Comparison of the calculation clusters.

Problem (L, k)	dof_h	time steps	space-time dofs
Transmission $(5, 2)$	1198080	1200	$1.44 \cdot 10^9$
Camembert $(0, 1)$	245760	4200	$1.03 \cdot 10^9$
Marmousi $(3, 1)$	1618176	1050	$1.69 \cdot 10^9$

Figure B.2: Degrees of freedom for the three problems. It should be noted that although we can compare the numbers, the different amount of damping factors L and polynomial degree k change the structure of the matrix and therefore change the cost of matrix multiplications.

Problem/ Calc. Time	HORST	PDE	HoreKa
Transmission (E)	$\approx 62\,\mathrm{s}(32)$	$\approx 16\,\mathrm{s}$	–
Transmission (I)	$\approx 90\,\mathrm{s}$ (64)	$\approx 16\,\mathrm{s}$	–
Camembert (I)	–	–	$\approx 5-6\,\mathrm{s}(256)$
Marmousi (E)	$\approx 30\,\mathrm{s}(64)$	$\approx 40\,\mathrm{s}$	–
Marmousi (I)	–	$\approx 35\,\mathrm{s}$ (64)	–

Figure B.3: Calculation times of one forward or one adjoint wave equation on the different architectures, with (E) and (I) we denote whether the time discretization was explicit or implicit. The number in parentheses behind the time is the number of processors used for the calculation

BIBLIOGRAPHY

[ABG+13] Amir Asnaashari, Romain Brossier, Stéphane Garambois, François Audebert, Pierre Thore, and Jean Virieux. Regularized seismic full waveform inversion with prior model information. *GEOPHYSICS*, 78(2):25–36, 2013.

[AGO20] Hossein S Aghamiry, Ali Gholami, and Stéphane Operto. Accurate and efficient data-assimilated wavefield reconstruction in the time domain. *GEOPHYSICS*, 85(2):7–12, 2020.

[AMH09] Awad H. Al-Mohy and Nicholas J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal. Appl.*, 31(3):970–989, 2009.

[AMH11] Awad H. Al-Mohy and Nicholas J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM J. Sci. Comput.*, 33(2):488–511, 2011.

[Bau23] Niklas Baumgarten. *A Fully Parallelized and Budgeted Multi-level Monte Carlo Framework for Partial Differential Equations: From Mathematical Theory to Automated Large-Scale Computations.* PhD thesis, Karlsruher Institut für Technologie (KIT), 2023.

[BFE+21] Thomas Bohlen, Mario Ruben Fernandez, Johannes Ernesti, Christian Rheinbay, Andreas Rieder, and Christian Wieners. Visco-acoustic full waveform inversion: From a DG forward solver to a Newton-CG inverse solver. *Computers and Mathematics with Applications*, 100:126–140, 2021.

[BHDlPF16] Christian Boehm, Mauricio Hanzich, Josep De la Puente, and Andreas Fichtner. Wavefield compression for adjoint methods in full-waveform inversion. *GEOPHYSICS*, 81:385–397, 2016.

[Boh98] Thomas Bohlen. *Viskoelastische FD-Modellierung seismischer Wellen zur Interpretation gemessener Seismogramme.* PhD thesis, Christian-Albrechts-Universität zu Kiel, 1998.

[Boh02] Thomas Bohlen. Parallel 3-D viscoelastic finite difference seismic modelling. *Computers and Geosciences*, 28(8):887 – 899, 2002.

[BOV09] Romain Brossier, Stéphane Operto, and Jean Virieux. Robust elastic frequency-domain full-waveform inversion using the L1 norm. *Geophysical Research Letters*, 36(20), 2009.

[Bre11] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations.* Universitext. Springer, New York, 2011.

[BRS95] Joakim O. Blanch, Johan O. A. Robertsson, and William W. Symes. Modeling of a constant Q: Methodology and algorithm for an efficient and optimally inexpensive viscoelastic technique. *GEOPHYSICS*, 60(1):176–184, 1995.

[BS77] James H. Bramble and Albert H. Schatz. Higher order local accuracy by averaging in the finite element method. *Mathematics of Computation*, 31:94–111, 1977.

[BSZC95] Carey Bunks, Fatimetou M. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *GEOPHYSICS*, 60(5):1457–1473, 1995.

[BW21] Niklas Baumgarten and Christian Wieners. The parallel finite element system M++ with integrated multilevel preconditioning and multilevel Monte Carlo methods. *Comput. Math. Appl.*, 81:391–406, 2021.

[Car14] José M. Carcione. *Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic, Porous and Electromagnetic Media.* Elsevier Science, 4. edition, 2014.

[CDW22] Daniele Corallo, Willy Dörfler, and Christian Wieners. Space-time discontinuous Galerkin methods for weak solutions of hyperbolic linear symmetric Friedrichs systems. *Journal of Scientific Computing*, 94(1):27, 2022.

[CK19] David Colton and Rainer Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93 of *Applied Mathematical Sciences.* Springer, Cham, 1. edition, 2019.

[DCC17] Meng-Xue Dai, Jing-Bo Chen, and Jian Cao. l_1-regularized full-waveform inversion with prior model information based on orthant-wise limited memory quasi-Newton method. *Journal of Applied GEOPHYSICS*, 142:49–57, 2017.

[DWZ20] Willy Dörfler, Christian Wieners, and Daniel Ziegler. Space-time discontinuous Galerkin methods for linear hyperbolic systems and the application to the forward problem in seismic imaging. In *Finite Volumes for Complex Applications IX - Methods, Theoretical Aspects, Examples*, pages 477–485. Springer International Publishing, 2020.

[EG04] Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 2004.

[EGR22] Matthias Eller, Roland Griesmaier, and Andreas Rieder. Tangential cone condition for the full waveform forward operator in the elastic regime: the non-local case. *CRC 1173 Preprint 2022/48*, Karlsruhe Institute of Technology 2022.

[EGvL+18] Ernie Esser, Lluis Guasch, Tristan van Leeuwen, Aleksandr Y Aravkin, and Felix J Herrmann. Total variation regularization strategies in full-waveform inversion. *SIAM Journal on Imaging Sciences*, 11(1):376–406, 2018.

[EHN96] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and its Applications.* Kluwer Academic Publishers Group, Dordrecht, 1996.

[EHO+12] Vincent Etienne, Guanghui Hu, Stéphane Operto, Jean Virieux, Olav I. Barkved, and Jan H. Kommedal. Three-dimensional acoustic full waveform inversion-algorithm and application to Valhall. In *74th EAGE Conference and Exhibition incorporating EUROPEC 2012*, pages cp–293. European Association of Geoscientists & Engineers, 2012.

[ER21] Matthias Eller and Andreas Rieder. Tangential cone condition and Lipschitz stability for the full waveform forward operator in the acoustic regime. *Inverse Problems*, 37(8):085011, 2021.

[Ern18] Johannes Ernesti. *Space-Time Methods for Acoustic Waves with Applications to Full Waveform Inversion.* PhD thesis, Karlsruher Institut für Technologie (KIT), 2018.

[EY20] Bjorn Engquist and Yunan Yang. Optimal transport based seismic inversion: Beyond cycle skipping, 2020.

[Fic11] Andreas Fichtner. *Full Seismic Waveform Modelling and Inversion.* Springer Berlin, Heidelberg, 1. edition, 2011.

[FKIB09] Andreas Fichtner, Brian L. N. Kennett, Heiner Igel, and Hans-Peter Bunge. Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods. *Geophysical Journal International*, 179(3):1703–1725, 2009.

[FKIB10] Andreas Fichtner, Brian L.N. Kennett, Heiner Igel, and Hans-Peter Bunge. Full waveform tomography for radially anisotropic structure: New insights into present and past states of the australasian upper mantle. *Earth and Planetary Science Letters*, 290(3):270–280, 2010.

[Fut62] Walter I. Futterman. Dispersive body waves. *Journal of Geophysical Research (1896-1977)*, 67(13):5279–5291, 1962.

[GAO20] Ali Gholami, Hossein S. Aghamiry, and Stéphane Operto. Compound regularization of full-waveform inversion for imaging piecewise media. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):1192–1204, 2020.

[GAO21] Ali Gholami, Hossein S. Aghamiry, and Stéphane Operto. Extended-space full-waveform inversion in the time domain with the augmented Lagrangian method. *GEOPHYSICS*, 87(1):63–77, 2021.

[GFB+04] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, 2004.

[GIP20] Yu Geng, Kristopher A. Innanen, and Wenyong Pan. Subspace methods in multi-parameter seismic full waveform inversion. *Communications in Computational Physics*, 28(1):228–248, 2020.

[GSFB14] Lisa Groos, Martin Schäfer, Thomas Forbriger, and Thomas Bohlen. The role of attenuation in 2d full-waveform inversion of shallow-seismic body and Rayleigh waves. *GEOPHYSICS*, 79(6):R247–R261, 2014.

[GVT86] Odile Gauthier, Jean Jean Virieau, and Albert Tarantola. Two-dimensional nonlinear inversion of seismic waveforms: Numerical results. *GEOPHYSICS*, 51:1387–1403, 1986.

[HA01] Eldad Haber and Uri M. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17(6):1847, 2001.

[HKLS07] Markus Haltmeier, Richard Kowar, Antonio Leitão, and Otmar Scherzer. Kaczmarz methods for regularizing nonlinear ill-posed equations II: Applications. *Inverse Problems and Imaging*, 1(3):507–523, 2007.

[HLS07] Markus Haltmeier, Antonio Leitão, and Otmar Scherzer. Kaczmarz methods for regularizing nonlinear ill-posed equations I: convergence analysis. *Inverse Problems and Imaging*, 1(2):289–298, 2007.

[HNS16] Bamdad Hosseini, Nilima Nigam, and John M. Stockie. On regularizations of the Dirac delta distribution. *Journal of Computational Physics*, 305:423–447, 2016.

[HO10] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

[Hof99] Bernd Hofmann. On ill-posedness and local ill-posedness of operator equations in Hilbert spaces. *Preprint 97-8*, Faculty of Mathematics, D–09107 Chemnitz, Germany 1999.

[HPS$^+$15] Marlis Hochbruck, Tomislav Pazur, Andreas Schulz, Ekkachai Thawinan, and Christian Wieners. Efficient time integration for discontinuous Galerkin approximations of linear wave equations. *ZAMM*, 95(3):237–259, 2015.

[HR96] Martin Hanke and Toomas Raus. A general heuristic for choosing the regularization parameter in ill-posed problems. *SIAM Journal on Scientific Computing*, 17(4):956–972, 1996.

[HR99] Frank Hettlich and William Rundell. A second degree method for nonlinear inverse problems. *SIAM Journal on Numerical Analysis*, 37(2):587–620, 1999.

[HW08] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008.

[Ige16] Heiner Igel. *Computational Seismology: A Practical Introduction*. Oxford University Press, 1. edition, 2016.

[JLM12] Woodon Jeong, Ho-Yong Lee, and Dong-Joo Min. Full waveform inversion strategy for density in the frequency domain. *Geophysical Journal International*, 188(3):1221–1242, 2012.

[Kal16] Barbara Kaltenbacher. Regularization based on all-at-once formulations for inverse problems. *SIAM Journal on Numerical Analysis*, 54(4):2594–2618, 2016.

[KDNK$^+$12] Daniel Köhn, Denise De Nil, André Kurzmann, Anna Przebindowska, and Thomas Bohlen. On the influence of model parametrization in elastic full waveform tomography. *Geophysical Journal International*, 191(1):325–345, 2012.

[KFBI18] Lion Krischer, Andreas Fichtner, Christian Boehm, and Heiner Igel. Automated large-scale full seismic waveform inversion for North America and the North Atlantic. *Journal of Geophysical Research: Solid Earth*, 123(7):5902–5928, 2018.

[KNS08] Barbara Kaltenbacher, Andreas Neubauer, and Otmar Scherzer. *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*. De Gruyter, Berlin, New York, 2008.

[KR16] Andreas Kirsch and Andreas Rieder. Inverse problems for abstract evolution equations with applications in electrodynamics and elasticity. *Inverse Problems*, 32(8):085001, 24, 2016.

[KR19] Andreas Kirsch and Andreas Rieder. Inverse problems for abstract evolution equations II: Higher order differentiability for viscoelasticity. *SIAM Journal on Applied Mathematics*, 79(6):2639–2662, 2019.

[KSW88] Brian L. N. Kennett, Malcom S. Sambridge, and Paul R. Williamson. Subspace methods for large inverse problems with multiple parameter classes. *Geophysical Journal International*, 94(2):237–247, 1988.

[Kur12] André Kurzmann. *Applications of 2D and 3D full waveform tomography in acoustic and viscoacoustic complex media*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2012.

[LAH12] Y. Lin, A. Abubakar, and T. M. Habashy. Seismic full-waveform inversion using truncated wavelet representations. In *SEG Technical Program Expanded Abstracts 2012*, pages 1–6. Society of Exploration Geophysicists, 2012.

[LAK76] Hsi-Ping Liu, Don L Anderson, and Hiroo Kanamori. Velocity dispersion due to anelasticity; implications for seismology and mantle composition. *Geophysical Journal International*, 47(1):41–58, 1976.

[LeV02] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[Lyo97] Richard G Lyons. *Understanding digital signal processing, 3/E.* Pearson Education India, 1997.

[MBB+14] Ludovic Métivier, Francois Bretaudeau, Romain Brossier, Stéphane Operto, and Jean Virieux. Full waveform inversion and the truncated Newton method: quantitative imaging of complex subsurface structures. *Geophysical Prospecting*, 62(6):1353–1375, 2014.

[MHMG10] Yong Ma, Dave Hale, Zhaobo (Joe) Meng, and Bin Gong. Full waveform inversion with image-guided gradient. In *SEG Technical Program Expanded Abstracts 2010*, pages 1003–1007. Society of Exploration Geophysicists, 2010.

[MN98] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, 1998.

[MRL14] Fábio Margotti, Andreas Rieder, and Antonio Leitão. A Kaczmarz version of the REGINN-Landweber iteration for ill-posed problems in Banach spaces. *SIAM Journal on Numerical Analysis*, 52(3):1439–1465, 2014.

[MVdKG+10] Giovanni Angelo Meles, Jan Van der Kruk, Steward Greenhalgh, Jacques Ernst, Hansruedi Maurer, and Alan Green. A new vector waveform inversion algorithm for simultaneous updating of conductivity and permittivity parameters from combination crosshole/borehole-to-surface gpr data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9):3391–3407, 2010.

[MWM06] Gary Martin, Robert Wiley, and Kurt Marfurt. Marmousi2: An elastic upgrade for Marmousi. *The Leading Edge*, 25:156–166, 2006.

[NN91] Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, New York, 2. edition, 2006.

[NZ05] Guy Narkiss and Michael Zibulevsky. Sequential subspace optimization method for large-scale unconstrained optimization. Technical report, Technion - The Israel Institute of Technology, Department of Electrical Engineering, 2005.

[OGP+13] Stéphane Operto, Operto, Yasser Gholami, Vicent Prieux, Alessandra Ribodetti, Romain Brossier, Ludovic Métivier, and Jean Virieux. A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice. *The Leading Edge*, 32(9):1040–1054, 2013.

[OME93] Doug W. Oldenburg, P. R. McGillivray, and R. G. Ellis. Generalized subspace methods for large-scale inverse problems. *Geophysical Journal International*, 114(1):12–20, 1993.

[PR23] Lukas Pieronek and Andreas Rieder. On the iterative regularization of non-linear illposed problems in L^∞. *Numerische Mathematik*, 154:1–39, 2023.

[Pra99] R. Gerhard Pratt. Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *GEOPHYSICS*, 64(3):888–901, 1999.

[Rie99] Andreas Rieder. On the regularization of nonlinear ill-posed problems via inexact Newton iterations. *Inverse Problems*, 15(1):309–327, 1999.

[Rie05] Andreas Rieder. Inexact Newton regularization using conjugate gradients as inner iteration. *SIAM Journal on Numerical Analysis*, 43(2):604–622, 2005.

[Rie21] Andreas Rieder. An all-at-once approach to full waveform inversion in the viscoelastic regime. *Mathematical Methods in the Applied Sciences*, 44(8):6376–6388, 2021.

[RR04] Michael Renardy and Robert C. Rogers. *An introduction to partial differential equations.* Texts in applied mathematics 13. Springer, New York, NY, 2. edition, 2004.

[Saa03] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, 2. edition, 2003.

[Sch19] Roland Schnaubelt. Evolution equations, lecture notes, 2019.

[Shi19] Renat Shigapov. *Probabilistic Waveform Inversion: Quest for the Law.* PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.

[SJM01] Changsoo Shin, Seonghyung Jang, and Dong-Joo Min. Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophysical Prospecting*, 49(5):592–606, 2001.

[STK91] Malcom Sambridge, Albert Tarantola, and Brian Kennett. An alternative strategy for non-linear inversion of seismic waveforms. *Geophysical Prospecting*, 39(6):723–736, 1991.

[SV09] Thomas Strohmer and Roman Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[Sym20] William W. Symes. Wavefield reconstruction inversion: an example. *Inverse Problems*, 36(10):105010, 2020.

[Tar84] Albert Tarantola. Inversion of seismic reflection data in the acoustic approximation. *GEOPHYSICS*, 49(8):1259–1266, 1984.

[Tar05] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* Society for Industrial and Applied Mathematics, 10. edition, 2005.

[VAB+17] Jean Virieux, Amir Asnaashari, Romain Brossier, Ludovic Métivier, Alessandra Ribodetti, and Wei Zhou. An introduction to full waveform inversion. In *Encyclopedia of Exploration GEOPHYSICS*, chapter 6, pages R1–1–R1–40. Society of Exploration Geophysicists, 2017.

[vLH13] Tristan van Leeuwen and Felix J. Herrmann. Mitigating local minima in full-waveform inversion by expanding the search space. *Geophysical Journal International*, 195(1):661–667, 2013.

[vLH15] Tristan van Leeuwen and Felix J Herrmann. A penalty method for PDE-constrained optimization in inverse problems. *Inverse Problems*, 32(1):015007, 2015.

[VO09] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *GEOPHYSICS*, 74, 2009.

[Win16] Robert Winkler. *A Model-Aware Inexact Newton Scheme for Electrical Impedance Tomography.* PhD thesis, Karlsruher Institut für Technologie (KIT), 2016.

[WR15] Robert Winkler and Andreas Rieder. Model-aware Newton-type inversion scheme for electrical impedance tomography. *Inverse Problems*, 31(4):045009, 2015.

[WS17] Anne Wald and Thomas Schuster. Sequential subspace optimization for nonlinear inverse problems. *J. Inverse Ill-Posed Probl.*, 25(1):99–117, 2017.

[XM14] Kun Xu and George Mcmechan. 2d frequency-domain elastic full-waveform inversion using time-domain modeling and a multistep-length gradient approach. *GEOPHYSICS*, 79, 2014.

[YBM+18] Pengliang Yang, Romain Brossier, Ludovic Métivier, Jean Virieux, and Wei Zhou. A time-domain preconditioned ted Newton approach to visco-acoustic multiparameter full waveform inversion. *SIAM Journal on Scientific Computing*, 40(4):B1101–B1130, 2018.

[YS13] Yang Yang and Chi-Wang Shu. Discontinuous Galerkin method for hyperbolic equations involving delta-singularities: Negative-order norm error estimates and applications. *Numerische Mathematik*, 124, 2013.

[YZ23] Pengliang Yang and Wei Zhou. Algorithmic analysis towards time-domain extended source waveform inversion, arXiv 2211.06300, 2023.

[Zel19] Uwe Christian Zeltmann. *The Viscoelastic Seismic Model: Existence, Uniqueness and Differentiability with Respect to Parameters.* PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.

[Zie20] Daniel Alexander Ziegler. *A Parallel and Adaptive Space-Time Discontinuous Galerkin method for Visco-Elastic and Visco-Acoustic Waves.* PhD thesis, Karlsruhe Institut für Technologie, Karlsruhe, 2020.