The Numerical Scheme of Approximate Inverse for 2D Linear Seismic Imaging

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

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

Dissertation

von

Kevin Ganster

Tag der mündlichen Prüfung: 18.03.2025

1. Referent: Prof. Dr. Andreas Rieder
2. Referent: apl. Prof. Dr. Peer Christian Kunstmann

Abstract

The inverse problem of seismic imaging entails reconstructing material structures, e.g. of the Earth's subsurface, from measured reflected wave fields. In this thesis, we consider a linearized version of this problem about an a priori known wave speed. We address the implementation of an approximate inversion scheme in two dimensions and cover the basics of the theoretical background, which is microlocal analysis. We thoroughly test our implementation in a series of different numerical experiments, including an experiment on a field data set.

Contents

CHAPTER 1

Introduction

There are many reasons to understand the composition of the Earth's subsurface. One important reason is to assess if an area is suitable for construction. Knowing the geological conditions helps architects to determine which types of buildings or infrastructure can be safely built. For instance, areas with loose or unstable soil might be not suitable for all building projects.

Another significant reason is the exploration of natural resources. Especially historically, this was vital for discovering valuable resources, such as oil and natural gas. While the focus today shifted towards more renewable alternatives of those resources, subsurface studies still continue to play a crucial role.

One possible method to gather that information is to carry out drillings on the area of interest. However, this is expensive and in the case of searching for mineral resources we also don't know whether a good location was chosen.

Therefore, in seismic imaging we aim to reconstruct material structures in a non-invasive way purely from measurements of a controlled excitation of waves, for example with an air gun or a hammer blow. The idea is that a wave contains information about the variational structure of the material it travelled through. A wave is excited at a source location on the surface and then propagates through the different material layers in the subsurface. As the wave passes from one medium to another it is (partially) reflected. These reflections can then be picked up by receivers that are also located on the surface. The recorded seismograms can be used to reconstruct the underlying material structure. In Figure 1.1 we illustrate this wave propagation through different materials such as water and various layers of soil. We have highlighted different paths a wave could take to reach a single receiver on the water's surface.

To describe this process accurately, we need a mathematical model that characterizes the propagation of waves through the material structures in the subsurface. To this end, we choose the acoustic wave equation.

It is also important to know where the sources and receivers were placed during the recording. This is described by the acquisition geometry. In practice, raw data are often recorded using the *common-source geometry*, where a single source and multiple receivers in a line are used. A full survey of an area then consists of many such recordings with

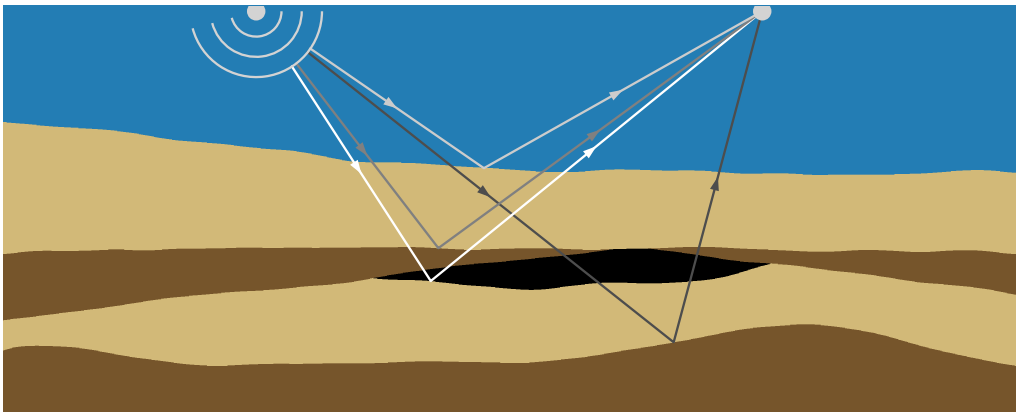


Fig. 1.1: Illustration of a wave traveling through various materials and its reflections going to the receiver on the surface. Left dot: source, right dot: receiver.

shifted source positions. It is possible to restructure this data set and treat it as if it was recorded with a different acquisition geometry. In this work, we will mainly use the *common-offset data acquisition geometry* with half offset $\alpha \geq 0$, meaning the distance of source and receiver will always be the same.

From a historical perspective, the first applications of seismic imaging can be traced as far back as 1910, where Mohorovičić analyzed seismograms of an earthquake to show that there is a surface seperating the Earth's crust from the mantle, according to [GO07]. In the following years, human-generated seismic waves were used and the methods further developed. John Clarence Karcher first applied the reflection seismic method in 1921 for petroleum exploration (see [Bro99]). In the book [BCS01] the authors outlined the developments in seismic imaging since then.

The goal of this work now is to present an algorithm to tackle seismic imaging for the acoustic wave equation in two space dimensions using different acquisition geometries, and in particular address the implementation of this scheme.

We begin by declaring the notation used throughout this thesis in Chapter 2. There, we also derive our setting and the approximate inversion algorithm starting from the acoustic wave equation, including the involved forward operator F.

Next, in Chapter 3, we state important definitions and results of microlocal analysis for the theoretical background of reconstruction operators for the seismic imaging problem. These include the concepts of Fourier integral and Pseudodifferential operators, and the wavefront set.

In Chapter 4 we first derive a representation of F as a Fourier integral operator and then present in detail the imaging operators we use.

Chapter 5 entails the implementation of the inversion scheme, including all components such as the solvers for the eikonal and transport equations arising in Chapter 2. Here, we

heavily make use of the common-offset geometry and an affine linear background velocity model to efficiently calculate the involved quantities. In the end, we present other commonly used acquisition geometries, and establish a connection back to the common-offset geometry. Finally, we outline the use of multiple measurements for a single reconstruction.

To test and validate our implementation, we conduct a comprehensive series of numerical experiments in Chapter 6. We finish that chapter with an experiment on a field data set.

Lastly, in Chapter 7, we present a summary of the thesis and provide an outlook for the three-dimensional case, including the potential challenges and changes we need to make.

CHAPTER 2

Notations and Setting

This chapter provides an overview of the notation used throughout this thesis. We derive our scheme starting from the acoustic wave equation and shortly present the types of imaging operators we use.

2.1 Notations

Given two topological vector spaces U and W, we denote by $\mathcal{L}(U, W)$ the set of linear continuous operators from U to W. We write U^* for the dual space $\mathcal{L}(U, \mathbb{R})$ of U. For an operator $T \in \mathcal{L}(U, W)$, we define the dual operator $T^*\colon W^* \to U^*$ by the relation $\langle T\boldsymbol{x}\,,\, \boldsymbol{y}^*\rangle_{W,\, W^*} = \langle \boldsymbol{x}\,,\, T^*\boldsymbol{y}^*\rangle_{U,\, U^*}$ for all $\boldsymbol{x} \in U$ and $\boldsymbol{y}^* \in W^*$. Here, $\langle \cdot\,,\, \cdot\rangle_{U,\, U^*}$ denotes the dual pairing, and we have $\boldsymbol{x}^*(\boldsymbol{x}) \coloneqq \langle \boldsymbol{x}\,,\, \boldsymbol{x}^*\rangle_{U,\, U^*}$ for all $\boldsymbol{x} \in U$ and $\boldsymbol{x}^* \in U^*$.

The k-th unit vector in $\mathbb{R}^d$ is denoted by $\boldsymbol{e_k}$. The partial derivative of a function $f : \mathbb{R}^d \to \mathbb{R},\ \boldsymbol{x} \mapsto f(\boldsymbol{x})$ will be denoted by $\partial_{x_k} f$, for $k = 1, \dots, d$, and the gradient will be given by $\nabla_{\boldsymbol{x}} f = (\partial_{x_1} f, \dots, \partial_{x_d} f)^\top$.

Let $X \neq \emptyset$ be an open subset of $\mathbb{R}^d$. The space of infinitely differentiable functions on $\mathbb{R}^d$ will be denoted by $\mathcal{C}^\infty(\mathbb{R}^d)$, and those functions which additionally have compact support in X make up $\mathcal{C}_c^\infty(X)$. We equip $\mathcal{C}^\infty(X)$ and $\mathcal{C}_c^\infty(X)$ with their standard topologies and denote them by $\mathcal{E}(X)$ and $\mathcal{D}(X)$, respectively. The dual space of $\mathcal{D}(X)$, denoted by $\mathcal{D}'(X)$, is the space of *distributions* or *generalized functions* on X. Analogously, the dual space $\mathcal{E}'(X)$ is the space of distributions with compact support on X. The dual spaces $\mathcal{D}'(X)$ and $\mathcal{E}'(X)$ are given the appropriate weak topologies, see [Rud73, Chapter 6]. Lastly, by $\mathcal{S}(\mathbb{R}^d)$ we denote the *Schwartz space* or space of rapidly decreasing functions

$$\mathcal{S}(\mathbb{R}^d) \coloneqq \left\{ f \in \mathcal{C}^\infty(\mathbb{R}^d)\colon \forall \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}_0^d\colon \sup_{\boldsymbol{x} \in \mathbb{R}^d} \left| \boldsymbol{x}^{\boldsymbol{\alpha}} D^{\boldsymbol{\beta}} f(\boldsymbol{x}) \right| < \infty \right\}.$$

Here, we use multi-index notation $\boldsymbol{x}^{\boldsymbol{\alpha}} \coloneqq x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d}$ and $D^{\boldsymbol{\beta}} \coloneqq \partial_{x_1}^{\beta_1} \partial_{x_2}^{\beta_2} \dots \partial_{x_d}^{\beta_d}$. The dual space $S'(\mathbb{R}^d)$ is then the space of tempered distributions. Again, $\mathcal{S}(\mathbb{R}^d)$ and $\mathcal{S}'(\mathbb{R}^d)$ are equipped with their appropriate topologies. To provide a better overview, all spaces are again summarized in Table 2.1.

Table 2.1: Summary of the different spaces and their dual counterparts.

$\mathcal{E}(X)$	$\mathcal{C}^\infty$ functions on X, i.e. the set of all infinitely differentiable functions on X.
$\mathcal{D}(X)$	$\mathcal{C}^\infty$ functions with compact support on X, also commonly known as $\mathcal{C}_c^\infty(X)$.
$\mathcal{S}(\mathbb{R}^d)$	Schwartz space, i.e. $\mathcal{C}^\infty$ functions on $\mathbb{R}^d$ that rapidly decrease at infinity along with their derivatives.
$\mathcal{E}'(X)$	space of distributions with compact support in X.
$\mathcal{D}'(X)$	space of distributions on X.
$\mathcal{S}'(\mathbb{R}^d)$	tempered distributions on $\mathbb{R}^d$.

When analyzing the growth of a function f, we use the symbol $\mathcal{O}$ to denote the well-known Landau notation. We will write $f = \mathcal{O}(g)$ or "f is $\mathcal{O}(g)$", if the asymptotic order of growth of f is bounded by the order of growth of a function g.

For any set $A \subseteq \mathbb{R}^d$ we define the characteristic function χ_A by

$$\chi_A(\boldsymbol{x}) \coloneqq \begin{cases} 1, & \boldsymbol{x} \in A, \\ 0, & \boldsymbol{x} \notin A, \end{cases} \qquad \boldsymbol{x} \in \mathbb{R}^d.$$

We write $B_p(\boldsymbol{x}; r) \coloneqq \left\{ \boldsymbol{v} \in \mathbb{R}^d \colon \|\boldsymbol{v} - \boldsymbol{x}\|_p < r \right\}$ for an open ball around $\boldsymbol{x} \in \mathbb{R}^d$ with radius r with respect to the p-Norm. In cases where the specific norm does not matter, we will instead write $B_r(\boldsymbol{x})$.

This list will be expanded by notations for the discrete case later on in the implementation Chapter 5.

2.2 The considered problem

The acoustic potential $u = u(t; \boldsymbol{x}, \boldsymbol{x_s})$ in $\boldsymbol{x} \in \mathbb{R}^d$, $d \in \{2, 3\}$ at time $t \geq 0$ satisfies the acoustic wave equation with constant mass density

$$\begin{aligned} \frac{1}{\nu_{\mathrm{pr}}^2} \partial_t^2 u - \Delta_{\boldsymbol{x}} u &= \delta(\boldsymbol{x} - \boldsymbol{x_s}) \delta(t), \\ u(0; \cdot, \boldsymbol{x_s}) = \partial_t u(0; \cdot, \boldsymbol{x_s}) &= 0, \end{aligned} \tag{2.2.1}$$

where $\nu_{\mathrm{pr}} = \nu_{\mathrm{pr}}(\boldsymbol{x})$ is the speed of sound and δ denotes the Dirac distribution in the related variables. The pressure wave is excited at location $\boldsymbol{x_s}$ and time $t = 0$. This accurately models wave propagation without damping in a medium ν_{pr} that does not admit shear stress.

The goal of seismic imaging is to recover ν_{pr} in an open set X from measurements of $u(t; \boldsymbol{x_r}, \boldsymbol{x_s})$ in the observation period $[0, t_{\max}]$, where $\boldsymbol{x_r}$ indicates receiver points, see Figure 1.1. Here, X is a subset of the lower half space

$$X \subseteq \mathbb{R}^d_+ \coloneqq \mathbb{R}^{d-1} \times [0, \infty),$$

where the positive x_d-coordinate points downwards into the Earth and therefore represents the depth-axis.

We make a linearization ansatz for this nonlinear inverse problem using a smooth background velocity $c = c(\boldsymbol{x})$, that is known in advance (see [BC79])

$$\frac{1}{\nu^2_{\text{pr}}(\boldsymbol{x})} = \frac{1 + n(\boldsymbol{x})}{c^2(\boldsymbol{x})}, \tag{2.2.2}$$

where n is compactly supported in X. Further, n is now the quantity of interest that captures the jumps (singularities) of ν_{pr} (see e.g. [BCS01, Chapter 3.2.1]).

The upcoming is followed from [BC79] and [Gra20, Section 1.2]. Let $\widetilde{u} = \widetilde{u}(t; \boldsymbol{x}, \boldsymbol{x_s})$ be the solution for the acoustic wave equation (2.2.1) with ν_{pr} replaced by c

$$\begin{aligned} \frac{1}{c^2}\partial_t^2 \widetilde{u} - \Delta_{\boldsymbol{x}} \widetilde{u} &= \delta(\boldsymbol{x} - \boldsymbol{x_s})\delta(t), \\ \widetilde{u}(0; \cdot, \boldsymbol{x_s}) = \partial_t \widetilde{u}(0; \cdot, \boldsymbol{x_s}) &= 0. \end{aligned} \tag{2.2.3}$$

By plugging in (2.2.2) into (2.2.1) and then subtracting (2.2.1) from (2.2.3), we get

$$\frac{1}{c^2}\partial_t^2(\widetilde{u} - u) - \Delta_{\boldsymbol{x}}(\widetilde{u} - u) = \frac{n}{c^2}\partial_t^2 u.$$

If we replace u by $\widetilde{u}$ on the right-hand side we perform the *Born approximation* (see [Dem21, Section 4.1]). By doing this, we disregard any multiple scattering, meaning we assume from now on, that only single scattering occurs. With $u_b \coloneqq \widetilde{u} - u$, we now obtain the linear operator

$$L \colon n \mapsto u_b|_{\mathcal{R}}, \qquad \text{where } u_b \text{ solves} \quad \frac{1}{c^2}\partial_t^2 u_b - \Delta_{\boldsymbol{x}} u_b = \frac{n}{c^2}\partial_t^2 \widetilde{u}, \tag{2.2.4}$$

and $\mathcal{R}$ is the set of receivers.

To avoid confusion, we now write ∂_1 for the derivative with respect to the first variable, which is time. Note that $\widetilde{u}$ is the fundamental solution for the equation in (2.2.4). Using Duhamel's principle, a solution to equation (2.2.4) is (see [Dem21, Proposition 2])

$$\begin{aligned} Ln(t;\boldsymbol{x_r},\boldsymbol{x_s}) &= \int_0^t \int_X \widetilde{u}(t-\eta;\boldsymbol{x_r},\boldsymbol{x}) \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \partial_1^2 \widetilde{u}(\eta;\boldsymbol{x},\boldsymbol{x_s})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\eta \\ &= \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \left(\int_0^t \partial_1^2 \widetilde{u}(\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(t-\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta \right) \mathrm{d}\boldsymbol{x} \\ &= \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \left(\int_0^t \partial_1^2 \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta \right) \mathrm{d}\boldsymbol{x}. \end{aligned}$$

By applying the Leibniz integral rule, we further calculate

$$\begin{aligned} &\partial_t^2 \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x} \\ &= \partial_t \Bigg(\int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t \partial_1 \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x} \\ &\qquad\qquad + \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \widetilde{u}(0;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(t;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \Bigg) \\ &= \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t \partial_1^2 \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x} \\ &\quad + \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \underbrace{\partial_1 \widetilde{u}(0;\boldsymbol{x},\boldsymbol{x_s})}_{=0} \widetilde{u}(t;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_X \frac{n(x)}{c^2(\boldsymbol{x})} \underbrace{\widetilde{u}(0;\boldsymbol{x},\boldsymbol{x_s})}_{=0} \widetilde{u}(t;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t \partial_1^2 \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x}, \end{aligned}$$

where we used the initial conditions from the wave equation (2.2.3) in the last step.

This leaves us with

$$Ln(t;\boldsymbol{x_r},\boldsymbol{x_s}) = \partial_t^2 \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t \widetilde{u}(t-\eta;\boldsymbol{x},\boldsymbol{x_s}) \widetilde{u}(\eta;\boldsymbol{x_r},\boldsymbol{x})\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x}. \tag{2.2.5}$$

From now on, we only consider the two-dimensional case $d = 2$. We further assume that there exists one and only one ray connecting $\boldsymbol{x} \in \mathrm{supp}(n)$ with each $\boldsymbol{x_s}$ and $\boldsymbol{x_r}$. This is called the *geometric optics assumption*. Under this assumption, we can approximate $\widetilde{u}$ by the *progressing wave expansion* (see [CH89, Chapter VI, §4, 3.], [Sym98, Chapter 5])

$$\widetilde{u}(t;\boldsymbol{x},\boldsymbol{x_s}) \approx a(\boldsymbol{x},\boldsymbol{x_s})\ \Psi\left(t-\tau(\boldsymbol{x},\boldsymbol{x_s})\right), \qquad \text{where} \quad \Psi(t) = \frac{1}{2\pi} \begin{cases} t^{-\frac{1}{2}}, & t > 0, \\ 0, & t \le 0. \end{cases} \tag{2.2.6}$$

The involved *traveltime* τ solves the eikonal equation

$$\|\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s})\|^2 = \frac{1}{c^2(\boldsymbol{x})}, \quad \tau(\boldsymbol{x_s},\boldsymbol{x_s}) = 0, \tag{2.2.7}$$

and *amplitude* a the transport equation

$$\begin{aligned} 2\nabla_{\boldsymbol{x}}a(\boldsymbol{x},\boldsymbol{x_s})\cdot\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s}) + a(\boldsymbol{x},\boldsymbol{x_s})\Delta_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s}) = 0, \\ \lim_{\boldsymbol{x}\to\boldsymbol{x_s}} a(\boldsymbol{x},\boldsymbol{x_s})\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|} = \frac{\sqrt{c(\boldsymbol{x_s})}}{2\sqrt{2\pi}}, \end{aligned} \tag{2.2.8}$$

see [LQB14, equation (8)].

The ansatz (2.2.6) is inspired by the fundamental solution for the wave equation. This crucial connection is detailed in Remark 2.1.

Remark 2.1. In 2D, the fundamental solution to the wave equation

$$\frac{1}{c^2}\partial_t^2\widetilde{u} - \Delta\widetilde{u} = \delta(\boldsymbol{x}-\boldsymbol{x_s})\delta(t)$$

with constant $c > 0$ is

$$\widetilde{u}(t;\boldsymbol{x},\boldsymbol{x_s}) = \frac{H(ct-\|\boldsymbol{x}-\boldsymbol{x_s}\|)}{2\pi\sqrt{c^2t^2-\|\boldsymbol{x}-\boldsymbol{x_s}\|^2}} = \frac{\Psi(ct-\|\boldsymbol{x}-\boldsymbol{x_s}\|)}{\sqrt{ct+\|\boldsymbol{x}-\boldsymbol{x_s}\|}},$$

where $H(t) = \int_{-\infty}^{t}\delta(\eta)\,\mathrm{d}\eta$ refers to the *Heaviside step function*. Since $\tau(\boldsymbol{x},\boldsymbol{x_s}) = \frac{\|\boldsymbol{x}-\boldsymbol{x_s}\|}{c}$ in this constant case, we have

$$\widetilde{u}(t;\boldsymbol{x},\boldsymbol{x_s}) = \frac{\Psi(c\,(t-\tau(\boldsymbol{x},\boldsymbol{x_s})))}{\sqrt{c\,(t+\tau(\boldsymbol{x},\boldsymbol{x_s}))}} = \frac{1}{c\sqrt{t+\tau(\boldsymbol{x},\boldsymbol{x_s})}}\Psi(t-\tau(\boldsymbol{x},\boldsymbol{x_s})).$$

By switching from the constant velocity to a space-dependent $c(\boldsymbol{x})$, this no longer holds, but it suggests the following form of $\widetilde{u}$

$$\widetilde{u}(t;\boldsymbol{x},\boldsymbol{x_s}) = a(\boldsymbol{x},\boldsymbol{x_s})\Psi(t-\tau(\boldsymbol{x},\boldsymbol{x_s})) + R(t,\boldsymbol{x},\boldsymbol{x_s}),$$

where the remainder R is not necessarily small, but smoother than the rest and therefore not relevant for analyzing the leading singularities of the solution. For a more in-depth explanation we refer to [Sym98, Chapter 5].

Inserting the progressing wave expansion (2.2.6) into equation (2.2.5), then using the relation $H' = \delta$, the equations from (A.1.1), and again the Leibniz integral rule, we now have

$$\begin{aligned} Ln(t;\boldsymbol{x_r},\boldsymbol{x_s}) &\approx \partial_t^2 \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t a(\boldsymbol{x},\boldsymbol{x_s})\Psi(t-\eta-\tau(\boldsymbol{x},\boldsymbol{x_s}))\; a(\boldsymbol{x_r},\boldsymbol{x})\Psi(\eta-\tau(\boldsymbol{x_r},\boldsymbol{x}))\,\mathrm{d}\eta\,\mathrm{d}\boldsymbol{x} \\ &= \partial_t^2 \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} a(\boldsymbol{x},\boldsymbol{x_s})a(\boldsymbol{x_r},\boldsymbol{x}) \underbrace{\int_0^t \Psi(t-\eta-\tau(\boldsymbol{x},\boldsymbol{x_s}))\Psi(\eta-\tau(\boldsymbol{x_r},\boldsymbol{x}))\,\mathrm{d}\eta}_{=\Psi\star\Psi(t-\tau(\boldsymbol{x},\boldsymbol{x_s})-\tau(\boldsymbol{x_r},\boldsymbol{x}))}\,\mathrm{d}\boldsymbol{x} \\ &= \partial_t \int_X \frac{n(\boldsymbol{x})}{4\pi c^2(\boldsymbol{x})} a(\boldsymbol{x},\boldsymbol{x_s})a(\boldsymbol{x_r},\boldsymbol{x})\; \partial_t H(t-\tau(\boldsymbol{x},\boldsymbol{x_s})-\tau(\boldsymbol{x_r},\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \\ &= \partial_t \int_X \frac{n(\boldsymbol{x})}{4\pi c^2(\boldsymbol{x})} a(\boldsymbol{x},\boldsymbol{x_s})a(\boldsymbol{x_r},\boldsymbol{x})\; \delta(t-\tau(\boldsymbol{x},\boldsymbol{x_s})-\tau(\boldsymbol{x_r},\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}. \end{aligned}$$

By integrating over time and multiplying with 4π, we can define the operator F as

$$Fn(t;\boldsymbol{x_r},\boldsymbol{x_s}) \coloneqq \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} a(\boldsymbol{x},\boldsymbol{x_s})a(\boldsymbol{x_r},\boldsymbol{x})\; \delta(t-\tau(\boldsymbol{x},\boldsymbol{x_s})-\tau(\boldsymbol{x_r},\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x},$$

and the right-hand side

$$g(t;\boldsymbol{x_r},\boldsymbol{x_s}) \coloneqq 4\pi \int_0^t (\widetilde{u}-u)(\eta;\boldsymbol{x_r},\boldsymbol{x_s})\,\mathrm{d}\eta.$$

We now proceed with an assumption regarding the acquisition geometry: pairs of source and receiver positions can be smoothly parametrized by a variable $s \in S \subset \mathbb{R}$, meaning $\boldsymbol{x_s} = \boldsymbol{x_s}(s)$ and $\boldsymbol{x_r} = \boldsymbol{x_r}(s)$. For instance, for the *common-offset acquisition geometry* with half offset $\alpha \geq 0$, source and receiver positions are given by

$$\boldsymbol{x_s}(s) = (s-\alpha,0)^\top, \qquad \boldsymbol{x_r}(s) = (s+\alpha,0)^\top, \qquad s \in S. \tag{2.2.9}$$

We set

$$\varphi(s,\boldsymbol{x}) \coloneqq \tau(\boldsymbol{x},\boldsymbol{x_s}(s)) + \tau(\boldsymbol{x},\boldsymbol{x_r}(s)) \quad \text{and} \quad A(s,\boldsymbol{x}) \coloneqq \frac{a(\boldsymbol{x},\boldsymbol{x_s}(s))a(\boldsymbol{x},\boldsymbol{x_r}(s))}{c^2(\boldsymbol{x})}. \tag{2.2.10}$$

Therefore our problem now reads: recover n from

$$Fn(s,t) = g(s,t), \qquad (s,t) \in S \times (0,t_{\max}), \tag{2.2.11}$$

with

$$Fn(s,t) = \int_X n(\boldsymbol{x})A(s,\boldsymbol{x})\,\delta(t-\varphi(s,\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \tag{2.2.12}$$

and

$$g(s,t) = 4\pi \int_0^t (\widetilde{u}-u)(\eta;\boldsymbol{x_r}(s),\boldsymbol{x_s}(s))\,\mathrm{d}\eta. \tag{2.2.13}$$

In general, an inversion formula for F is not known, as we will see later in Section 6.1.3. For this reason, we try to find so called *reconstruction operators*, that preserve the original structure of n as much as possible. The reconstruction operators we consider are motivated by approximate inversion formulas similar to those in Kirchhoff migration, see e.g. [BCS01]. These can be expressed as $F^\dagger KFn$ with a convolution filter K, where $F^\dagger$ is the *generalized backprojection*

$$F^\dagger g(\boldsymbol{x}) = \int_S \int_0^{t_{\max}} A(s,\boldsymbol{x}) g(s,t)\, \delta(t-\varphi(s,\boldsymbol{x}))\, \mathrm{d}t\, \mathrm{d}s = \int_S A(s,\boldsymbol{x}) g(s,\varphi(s,\boldsymbol{x}))\, \mathrm{d}s.$$

Similar to the backprojection of the classical Radon transform, see [Hel11], where one integrates over the set of all lines containing the point $\boldsymbol{x}$, we here integrate over the set of all contour lines of φ that contain $\boldsymbol{x}$. Therefore this can be understood as a generalized version of the backprojection for the classical Radon transform. In [Bey85], Beylkin showed that the imaging operator $F^\dagger KFn$ is close to the identity operator, in the sense that it can be divided into a partial identity operator (a low-pass filter) and a smoothing operator.

Another approach proposed by the authors in [NS97; De +09; Sto00; Fel+16] is to use the normal operator F^*F, which was then extended to operators of the form $KF^\dagger F$ in [Gra+17; Gra+18; Gra+20; KQR23; GR23], where K is a suitable pseudodifferential operator.

In summary, we consider reconstruction operators of the following two types:

$$\frac{1}{2\pi} K F_W^\dagger \psi F \qquad \text{and} \qquad \frac{1}{2\pi} F_W^\dagger M \psi F, \tag{2.2.14}$$

where ψ is a smooth and compactly supported cutoff function and K and M are suitable (pseudo-)differential operators. Further, $F_W^\dagger$ is the generalized backprojection with weight $W \in \mathcal{E}(S \times X)$, that is

$$F_W^\dagger g(\boldsymbol{x}) = \int_S W(s,\boldsymbol{x}) g(s,\varphi(s,\boldsymbol{x}))\, \mathrm{d}s.$$

Note that $F_W^\dagger$ is the formal L^2-adjoint of

$$F_W n(s,t) \coloneqq \int_X n(\boldsymbol{x}) W(s,\boldsymbol{x}) \delta(t-\varphi(s,\boldsymbol{x}))\, \mathrm{d}\boldsymbol{x},$$

(see (A.2.1)) and the formal L^2-adjoint $F^\star$ has weight $W = A$. The cutoff function ψ is needed so that $F^\dagger$ and F can be composed in general, see Chapter 4 for more details.

To cover both types of reconstruction operators, set

$$\Lambda \coloneqq \frac{1}{2\pi} K F_W^\dagger M \psi F.$$

The quality and suitability of those imaging operators depend on how well they can emulate the identity operator. As mentioned earlier, to the best of our knowledge there is no Λ to yield the identity operator. Therefore we study choices of the pseudodifferential operators K and M and weights W to understand which features of n will be recovered, emphasized or de-emphasized by the corresponding imaging operator Λ. Our choices for K, M and W will be discussed in Chapter 4.

We want to calculate Λn at $\boldsymbol{p} \in X$. But this is not necessarily possible depending on which space n belongs to, since in general $n \in \mathcal{E}'(X)$. Therefore we need an approximation for $\Lambda n(\boldsymbol{p})$. The concept of *approximate inverse* fits the structure of Λ perfectly.

2.2.1 Approximate inverse

By the concept of *approximate inverse* (see [Lou96]), instead of $\Lambda n(\boldsymbol{p})$, we compute the approximation $\Lambda n(\boldsymbol{p}) \approx \Lambda n * e_\gamma(\boldsymbol{p})$ for a family of mollifiers $\{e_\gamma\}_{\gamma>0}$ that are compactly supported in $\overline{B_\gamma(0)}$. A mollifier fulfills the following properties:

$$\begin{aligned} &(1) \quad \text{it is compactly supported,} \\ &(2) \quad \int e_\gamma(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1, \\ &(3) \quad \lim_{\gamma\to 0} e_\gamma = \delta. \end{aligned} \tag{2.2.15}$$

The parameter γ acts as a regularization parameter, since the smaller γ is, the smaller the support of e_γ and therefore the size of the neighbourhood where the approximation of the δ-distribution does not vanish.

With a fixed and sufficiently small $\gamma > 0$ it then holds

$$\begin{aligned} \Lambda n(\boldsymbol{p}) \approx \Lambda n * e_\gamma(\boldsymbol{p}) &= \frac{1}{2\pi}\left\langle K F_W^\dagger M \psi F n\,,\, e_\gamma(\cdot - \boldsymbol{p})\right\rangle_{\mathcal{D}'(X),\mathcal{D}(X)} \\ &= \frac{1}{2\pi}\left\langle \psi F n\,,\, M^\star F_W K^\star e_\gamma(\cdot - \boldsymbol{p})\right\rangle_{\mathcal{E}'(Y),\mathcal{E}(Y)} \\ &= \left\langle \psi F n\,,\, v_{\boldsymbol{p},\gamma,W}\right\rangle_{\mathcal{E}'(Y),\mathcal{E}(Y)} \\ &= \int_S \int_0^{t_{\max}} \psi(s,t) g(s,t) v_{\boldsymbol{p},\gamma,W}(s,t)\,\mathrm{d}t\,\mathrm{d}s, \end{aligned} \tag{2.2.16}$$

where $Y \coloneqq S \times (0, t_{\max})$, and the *reconstruction kernel*

$$v_{\boldsymbol{p},\gamma,W} = \frac{1}{2\pi} M^\star F_W\, K^\star\, e_\gamma(\cdot - \boldsymbol{p}).$$

For $\rho : X \to \mathbb{R}$ define

$$\widetilde{F}\rho(s,t) \coloneqq \int_X \rho(\boldsymbol{x})\,\delta(t - \varphi(s,\boldsymbol{x}))\,\mathrm{d}x.$$

Then $\widetilde{F}(W(s,\cdot)n) = F_W n$. Now we claim that we can reformulate $\widetilde{F}p(s,t)$ as follows.

Lemma 2.2.
Let $\widetilde{F}$ defined as above. Then

$$\begin{aligned}\widetilde{F}\rho(s,t) &= \int_{\mathcal{L}_{(s,t)}} \frac{\rho(s,\boldsymbol{x})}{\|\nabla_x \varphi(s,\boldsymbol{x})\|}\,\mathrm{ds}(\boldsymbol{x}) \\ &= \frac{1}{\sqrt{2}} \int_{\mathcal{L}_{(s,t)}} \frac{c(\boldsymbol{x})\rho(s,\boldsymbol{x})}{\sqrt{1+c^2(\boldsymbol{x})\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s}(s))\cdot\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_r}(s))}}\,\mathrm{ds}(\boldsymbol{x}),\end{aligned} \tag{2.2.17}$$

by integrating over reflection isochrones

$$\mathcal{L}_{(s,t)} \coloneqq \{\boldsymbol{x}\in X\colon \varphi(s,\boldsymbol{x}) = t\}\,, \qquad s\in S,\ t\in(0,t_{\max}). \tag{2.2.18}$$

Proof. It holds

$$\begin{aligned}\tilde{F}\rho(s,t) &= \int_X \rho(\boldsymbol{x})\,\delta(t-\varphi(s,\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \\ &= \lim_{h\searrow 0}\frac{1}{h}\int_{\bigcup\{\mathcal{L}_{(s,r)}\colon t\le r\le t+h\}} \rho(s,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \lim_{h\searrow 0}\frac{1}{h}\int_t^{t+h}\int_{\mathcal{L}_{(s,r)}} \frac{\rho(s,\boldsymbol{x})}{\|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})\|}\,\mathrm{ds}(\boldsymbol{x})\,\mathrm{d}r \quad \text{(coarea formula)} \\ &= \int_{\mathcal{L}_{(s,t)}} \frac{\rho(s,\boldsymbol{x})}{\|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})\|}\,\mathrm{ds}(\boldsymbol{x}). \quad \text{(Lebesgue differentiation thm.)}\end{aligned}$$

Further, with the abbreviations $c = c(\boldsymbol{x})$, $\tau_s = \tau(\boldsymbol{x},\boldsymbol{x_s}(s))$ and $\tau_r = \tau(\boldsymbol{x},\boldsymbol{x_r}(s))$

$$\begin{aligned}\|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})\| = \|\nabla_{\boldsymbol{x}}\tau_s + \nabla_{\boldsymbol{x}}\tau_r\| &= \sqrt{(\partial_{x_1}\tau_s+\partial_{x_1}\tau_r)^2+(\partial_{x_2}\tau_s+\partial_{x_2}\tau_r)^2} \\ &= \sqrt{\|\nabla_{\boldsymbol{x}}\tau_s\|^2+\|\nabla_{\boldsymbol{x}}\tau_r\|^2+2\nabla_{\boldsymbol{x}}\tau_s\cdot\nabla_{\boldsymbol{x}}\tau_r} \\ &= \sqrt{c^{-2}+c^{-2}+2\nabla_{\boldsymbol{x}}\tau_s\cdot\nabla_{\boldsymbol{x}}\tau_r} \quad \text{(eikonal eq. (2.2.7))} \\ &= \frac{\sqrt{2}}{c}\sqrt{1+c^2\nabla_{\boldsymbol{x}}\tau_s\cdot\nabla_{\boldsymbol{x}}\tau_r}.\end{aligned}$$

This justifies the representation (2.2.17). □

The reconstruction kernel can therefore be calculated by using the following formula

$$v_{\boldsymbol{p},\gamma,W}(s,t) = \frac{1}{2\pi}M^\star\,\dot{v}_{\boldsymbol{p},\gamma,W}(s,t) \tag{2.2.19}$$

with

$$\dot{v}_{\boldsymbol{p},\gamma,W}(s,t) \coloneqq \frac{1}{\sqrt{2}}\int_{\mathcal{L}_{(s,t)}} \frac{c(\boldsymbol{x})W(s,\boldsymbol{x})\ K^\star e_\gamma(\boldsymbol{x}-\boldsymbol{p})}{\sqrt{1+c^2(\boldsymbol{x})\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s})\cdot\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_r})}}\,\mathrm{ds}(\boldsymbol{x}). \tag{2.2.20}$$

Note that e_γ is assumed to be known explicitly, therefore we can calculate an explicit expression of $K^\star e_\gamma(\cdot - \boldsymbol{p})$ for suitable K. This will be handled in Chapter 4.

To summarize, our procedure looks as follows:

1. Numerically solve the eikonal (2.2.7) and transport equations (2.2.8). The algorithms we use for that will be presented in Chapter 5.

2. Find the reflection isochrones $\mathcal{L}_{(s,t)}$ for every discrete s and t. This can be done efficiently with the Marching Squares algorithm, which is the 2D version of the Marching Cubes algorithm by [LC87].

3. Calculate reconstruction kernels $v_{\boldsymbol{p},\gamma,W}$ for every point $\boldsymbol{p}$ in the reconstruction domain using (2.2.20) and a trapezoidal rule to approximate the integral and finite differences of order 2 for $\nabla_{\boldsymbol{x}}\tau$ therein.

4. Evaluate the (L^2-) scalar product $\langle \psi g\,,\, v_{\boldsymbol{p},\gamma,W}\rangle \approx \Lambda n(\boldsymbol{p})$ in equation (2.2.16) by using a trapezoidal sum.

CHAPTER 3

Microlocal basics

Here we collect important definitions and known results about microlocal analysis from the literature. When deemed appropriate, we provide some examples.

Microlocal analysis derived from the theory of partial differential equations and Fourier analysis in the 1960s and 1970s. The term originates from joint work by Mikio Satō, Takahiro Kawai and Masaki Kashiwara, see [Sjö00, Section 2]. Notably, Lars Hörmander and Mikio Satō played significant roles to develop the theory of microlocal analysis.

We will only cover the definitions and results necessary for our presentation in the following chapters. For a full survey of all important results concerning microlocal analysis, we refer the reader to publications of Lars Hörmander, namely [Hör65] and [Hör71]. In addition, we recommend the book [Pet83] by Bent E. Petersen and, in particular, the two-volume work by François Trèves [Trè80; Trè81] for a more elaborate presentation of the theory. For a first impression of microlocal analysis, we suggest the article [KQ15] by Venkateswaran P. Krishnan and Eric Todd Quinto.

The Fourier transform plays an important role in this chapter. We will use the following form for the Fourier transform of a function $f \in L^1(\mathbb{R}^d)$

$$(\mathcal{F}f)(\boldsymbol{\xi}) \coloneqq \widehat{f}(\boldsymbol{\xi}) \coloneqq \frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{-\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x},$$

and the inverse Fourier transform will be given by

$$(\mathcal{F}^{-1}\widehat{f})(\boldsymbol{x}) \coloneqq \frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \widehat{f}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi} = f(\boldsymbol{x}).$$

The Fourier transform is a linear continuous map from $\mathcal{S}(\mathbb{R}^d)$ to $\mathcal{S}(\mathbb{R}^d)$, see [Rud73, Theorem 7.7], and can be extended to a linear continuous map from $\mathcal{S}'(\mathbb{R}^d)$ to $\mathcal{S}'(\mathbb{R}^d)$ according to [Rud73, Theorem 7.15].

Let $X \subseteq \mathbb{R}^d$ be an open subset. As the title of this chapter suggests, we want to analyze properties of distributions in a local sense. To this end, we introduce the notion of a conic neighborhood.

Definition 3.1. (conic neighborhood)
Let $\boldsymbol{\xi_0} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$. A set $V \subseteq \mathbb{R}^N \setminus \{\mathbf{0}\}$ is called *conic neighborhood* of $\boldsymbol{\xi_0}$ if

- $\boldsymbol{\xi_0} \in V$,
- $B_\varepsilon(\boldsymbol{\xi_0}) \subseteq V$ for some $\varepsilon > 0$, and
- $\boldsymbol{\xi} \in V \Rightarrow \lambda\boldsymbol{\xi} \in V$ for all $\lambda > 0$.

A further tool of localization will be a *cutoff function*. This is simply a compactly supported $\mathcal{C}^\infty$ function $\psi \in \mathcal{D}(X)$ that is different from 0 at some point $\boldsymbol{x_0} \in X$, meaning $\psi(\boldsymbol{x_0}) \neq 0$. Note that it is equivalent to use a $\psi \in \mathcal{D}(X)$ that is identically 1 near $\boldsymbol{x_0}$, see [Hör03, Chaper VIII].

3.1 The $\mathcal{C}^\infty$ wavefront set

As we are interested in changes of a material, we want to investigate where non-smooth structures are. This is where the following definition comes into play.

Definition 3.2. (singular support)
For a distribution $u \in \mathcal{D}'(X)$ we define the *singular support* of u as

$$\operatorname{ssup}(u) \coloneqq X \setminus \{\boldsymbol{x} \in X : u \text{ coincides with a } \mathcal{C}^\infty \text{ function in a neighborhood of } \boldsymbol{x}\}.$$

Or, more precicely, a point $\boldsymbol{x_0} \in X$ belongs to the complement of $\operatorname{ssup}(u)$ if there is a neighborhood $U \subset X$ of $\boldsymbol{x_0}$ and a cutoff function $\psi \in \mathcal{D}(U)$, $\psi(\boldsymbol{x_0}) \neq 0$, so that $\psi u \in \mathcal{E}(X)$ (see [Hör03, Definition 2.2.3] or [Sjö00, p. 970]).

Therefore, $\operatorname{ssup}(u)$ is the complement in X of the largest open set on which u is smooth.

For instance, for the characteristic function of a set $B \subset \mathbb{R}^d$, we have $\operatorname{ssup}(\chi_B) = \partial B$.

The smoothness of a distribution $u \in \mathcal{E}'(\mathbb{R}^d) \subseteq \mathcal{S}'(\mathbb{R}^d)$ is closely related to the following property of its Fourier transform, which is more accessible than checking the smoothness of u.

Definition 3.3. (rapid decay)
A function $f : \mathbb{R}^d \to \mathbb{C}$ is *rapidly decaying (at infinity)* on the cone $V \subseteq \mathbb{R}^d \setminus \{\mathbf{0}\}$ if for every $N \in \mathbb{N}_0$ there is a constant $C_N > 0$ so that

$$|f(\boldsymbol{x})| \leq C_N (1 + \|\boldsymbol{x}\|)^{-N}, \qquad \forall \boldsymbol{x} \in V.$$

Note that the Schwartz space $\mathcal{S}(\mathbb{R}^d)$ consists of $\mathcal{C}^\infty$ functions, whose derivatives are all rapidly decaying. This property is sometimes called *rapidly decreasing*, see e.g. [FJ98, Definition 8.2.1]. A common example for a Schwartz function is $\mathrm{e}^{-\|\cdot\|^2}$.

Remark 3.4. Another characterization of a rapidly decaying function is given as follows. A function $f\colon \mathbb{R}^d \to \mathbb{C}$ is rapidly decaying on the cone $V \subseteq \mathbb{R}^d \setminus \{\mathbf{0}\}$ if and only if

$$\forall N \in \mathbb{N}_0\colon \quad |f(\boldsymbol{x})| = \mathcal{O}\left(\frac{1}{\|\boldsymbol{x}\|^N}\right) \qquad (\text{as } \|\boldsymbol{x}\| \to \infty \text{ with } \boldsymbol{x} \in V),$$

meaning $|f(\boldsymbol{x})|$ is decreasing faster on the cone V than any reciprocal power of $\|\boldsymbol{x}\|$ as $\|\boldsymbol{x}\|$ tends to infinity.

Proof. Without loss of generality assume $\|\boldsymbol{x}\| \geq 1$. Then for $k \in \mathbb{N}_0$ we get

$$\frac{(1+\|\boldsymbol{x}\|)^k}{\|\boldsymbol{x}\|^k} = \sum_{m=0}^{k} \binom{k}{m} \frac{\|\boldsymbol{x}\|^m}{\|\boldsymbol{x}\|^k} \leq \sum_{m=0}^{k} \binom{k}{m} =: \widetilde{C_k}.$$

Since $|f(\boldsymbol{x})| = \mathcal{O}\left(\frac{1}{\|\boldsymbol{x}\|^k}\right)$ we therefore have

$$|f(\boldsymbol{x})| \leq C\frac{1}{\|\boldsymbol{x}\|^k} \leq \underbrace{C\widetilde{C_k}}_{=:C_k}(1+\|\boldsymbol{x}\|)^{-k}$$

for some constant $C > 0$ and $\|\boldsymbol{x}\|$ large enough.

On the other hand, given $|f(\boldsymbol{x})| \leq C_k(1+\|\boldsymbol{x}\|)^{-k}$, we see that

$$|f(\boldsymbol{x})| \leq C_k\frac{1}{(1+\|\boldsymbol{x}\|)^k} \leq C_k\frac{1}{\|\boldsymbol{x}\|^k} = \mathcal{O}\left(\frac{1}{\|\boldsymbol{x}\|^k}\right).$$

□

Now we state the lemma that establishes the connection between the smoothness of a distribution and the rapid decay of its Fourier transform.

Lemma 3.5.
A distribution $u \in \mathcal{E}'(X)$ is in $\mathcal{D}(X)$ if and only if its Fourier transform is rapidly decaying on $\mathbb{R}^d$, meaning for each $N \in \mathbb{N}_0$ there is a constant $C_N > 0$ so that

$$|\mathcal{F}u(\boldsymbol{\xi})| \leq C_N(1+\|\boldsymbol{\xi}\|)^{-N}, \qquad \forall \boldsymbol{\xi} \in \mathbb{R}^d \qquad \text{as } \|\boldsymbol{\xi}\| \to \infty.$$

Proof. see [Pet83, Lemma 2.13.1] or [Rud73, Theorem 7.22]. □

Note that this is a global statement, it does not yet help us with finding the singular support of a distribution $u \in \mathcal{D}'(X)$. Therefore we utilize a cutoff function and note that for $\psi \in \mathcal{D}(X)$, we have $\psi u \in \mathcal{E}'(X)$. This enables us to employ the results of Lemma 3.5, which leads to the following corollary.

Corollary 3.6. (see [Pet83, Corollary 2.13.2])
Let $u \in \mathcal{D}'(X)$ and $U \subseteq X$ an open subset. Then $u|_U \in \mathcal{E}(U)$ if and only if the localized Fourier transform $\mathcal{F}\psi u$ is rapidly decaying on $\mathbb{R}^d$ for each $\psi \in \mathcal{D}(U)$, meaning for all $\psi \in \mathcal{D}(U)$ and $N \in \mathbb{N}_0$ there is a constant $C_{N,\psi} > 0$ such that

$$|(\mathcal{F}\psi u)(\boldsymbol{\xi})| \leq C_{N,\psi}(1 + \|\boldsymbol{\xi}\|)^{-N}, \qquad \forall \boldsymbol{\xi} \in \mathbb{R}^d \qquad \text{as } \|\boldsymbol{\xi}\| \to \infty.$$

Recall that the singular support consists of the points, where $u \in \mathcal{D}'(X)$ is not smooth. Lemma 3.5 and Corollary 3.6 imply that if u is not smooth, then there are nonzero directions $\boldsymbol{\xi}$ so that the inequalities for rapid decay do not hold. Motivated by this, we refine the concept of the singular support by including the directions in which $u \in \mathcal{D}'(X)$ is not smooth. Theorem 2.13.5 in [Pet83] allows us to restrict the set of functions ψ from Corollary 3.6 further, which now leads to our definition of the wavefront set.

Definition 3.7. ($\mathcal{C}^\infty$ wavefront set)
Let $u \in \mathcal{D}'(X)$. Then $(\boldsymbol{x_0}, \boldsymbol{\xi_0}) \in X \times (\mathbb{R}^d \backslash \{\mathbf{0}\})$ is *not* in the wavefront set of u, if there is a cutoff function $\psi \in \mathcal{D}(X)$ with $\psi(\boldsymbol{x_0}) \neq 0$, and a conic neighborhood $V \subseteq \mathbb{R}^d \backslash \{\mathbf{0}\}$ of $\boldsymbol{\xi_0}$, such that the localized Fourier transform $\mathcal{F}\psi u$ is rapidly decaying on V.

The $\mathcal{C}^\infty$ wavefront set of u will be denoted by $\mathrm{WF}(u)$.

Remark 3.8 (see [Hör03, Section 8.1])**.** Another way to define the wavefront set is through the frequency set $\Sigma(u)$. This is the cone of all $\boldsymbol{\xi} \in \mathbb{R}^d \backslash \{\mathbf{0}\}$, that do not have a conic neighborhood V of $\boldsymbol{\xi}$ such that $\mathcal{F}u$ is rapidly decaying on V. Then we can define the set of directions at which u is not smooth at $\boldsymbol{x} \in X$ as

$$\Sigma_{\boldsymbol{x}}(u) := \bigcap_{\psi \in \mathcal{D}(X),\ \psi(\boldsymbol{x}) \neq 0} \Sigma(\psi u).$$

The wavefront set is then given by

$$\mathrm{WF}(u) = \left\{ (\boldsymbol{x}, \boldsymbol{\xi}) \in X \times (\mathbb{R}^d \backslash \{\mathbf{0}\}) \colon \boldsymbol{\xi} \in \Sigma_{\boldsymbol{x}}(u) \right\}.$$

The following lemma will state some properties of the wavefront set.

Lemma 3.9.
Let $u \in \mathcal{D}'(X)$. Then the following holds

(a) $\mathrm{WF}(u)$ *is a conic set, meaning if* $(\boldsymbol{x}, \boldsymbol{\xi}) \in \mathrm{WF}(u)$, *then* $(\boldsymbol{x}, \lambda\boldsymbol{\xi}) \in \mathrm{WF}(u)$ *for all* $\lambda > 0$. *Further, it is a closed set in* $X \times (\mathbb{R}^d \backslash \{\mathbf{0}\})$.

(b) For $\phi \in \mathcal{E}(X)$ *we have* $\mathrm{WF}(\phi u) \subseteq \mathrm{WF}(u)$.

(c) $\mathrm{ssup}(u) = \pi(\mathrm{WF}(u))$, *where* π *is the projection of* $X \times (\mathbb{R}^d \backslash \{\mathbf{0}\})$ *onto* X.

Proof. See [Gra20, Lemma 2.13]. □

Example 3.10.
Consider the triangle

$$T = \left\{ \lambda_1(-1,0)^\top + \lambda_2(1,0)^\top + \lambda_3(0,1)^\top : \lambda_1, \lambda_2, \lambda_3 \geq 0,\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \right\}$$

in $\mathbb{R}^2$. Let u be the characteristic function

$$u(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in T, \\ 0, & \boldsymbol{x} \notin T. \end{cases}$$

Then $\mathrm{ssup}(u)$ is the boundary of the triangle, as it is not smooth there. The wavefront set $\mathrm{WF}(u)$ consists of the (nonzero) normal directions at all singular points except the three corners, at which all (nonzero) directions are contained in the wavefront set, see Figure 3.1.

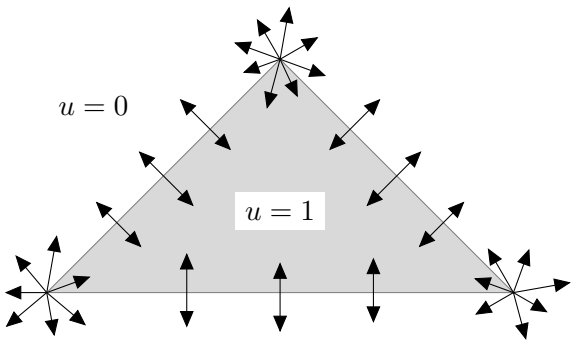


Fig. 3.1: Illustration of the wavefront set of the characteristic function u on a triangle.

Proof (inspired by [KQ15, Example 5]; with input from Eric Todd Quinto). Let $\psi \in \mathcal{D}(\mathbb{R})$ be a cutoff function with $\psi(0) \neq 0$. Then, by repeated integration by parts, we have for the localized Fourier transform of the Heaviside function

$$\begin{aligned}
\sqrt{2\pi}(\mathcal{F}\psi H)(\xi) &= \int_0^\infty e^{-\mathrm{i}x\xi}\psi(x)\,\mathrm{d}x = \left[\frac{e^{-\mathrm{i}x\xi}}{-\mathrm{i}\xi}\psi(x)\right]_{x=0}^{\infty} - \int_0^\infty \frac{e^{-\mathrm{i}x\xi}}{-\mathrm{i}\xi}\psi'(x)\,\mathrm{d}x \\
&= -\frac{1}{\mathrm{i}\xi}\left(\psi(0) - \int_0^\infty e^{-\mathrm{i}x\xi}\psi'(x)\,\mathrm{d}x\right) \\
&= -\frac{1}{\mathrm{i}\xi}\left(\psi(0) - \left[\frac{e^{-\mathrm{i}x\xi}}{-\mathrm{i}\xi}\psi'(x)\right]_{x=0}^{\infty} + \int_0^\infty \frac{e^{-\mathrm{i}x\xi}}{-\mathrm{i}\xi}\psi''(x)\,\mathrm{d}x\right) \\
&= -\frac{1}{\mathrm{i}\xi}\left(\psi(0) + \frac{1}{\mathrm{i}\xi}\left(\psi'(0) - \int_0^\infty e^{-\mathrm{i}x\xi}\psi''(x)\,\mathrm{d}x\right)\right) \\
&= \mathcal{O}\left(\frac{1}{|\xi|}\right) + \mathcal{O}\left(\frac{1}{|\xi|^2}\right) + \cdots = \mathcal{O}\left(\frac{1}{|\xi|}\right),
\end{aligned}$$

where the last equality is to be understood in terms of the absolute value of the previous terms. Also note that the first term is (absolutely) bounded at $\xi = 0$.

Let $\boldsymbol{x^0} \in E \coloneqq \left\{(\lambda - 1, \lambda)^\top : \lambda \in (0,1)\right\}$ be a point on the upper left edge of the triangle. Fix a direction $\boldsymbol{\xi^0} = (\xi_1^0, \xi_2^0)^\top$ with $\xi_1^0 \neq -\xi_2^0$. We now show that the localized Fourier transform is rapidly decaying in a conic neighborhood Γ of $\boldsymbol{\xi^0}$. We can choose a constant $C > 0$ so that the cone $\Gamma \coloneqq \{|\xi_1 + \xi_2| \geq C\,\|\boldsymbol{\xi}\|\}$ includes $\boldsymbol{\xi^0}$. This setup is also illustrated in Figure 3.2.

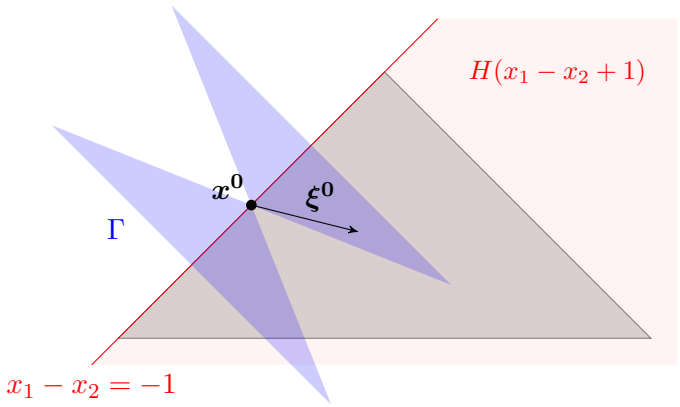


Fig. 3.2: Illustration of the situation for the upper left corner of the triangle.

Consider $f\colon (x_1, x_2)^\top \mapsto \psi(x_1 - x_2 + 1)\psi(x_2 - x_2^0)H(x_1 - x_2 + 1)$. Here, we need to choose $\operatorname{supp}(\psi(x_2 - x_2^0))$ small enough that f is supported away from any other boundary of

the triangle. Therefore assume $\operatorname{supp}(\psi(x_2 - x_2^0)) \subset (0,1)$. Then for the Fourier transform of f, we get

$$\begin{aligned}
2\pi\,(\mathcal{F}f)\,(\boldsymbol{\xi}) &= \int_{\mathbb{R}^2} e^{-\mathrm{i}(x_1\xi_1+x_2\xi_2)}\psi(x_1-x_2+1)\psi(x_2-x_2^0)H(x_1-x_2+1)\,\mathrm{d}\boldsymbol{x}\\
&= \int_{\mathbb{R}} e^{-\mathrm{i}x_2\xi_2}\psi(x_2-x_2^0)\int_{\mathbb{R}} e^{-\mathrm{i}x_1\xi_1}\psi(x_1-x_2+1)H(x_1-x_2+1)\,\mathrm{d}x_1\,\mathrm{d}x_2\\
&= \int_{\mathbb{R}} e^{-\mathrm{i}x_2\xi_2}\psi(x_2-x_2^0)\int_{\mathbb{R}} e^{-\mathrm{i}(z+x_2-1)\xi_1}\psi(z)H(z)\,\mathrm{d}z\,\mathrm{d}x_2\\
&= e^{\mathrm{i}\xi_1}\underbrace{\int_0^\infty e^{-\mathrm{i}z\xi_1}\psi(z)\,\mathrm{d}z}_{=\mathcal{O}\left(\frac{1}{|\xi_1|}\right)}\underbrace{\int_{\mathbb{R}} e^{-\mathrm{i}x_2(\xi_1+\xi_2)}\psi(x_2-x_2^0)\,\mathrm{d}x_2}_{=:R},
\end{aligned}$$

where the equality under the first underbrace is again to be understood in terms of the absolute value.

Now we use partial integration for the R term and obtain

$$\begin{aligned}
R &= \left[\frac{e^{-\mathrm{i}x_2(\xi_1+\xi_2)}}{-\mathrm{i}(\xi_1+\xi_2)}\psi(x_2-x_2^0)\right]_{x_2=-\infty}^{\infty} - \int_{\mathbb{R}}\frac{e^{-\mathrm{i}x_2(\xi_1+\xi_2)}}{-\mathrm{i}(\xi_1+\xi_2)}\psi'(x_2-x_2^0)\,\mathrm{d}x_2\\
&= \frac{1}{\mathrm{i}(\xi_1+\xi_2)}\int_{\mathbb{R}} e^{-\mathrm{i}x_2(\xi_1+\xi_2)}\psi'(x-x_2^0)\,\mathrm{d}x_2.
\end{aligned}$$

By repeating the partial integration N times, we see that $|R|$ is $\mathcal{O}\left(\frac{1}{|\xi_1+\xi_2|^N}\right)$ for all $N \in \mathbb{N}_0$. Therefore $|(\mathcal{F}f)\,(\boldsymbol{\xi})| = \mathcal{O}\left(\frac{1}{|\xi_1||\xi_1+\xi_2|^N}\right)$, and $\mathcal{F}f$ is rapidly decaying on Γ as $|\xi_1+\xi_2| \geq C\,\|\boldsymbol{\xi}\|$ for $\boldsymbol{\xi} \in \Gamma$. On the other hand, if $\xi_1 = -\xi_2$, then $(\mathcal{F}f)\,(\boldsymbol{\xi})$ is *not* rapidly decaying, meaning the only directions in $\mathrm{WF}(u)$ at $\boldsymbol{x^0}$ are those with $\xi_1 = -\xi_2$.

In the second part of this proof we show that all possible directions $\boldsymbol{\xi} \in \mathbb{R}^2\setminus\{\mathbf{0}\}$ are in the wavefront set $\mathrm{WF}(u)$ at the lower left corner $(-1,0)^\top$. Once more, we define a function $f\colon (x_1,x_2)^\top \mapsto \psi(x_1-x_2+1)\psi(x_2)H(x_1-x_2+1)H(x_2)$, which describes the localized intersection of the half-spaces $x_1 - x_2 \geq -1$ and $x_2 \geq 0$. Its Fourier transform is

$$\begin{aligned}
2\pi(\mathcal{F}f)(\boldsymbol{\xi}) &= \int_{\mathbb{R}^2} e^{-\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}}\psi(x_1-x_2+1)\psi(x_2)H(x_1-x_2+1)H(x_2)\,\mathrm{d}\boldsymbol{x}\\
&= \int_0^\infty e^{-\mathrm{i}x_2\xi_2}\psi(x_2)\int_{\mathbb{R}} e^{-\mathrm{i}(z+x_2-1)\xi_1}\psi(z)H(z)\,\mathrm{d}z\,\mathrm{d}x_2\\
&= e^{\mathrm{i}\xi_1}\int_0^\infty e^{-\mathrm{i}x_2(\xi_1+\xi_2)}\psi(x_2)\,\mathrm{d}x_2\int_0^\infty e^{-\mathrm{i}z\xi_1}\psi(z)\,\mathrm{d}z
\end{aligned}$$

Therefore $|(\mathcal{F}f)(\boldsymbol{\xi})|$ is $\mathcal{O}\left(\frac{1}{|\xi_1+\xi_2||\xi_1|}\right)$, and $\mathcal{F}f$ is *not* rapidly decaying in any direction $\boldsymbol{\xi} \in \mathbb{R}^2\setminus\{\mathbf{0}\}$.

The proof for the two other edges and corners can be done in a similar way. □

In general, for a set $\Omega \subset \mathbb{R}^d$ with a $\mathcal{C}^\infty$ boundary, we get

$$\mathrm{WF}(\chi_\Omega) = \left\{(\boldsymbol{x}, \boldsymbol{\xi}) \in \mathbb{R}^d \times (\mathbb{R}^d \backslash \{\mathbf{0}\}) \colon \boldsymbol{x} \in \partial\Omega,\ \boldsymbol{\xi} \perp \partial\Omega \text{ at } \boldsymbol{x}\right\},$$

see [Hör03, Example 8.2.5] or [KQ15, Example 6].

3.2 Fourier Integral Operators

The essential terms in this section are often defined on open sets $X \subseteq \mathbb{R}^{n_x}$ and $Y \subseteq \mathbb{R}^{n_y}$ for different $n_x, n_y \in \mathbb{N}$, but we do not need this generality here. Instead, we let X and Y both be open subsets of $\mathbb{R}^d$. Further, let $N \in \mathbb{N}$. We start with the two core components of Fourier integral operators.

Definition 3.11. (Symbol)
A function $p \in \mathcal{E}(Y \times X \times \mathbb{R}^N)$ is a *symbol of order* $m \in \mathbb{R}$ if for every compact set $K \subseteq Y \times X$ and all multi-indices $\boldsymbol{\alpha} \in \mathbb{N}_0^N$, $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}_0^d$ there is a constant $C_{K,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} > 0$ so that

$$\left|D_{\boldsymbol{\xi}}^{\boldsymbol{\alpha}} D_{\boldsymbol{x}}^{\boldsymbol{\beta}} D_{\boldsymbol{y}}^{\boldsymbol{\gamma}}\, p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi})\right| \leq C_{K,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} (1 + \|\boldsymbol{\xi}\|)^{m - \|\boldsymbol{\alpha}\|}, \qquad \forall (\boldsymbol{y}, \boldsymbol{x}) \in K,\ \boldsymbol{\xi} \in \mathbb{R}^N. \tag{3.2.1}$$

The set of all symbols of order $m \in \mathbb{R}$ will be denoted by $S^m(Y \times X \times \mathbb{R}^N)$. Further, we define $S^{-\infty}(Y \times X \times \mathbb{R}^N) \coloneqq \bigcap_{m \in \mathbb{R}} S^m(Y \times X \times \mathbb{R}^N)$.

Later on, we will also consider symbols depending only on two variables. In this case, (3.2.1) holds without the derivative in $\boldsymbol{y}$, and we write $S^m(X \times \mathbb{R}^N)$ or, in case of $N = d$, simply $S^m(X)$ for symbols of order m on $X \times \mathbb{R}^N$.

Further, a symbol $p \in S^m(Y \times X \times \mathbb{R}^N)$ is *elliptic* (of order m) if for every compact set $K \subseteq Y \times X$ there are constants $C_K > 0$ and $M > 0$ such that

$$|p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi})| \geq C_K (1 + \|\boldsymbol{\xi}\|)^m, \qquad \forall (\boldsymbol{y}, \boldsymbol{x}) \in K,\ \|\boldsymbol{\xi}\| \geq M. \tag{3.2.2}$$

Sometimes in the literature, the condition (3.2.1) is only required to hold for $\|\boldsymbol{\xi}\| \geq 1$, and an integrability condition of p on $K \times \left\{\boldsymbol{\xi} \in \mathbb{R}^N \colon \|\boldsymbol{\xi}\| \leq 1\right\}$ is added, see for example [Gra20, Section 2.1.1]. However, if p fulfills estimate (3.2.1), then it is already integrable.

Definition 3.12. (Phase function)
A real valued function $\phi \in \mathcal{E}(Y \times X \times (\mathbb{R}^N \setminus \{\mathbf{0}\}))$ is called a *phase function* if

- ϕ is positive homogeneous of degree 1 in $\boldsymbol{\xi}$, meaning for $\lambda > 0$ we have

$$\phi(\boldsymbol{y}, \boldsymbol{x}, \lambda\boldsymbol{\xi}) = \lambda\, \phi(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) \qquad \text{for all} \quad (\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) \in Y \times X \times (\mathbb{R}^N \setminus \{\mathbf{0}\})$$

- $(\nabla_{\boldsymbol{y}}\phi^\top, \nabla_{\boldsymbol{\xi}}\phi^\top)$ and $(\nabla_{\boldsymbol{x}}\phi^\top, \nabla_{\boldsymbol{\xi}}\phi^\top)$ do not vanish on $Y \times X \times (\mathbb{R}^N \setminus \{\mathbf{0}\})$.

A phase function ϕ is called *non-degenerate* if the set

$$\left\{ \left(\nabla_{(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\xi})} \partial_{\xi_j} \phi \right)^\top \; : j = 1, \dots, N \right\}$$

is linearly independent on

$$\Sigma_\phi \coloneqq \left\{ (\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) \in Y \times X \times (\mathbb{R}^N \setminus \{\mathbf{0}\}) : \nabla_{\boldsymbol{\xi}} \phi(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) = \mathbf{0} \right\}.$$

Now we can define Fourier integral operators.

Definition 3.13. (Fourier integral operator)
Let $p \in S^m(Y \times X \times \mathbb{R}^N)$ be a symbol of order m and $\phi \in \mathcal{E}(Y \times X \times \mathbb{R}^N \setminus \{\mathbf{0}\})$ a non-degenerate phase function. A *Fourier integral operator* (FIO) applied to $u \in \mathcal{D}(X)$ is given by

$$Fu(\boldsymbol{y}) = \int_{\mathbb{R}^N} \int_X \mathrm{e}^{\mathrm{i}\phi(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\xi})} p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{\xi}, \qquad \boldsymbol{y} \in Y.$$

The *order* of the FIO F is $k \coloneqq m - \frac{d-N}{2}$.

This operator F maps $\mathcal{D}(X)$ continuously into $\mathcal{E}(Y)$, which can be uniquely extended as a continuous map from $\mathcal{E}'(X)$ to $\mathcal{D}'(Y)$, see [Trè81, Theorem VIII.5.1] or [Hör71, Section 1.4].

Definition 3.14. (Canonical relation)
Let F be a FIO with phase function $\phi \in \mathcal{E}(Y \times X \times \mathbb{R}^N \setminus \{\mathbf{0}\})$. Then the *canonical relation* $C \subseteq (Y \times \mathbb{R}^d \setminus \{\mathbf{0}\}) \times (X \times \mathbb{R}^d \setminus \{\mathbf{0}\})$ is defined by

$$C \coloneqq \{(\boldsymbol{y}, \nabla_{\boldsymbol{y}}\phi(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi});\ \boldsymbol{x}, -\nabla_{\boldsymbol{x}}\phi(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi})) :\ (\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) \in \Sigma_\phi\}.$$

For canonical relations C and $\widetilde{C}$, and $u \in \mathcal{D}'(X)$, we define

$$C \circ \mathrm{WF}(u) \coloneqq \{(\boldsymbol{y}, \boldsymbol{\eta}) \colon \exists (\boldsymbol{x}, \boldsymbol{\xi}) \in \mathrm{WF}(u) \colon (\boldsymbol{y}, \boldsymbol{\eta}; \boldsymbol{x}, \boldsymbol{\xi}) \in C\},$$

and

$$\widetilde{C} \circ C \coloneqq \left\{(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{\eta}}, \boldsymbol{x}, \boldsymbol{\xi}) \colon \exists (\boldsymbol{y}, \boldsymbol{\eta}) \text{ with } (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{\eta}}; \boldsymbol{y}, \boldsymbol{\eta}) \in \widetilde{C} \text{ and } (\boldsymbol{y}, \boldsymbol{\eta}; \boldsymbol{x}, \boldsymbol{\xi}) \in C\right\}.$$

The following theorem describes how a FIO propagates singularities.

Theorem 3.15. (Hörmander-Sato Lemma)
Let $u \in \mathcal{E}'(X)$ and F, $\widetilde{F}$ be FIOs with canonical relations C and $\widetilde{C}$ respectively. Assume the composition $F\widetilde{F}$ is well defined on $\mathcal{E}'(X)$. Then we have

$$\mathrm{WF}(Fu) \subseteq C \circ \mathrm{WF}(u),$$
$$\mathrm{WF}((F\widetilde{F})u) \subseteq (C \circ \widetilde{C}) \circ \mathrm{WF}(u).$$

Proof. The assertions follow by results in [Hör03], namely Theorem 8.2.12. They are stated in this form in Theorem 15 and Theorem 16 in [KQ15]. □

Theorem 3.15 provides us with a superset of singularities that can be preserved when a FIO is applied to a distribution. In particular, this means that potentially some of these singularities are not visible when a FIO is applied. See [KQR23, Proposition 3.5] for a characterization of visible and invisible singularities. Furthermore, because of the composition with the canonical relation C, a FIO can even add singularities that were not originally present in u. Microlocal analysis can help to understand which singularities will be visible and what types of those *artifacts* are introduced, see, e.g. [FQ15].

Next, we look into a special case of Fourier integral operators which do not have the problem of potentially adding artifacts.

3.3 Pseudodifferential operators

Here, we consider FIOs with $X = Y$, $N = d$, and the non-degenerate phase function $\phi(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi}) = (\boldsymbol{y} - \boldsymbol{x}) \cdot \boldsymbol{\xi}$. Further, the pseudodifferential symbol will only depend on the two variables $\boldsymbol{x}$ and $\boldsymbol{\xi}$. We first generalize the definition of a symbol by allowing functions which are not necessarily $\mathcal{C}^\infty$ in $\boldsymbol{\xi} = \mathbf{0}$.

Definition 3.16. (General symbol)
A function $p \in \mathcal{E}(X \times (\mathbb{R}^d \backslash \{\mathbf{0}\}))$ is a *general symbol of order* $m \in \mathbb{R}$ if

1. for every compact set $K \subseteq X$ and all multi-indices $\boldsymbol{\alpha}, \boldsymbol{\beta}, \in \mathbb{N}_0^d$ there is a constant $C_{K,\boldsymbol{\alpha},\boldsymbol{\beta}} > 0$ so that

$$\left| D_{\boldsymbol{\xi}}^{\boldsymbol{\alpha}} D_{\boldsymbol{x}}^{\boldsymbol{\beta}} p(\boldsymbol{x}, \boldsymbol{\xi}) \right| \leq C_{K,\boldsymbol{\alpha},\boldsymbol{\beta}} (1 + \|\boldsymbol{\xi}\|)^{m - \|\boldsymbol{\alpha}\|}, \qquad \forall \boldsymbol{x} \in K,\ \|\boldsymbol{\xi}\| \geq 1.$$

2. for all multi-indices $\boldsymbol{\beta} \in \mathbb{N}_0^d$, $D_{\boldsymbol{x}}^{\boldsymbol{\beta}} p(\boldsymbol{x}, \boldsymbol{\xi})$ is integrable on $K \times \left\{ \boldsymbol{\xi} \in \mathbb{R}^d \colon\ \|\boldsymbol{\xi}\| \leq 1 \right\}$.

This is a generalization of Definition 3.11. The set of all general symbols of order $m \in \mathbb{R}$ will be denoted by $S_g^m(X)$.

Definition 3.17. (Pseudodifferential operator and general pseudodifferential operator)
Let $p \in S^m(X)$ be a symbol of order $m \in \mathbb{R}$ on $X \times \mathbb{R}^d$. For $u \in \mathcal{D}(X)$, a *pseudodifferential operator* (ΨDO) of order m has the following form

$$Pu(\boldsymbol{x}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \int_X \mathrm{e}^{\mathrm{i}(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{\xi}} p(\boldsymbol{x}, \boldsymbol{\xi}) u(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\boldsymbol{\xi} = \frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} p(\boldsymbol{x}, \boldsymbol{\xi}) \widehat{u}(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}.$$

If instead $p \in S_g^m(X)$ is a general symbol of order $m \in \mathbb{R}$, we will call P a *general pseudodifferential operator* of order m.

Again, P maps $\mathcal{D}(X)$ continuously into $\mathcal{E}(X)$, see [Pet83, Chapter 3, Theorem 2.4]. This operator can also be uniquely extended to a continuous linear map from $\mathcal{E}'(X)$ into $\mathcal{D}'(X)$, see [Pet83, Chapter 3, Corollary 3.13].

A *smoothing operator* maps $\mathcal{E}'(X)$ continuously into $\mathcal{E}(X)$. Now, a ΨDO with symbol $p \in S^{-\infty}(X)$ is a smoothing operator, see [Pet83, Chapter 3, Lemma 3.14]. For the study of singularities, those smoothing operators can be neglected. Note that if P is a general ΨDO, then P differs from a standard ΨDO by a smoothing operator, see [GQR24, text below Definition 2.10].

Note that the obvious choice would have been to use a symbol depending on all three variables as in the last section for this definition, matching the idea that ΨDOs are FIOs with a specific phase function. It can be shown, however, that operators in the form of Definition 3.17 generate the same class of operators modulo smoothing operators as those with symbol $p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\xi})$, see [Pet83, p.188] or [Hör71, Chapter 2.1].

Example 3.18.
Consider the operator $-\Delta u(\boldsymbol{x}) = -\sum_{k=1}^{d} D^{2\boldsymbol{e_k}} u(\boldsymbol{x})$ on $\mathbb{R}^d$, $u \in \mathcal{D}'(\mathbb{R}^d)$. In terms of a Fourier representation, we get

$$\begin{aligned}
-\Delta u(\boldsymbol{x}) &= \frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \widehat{-\Delta u}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \int_{\mathbb{R}^d} \mathrm{e}^{-\mathrm{i}\boldsymbol{y}\cdot\boldsymbol{\xi}} (-\Delta u)(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{y} \,\mathrm{d}\boldsymbol{\xi} \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left(-\sum_{k=1}^{d} (\mathrm{i}\xi_k)^2 \right) \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{\xi}} u(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{y} \,\mathrm{d}\boldsymbol{\xi} \qquad (p.i.) \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{\xi}} (-\mathrm{i}^2 \|\boldsymbol{\xi}\|^2) u(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{y} \,\mathrm{d}\boldsymbol{\xi}.
\end{aligned}$$

The symbol of $-\Delta$ is therefore given by $p(\boldsymbol{x}, \boldsymbol{\xi}) = -\mathrm{i}^2 \|\boldsymbol{\xi}\|^2 = \|\boldsymbol{\xi}\|^2 \in S^2(\mathbb{R}^d)$. As the symbol further fulfills inequality (3.2.2), $-\Delta$ is an elliptic pseudodifferential operator of order 2.

Example 3.19.
Similar to the first example, we see that

$$(-\Delta)^{1/2} u(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \|\boldsymbol{\xi}\| \, \widehat{u}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}.$$

As $\boldsymbol{\xi} \mapsto \|\boldsymbol{\xi}\|$ is not $\mathcal{C}^\infty$ at $\mathbf{0}$, this is not a symbol as per Definition 3.11 and therefore $(-\Delta)^{1/2}$ is not a pseudodifferential operator. It is, however, a general symbol, see Definition 3.16.

Given a cutoff function $\psi \in \mathcal{D}(\mathbb{R}^d)$ with $\psi(\mathbf{0}) = 1$, we can split the integral into

$$(-\Delta)^{1/2} u(\boldsymbol{x}) = \underbrace{\frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \psi(\boldsymbol{\xi}) \|\boldsymbol{\xi}\| \, \widehat{u}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}}_{:= Lu(\boldsymbol{x})} + \underbrace{\frac{1}{\sqrt{2\pi}^d} \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} (1 - \psi(\boldsymbol{\xi})) \|\boldsymbol{\xi}\| \, \widehat{u}(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}}_{:= Ru(\boldsymbol{x})}.$$

Now, Ru is a ΨDO of order 1 with symbol $(1 - \psi(\boldsymbol{\xi})) \|\boldsymbol{\xi}\|$. Further, $\psi(\boldsymbol{\xi}) \|\boldsymbol{\xi}\| \, \widehat{u}(\boldsymbol{\xi}) \in \mathcal{E}'(\mathbb{R}^d)$ and $Lu \in \mathcal{E}(\mathbb{R}^d)$ according to [Pet83, Chapter 2, Theorem 8.1], meaning L is a smoothing operator.

With the use of pseudodifferential operators, we can predict how the singular support and wavefront set of $u \in \mathcal{E}'(X)$ changes when such an operator is applied to u. This is stated in the following theorem.

Theorem 3.20. (Pseudolocal property)
Let P be a ΨDO and $u \in \mathcal{E}'(X)$. Then P satisfies the pseudolocal property (see [Trè80, Chapter I, Corollary 6.2]):

$$\operatorname{ssup}(Pu) \subseteq \operatorname{ssup}(u) \qquad \textit{and} \qquad \operatorname{WF}(Pu) \subseteq \operatorname{WF}(u).$$

If P is elliptic, meaning its symbol is elliptic, then the singular support and wavefront set is preserved (see [Pet83, Chapter 4 §1.])

$$\operatorname{ssup}(Pu) = \operatorname{ssup}(u) \qquad \textit{and} \qquad \operatorname{WF}(Pu) = \operatorname{WF}(u).$$

This especially means that a ΨDO does not add any artifacts.

A part of the symbol of a pseudodifferential operator plays a crucial role in analyzing its properties.

Definition 3.21. (Top order symbol)
Let P be a ΨDO of order $m \in \mathbb{R}$ and $p \in S^m(X)$ its corresponding symbol. A function $\sigma(P) \in S^m(X)$ is the *top order or principal symbol* of P if

$$p - \sigma(P) \in S^{m-\varepsilon}(X), \quad \text{for some } \varepsilon > 0,$$

and $\sigma(P)$ is positively homogeneous of degree m in the last variable.

Besides the obvious fact that the order of a ΨDO coincides with the order of the top order symbol, there are also other useful properties that transfer over. For instance, ellipticity of a pseudodifferential operator, and therefore of its symbol, is a property shared with the corresponding top order symbol (see [Gra20, p. 26]).

CHAPTER 4

Imaging operators

In this chapter we specify the different imaging operators we use by stating the pseudodifferential operators K or M and weight W. For each operator, we calculate explicit expressions for the term $K^\star e_\gamma^k$, which is an integral part of the reconstruction kernel, see equation (2.2.20). The mollifier function e_γ^k we rely on is also declared here.

Let $S \subset \mathbb{R}$ be an open interval in $\mathbb{R}$ containing the parameters for source and receiver positions as in Section 2.2. Recall φ from equation (2.2.10) and define

$$\mathcal{X} \coloneqq \operatorname{int}\left\{\boldsymbol{x} \in \mathbb{R}^2_+ \colon \forall s \in S \colon \nabla_{\boldsymbol{x}} \varphi(s, \boldsymbol{x}) \neq \mathbf{0}\right\} \qquad \text{and} \quad Y \coloneqq \{(s,t) \colon s \in S, t > t_{\text{first}}(s)\},$$

where $t_{\text{first}}(s)$ is the first arrival time of the unique ray traveling from source $\boldsymbol{x_s}(s)$ to receiver $\boldsymbol{x_r}(s)$.

First, we show that the operator $F \colon \mathcal{E}'(\mathcal{X}) \to \mathcal{D}'(Y)$ from (2.2.12) is in fact a Fourier integral operator of order $-1/2$. By using the Fourier representation of the δ function, we obtain

$$\begin{aligned}
Fn(s,t) &= \int_{\mathcal{X}} n(\boldsymbol{x}) A(s,\boldsymbol{x}) \left(\mathcal{F}^{-1}\mathcal{F}\left[\eta \mapsto \delta(\eta - \varphi(s,\boldsymbol{x}))\right]\right)(t)\,\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} n(\boldsymbol{x}) A(s,\boldsymbol{x}) \left(\mathcal{F}^{-1}\left[\xi \mapsto \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\mathrm{i}\xi\varphi(s,\boldsymbol{x})}\right]\right)(t)\,\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} n(\boldsymbol{x}) A(s,\boldsymbol{x}) \frac{1}{2\pi} \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}\xi(t-\varphi(s,\boldsymbol{x}))}\,\mathrm{d}\xi\,\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathbb{R}} \int_{\mathcal{X}} \mathrm{e}^{\mathrm{i}\xi(t-\varphi(s,\boldsymbol{x}))} \frac{1}{2\pi} A(s,\boldsymbol{x}) n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\xi,
\end{aligned}$$

Note that a solution to the the eikonal equation (2.2.7) can be obtained by solving systems of ordinary differential equations that depend on a further parameter. This has been done, for instance, in [BCS01, Chapter E.2] and [KQR23, Section 4.1]. We assume c is bounded away from 0 and $\nabla_{\boldsymbol{x}} c$ is bounded, which are both reasonable assumptions for the background velocity. Then, under the geometric optics assumption (recall Section 2.2) and since $c \in \mathcal{E}(\mathcal{X})$, it can be shown using theory for ordinary differential equations, see e.g. [Har02, Chapter 5, Section 4], that $\tau(\cdot, \boldsymbol{x_s}) \in \mathcal{E}(\mathcal{X})$. Similar steps can be applied to the transport equation (2.2.8), see [BCS01, Chapter E.3] and [KQR23, Appendix A], to then show $a(\cdot, \boldsymbol{x_s}) \in \mathcal{E}(\mathcal{X})$.

Since $p(s,t,\boldsymbol{x},\xi) \coloneqq \frac{1}{2\pi}A(s,\boldsymbol{x})$ is independent of ξ, it is a symbol of order 0. Further set $\phi(s,t,\boldsymbol{x},\xi) \coloneqq \xi(t-\varphi(s,\boldsymbol{x}))$. We have $\phi(s,t,\boldsymbol{x},\lambda\xi) = \lambda\phi(s,t,\boldsymbol{x},\xi)$ for $\lambda > 0$, meaning ϕ is positive homogeneous of degree 1 in ξ. Consider

$$\begin{aligned}\nabla_{(s,t)}\phi(s,t,\boldsymbol{x},\xi) &= (-\xi\partial_s\varphi(s,\boldsymbol{x}),\ \xi)^\top,\\ \nabla_{\boldsymbol{x}}\phi(s,t,\boldsymbol{x},\xi) &= -\xi\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x}),\\ \partial_\xi\phi(s,t,\boldsymbol{x},\xi) &= t-\varphi(s,\boldsymbol{x}).\end{aligned}$$

As $\xi \neq 0$, and $\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x}) \neq \mathbf{0}$ by choice of $\mathcal{X}$, neither $(\nabla_{(s,t)}\phi^\top, \partial_\xi\phi)$, nor $(\nabla_{\boldsymbol{x}}\phi^\top, \partial_\xi\phi)$ vanish on $Y\times\mathcal{X}\times\mathbb{R}\setminus\{\mathbf{0}\}$. Therefore, ϕ is a phase function. We can see that the matrix

$$\begin{aligned}&\begin{pmatrix}\nabla_{(s,t)}\partial_\xi\phi(s,t,\boldsymbol{x},\xi)^\top & \nabla_{\boldsymbol{x}}\partial_\xi\phi(s,t,\boldsymbol{x},\xi)^\top & \partial_\xi^2\phi(s,t,\boldsymbol{x},\xi)^\top\end{pmatrix}\\ =\ &\begin{pmatrix}-\partial_s\varphi(s,\boldsymbol{x}) & 1 & -\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})^\top & 0\end{pmatrix}\end{aligned}$$

has full rank 1, meaning ϕ is a non-degenerate. This concludes that F is a FIO of order $0-\frac{2-1}{2} = -\frac{1}{2}$.

Similarily for the generalized backprojection $F_W^\dagger\colon \mathcal{E}'(Y)\to\mathcal{D}'(\mathcal{X})$, we get

$$\begin{aligned}F_W^\dagger g(\boldsymbol{x}) &= \int_Y W(s,\boldsymbol{x})g(s,t)\frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}\xi(t-\varphi(s,\boldsymbol{x}))}\,\mathrm{d}\xi\,\mathrm{d}(s,t)\\ &= \int_{\mathbb{R}}\int_Y \mathrm{e}^{\mathrm{i}\xi(t-\varphi(s,\boldsymbol{x}))}\frac{1}{2\pi}W(s,\boldsymbol{x})g(s,t)\,\mathrm{d}(s,t)\,\mathrm{d}\xi.\end{aligned}$$

using the same arguments from above, we conclude that $F_W^\dagger$ is also a FIO of order $-1/2$. Note that $F\colon\mathcal{E}'(X)\to\mathcal{D}'(Y)$ and $F_W^\dagger\colon\mathcal{E}'(Y)\to\mathcal{D}'(X)$ cannot be composed in general, which is why the cutoff function $\psi\in\mathcal{D}(S)$ is introduced in (2.2.14). With this cutoff function, we get $\psi Fn \in \mathcal{E}'(Y)$, making the composition of $F_W^\dagger$ and ψF possible, see [GQR24, Proof of Theorem 3.2]. For the symbol analysis of the reconstruction operators (2.2.14) in this general setting, we refer to [GQR24, Section 3.1].

For the family of mollifiers we will use the following function, given $\gamma > 0$

$$e_\gamma^k(\boldsymbol{x}) = C_{k,\gamma}\begin{cases}(\gamma^2-\|\boldsymbol{x}\|^2)^k, & \|\boldsymbol{x}\|<\gamma,\\ 0, & \|\boldsymbol{x}\|\geq\gamma,\end{cases} \tag{4.0.1}$$

with a design parameter $k > 2$ that controls the smoothness of e_γ^k, and normalization factor

$$C_{k,\gamma} = \left(\int_{B_\gamma(\mathbf{0})}(\gamma^2-\|\boldsymbol{x}\|^2)^k\,\mathrm{d}\boldsymbol{x}\right)^{-1} = \left(2\pi\int_0^\gamma(\gamma^2-r^2)^k r\,\mathrm{d}r\right)^{-1} = \frac{k+1}{\pi\gamma^{2(k+1)}}.$$

If not stated otherwise, we use $k = 3$. Note that the function (4.0.1) is in fact a mollifier according to (2.2.15), and we have $e_\gamma^k(\boldsymbol{x}) = \frac{1}{\gamma^2} e_1^k(\frac{\boldsymbol{x}}{\gamma})$.

Remark 4.1. The parameter k must be so large that the expression $K^* e_\gamma^k$ is continuous. Since the operator K^*, which we will specify later on, differentiates e_γ^k up to two times, k must be strictly greater than 2 in our case.

Set $X \coloneqq \{\boldsymbol{x} \in \mathbb{R}^2_+ \colon x_2 > x_{\min}\}$, where $x_{\min}$ is the deepest/largest x_2-coordinate of any t_{first} isochrone, $x_{\min} \coloneqq \max\left\{x_2 \colon (x_1, x_2)^\top \in \mathcal{L}_{(s,t_{\text{first}}(s))}, s \in S\right\}$. The quantity $x_{\min}$ depends on the common offset α and background velocity c. For instance, given the affine linear velocity model $c(\boldsymbol{x}) = b + mx_2$, $m, b > 0$, we have (see [KQR23, equation (3.4)])

$$x_{\min} = \frac{b}{m}\left(\sqrt{1 + \frac{m^2\alpha^2}{b^2}} - 1\right), \qquad t_{\text{first}}(s) = t_{\text{first}} = \frac{2}{m}\operatorname{asinh}\left(\frac{m\alpha}{b}\right),$$

see Figure 4.1 for an illustration. In case of a constant velocity $c = b$, $b > 0$, we have

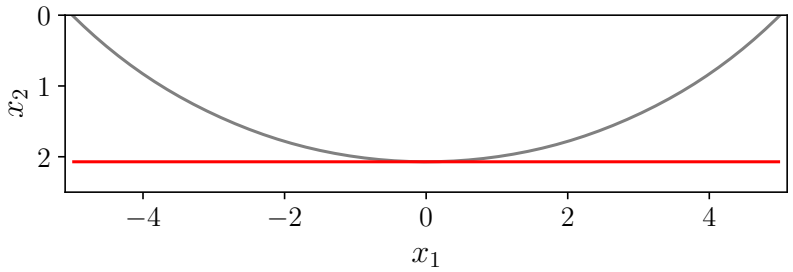


Fig. 4.1: First arrival reflection isochrone $\mathcal{L}_{(0,t_{\text{first}})}$ (colored gray) for the linear velocity model $c(\boldsymbol{x}) = 0.5 + 0.1x_2$ and $\alpha = 5$. The red line is placed at height of the corresponding value $x_{\min}$.

$x_{\min} = 0$ and $t_{\text{first}} = 2\alpha$, see e.g. [Gra+18]. Note that $X \subseteq \mathcal{X}$, see [KQR23, Remark 3.2]. Recall again the considered types of reconstruction operators (2.2.14)

$$\frac{1}{2\pi} K F_W^\dagger \psi F \quad \text{and} \quad \frac{1}{2\pi} F_W^\dagger M \psi F, \qquad \Lambda = \frac{1}{2\pi} K F_W^\dagger M \psi F.$$

Let M have the form $\mathrm{Id} \otimes L$ with a general ΨDO L, where $\otimes$ denotes the tensor product (L operates on the second/time variable). Also, let K be a properly supported general ΨDO, and its general symbol be bounded at $\boldsymbol{\xi} = 0$. In [GQR24, Theorem 3.2] it was shown that under a specific condition, known as the *Bolker condition*, the operators of type Λ are in fact general pseudodifferential operators from $\mathcal{E}'(X)$ to $\mathcal{D}'(X)$. We will not go into further details here, but instead refer to [KQR23] for the definition and analysis of the Bolker condition in the setting of a linear velocity model and common-offset geometry.

The critical insight for us is that by definition of X, and provided we use the affine linear background velocity, the Bolker condition is fulfilled and Λ is a ΨDO in our setting.

In the following, we specify the (general) ΨDOs K and M, and also the weights W we consider in both cases of (2.2.14). For every reconstruction operator, we calculate an explicit expression for $K^* e_\gamma^k$, which is needed for the reconstruction kernel (2.2.20).

4.1 Filtered normal operators

Here we want to tackle a generalized normal operator combined with a pseudodifferential operator:

$$\Lambda_{\mathrm{FN}} = \frac{1}{2\pi} K F_W^\dagger \psi F.$$

K must be properly supported, see [Gra20, p. 28], and of order $\kappa > 1$, so that Λ_{FN} has non-negative order. This operator is motivated by the inversion formula for the classical Radon transform R given by $u = \frac{1}{2\pi}(-\Delta)^{1/2} R^* R u$, see [Hel11, Chapter 1, Theorem 3.1].

Assume the zero offset geometry ($\alpha = 0$) and let k be the top order symbol of K. Then the top order symbol of Λ_{FN} is given by (see [GQR24, Section 3.2])

$$\sigma\left(\Lambda_{\mathrm{FN}}\right)(\boldsymbol{x}, \boldsymbol{\xi}) = k(\boldsymbol{x}, \boldsymbol{\xi}) \sum_{(s,\omega) \in J(\boldsymbol{x}, \boldsymbol{\xi})} \frac{\psi(s) W(s, \boldsymbol{x}) A(s, \boldsymbol{x})}{|\omega B(s, \boldsymbol{x})|},$$

where $J(\boldsymbol{x}, \boldsymbol{\xi}) \coloneqq \{(s, \omega) \in \mathrm{cl}(S) \times (\mathbb{R} \setminus \{\mathbf{0}\}) \colon \boldsymbol{\xi} = \omega \nabla_{\boldsymbol{x}} \varphi(s, \boldsymbol{x})\}$ and

$$B(s, \boldsymbol{x}) = \det \begin{pmatrix} \nabla_{\boldsymbol{x}} \varphi(s, \boldsymbol{x})^\top \\ \partial_s \nabla_{\boldsymbol{x}} \varphi(s, \boldsymbol{x})^\top \end{pmatrix}$$

is the Beylkin determinant, see [BCS01, equation (5.1.17)]

Here we will use $K = (-\Delta)^{i/2}$, $i = 1, 2$, which has symbol $\|\boldsymbol{\xi}\|^i$. By looking at the top order symbol above, an obvious choice for the weight is $W = |B| / (A \|\nabla_{\boldsymbol{x}} \varphi\|)$. The reconstruction operator then has the following form

$$\boxed{\Lambda_{\mathrm{FN}}^{(i)} = \frac{1}{2\pi} (-\Delta)^{i/2} F^\dagger_{|B|/(A\|\nabla_{\boldsymbol{x}}\varphi\|)} \psi F, \qquad i \in \{1, 2\}.}$$

Now, the top order symbol of $\Lambda_{\mathrm{FN}}^{(i)}$, $i = 1, 2$, is given by (see [GQR24, Lemma 3.5])

$$\sigma\left(\Lambda_{\mathrm{FN}}^{(i)}\right)(\boldsymbol{x}, \boldsymbol{\xi}) = \|\boldsymbol{\xi}\|^{i-1} \sum_{(s,\omega) \in J(\boldsymbol{x}, \boldsymbol{\xi})} \psi(s),$$

and its order is $i - 1$.

Note that $(-\Delta)^{1/2}$ is technically not a ΨDO, but a general ΨDO that can be split into a ΨDO and a smoothing operator, as shown in Example 3.19. Further, $(-\Delta)^{1/2}$ is not properly supported and would in theory require an additional cutoff function between $(-\Delta)^{1/2}$ and $F^{\dagger}_{|B|/(A\|\nabla_{\boldsymbol{x}}\varphi\|)}$, but as we only require the evaluation of $(-\Delta)^{1/2}e_\gamma^k$ for the reconstruction kernel (2.2.20) with e_γ^k being compactly supported, we will keep the above representation.

We proceed with exactly this calculation of the term $K^\star e_\gamma^k$ for both operators.

Mollifier evaluation

For $i = 2$ we have the standard Laplacian operator. Since $-\Delta$ is symmetric (see A.2.2), we obtain (see equation (A.2.1))

$$(-\Delta)^* e_\gamma^k(\boldsymbol{x}) = (-\Delta)e_\gamma^k(\boldsymbol{x}) = \begin{cases} \frac{4(k+1)k}{\pi\gamma^{2k+2}}\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-2}\left(\gamma^2 - k\,\|\boldsymbol{x}\|^2\right), & \|\boldsymbol{x}\| < \gamma, \\ 0, & \|\boldsymbol{x}\| \geq \gamma. \end{cases}$$

So now, consider the case $i = 1$. For the square root of the Laplacian, we have, see also Example 3.19,

$$\left(\mathcal{F}(-\Delta)^{1/2} f\right)(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|\,\widehat{f}(\boldsymbol{\xi}).$$

This again is a symmetric operator (see A.2.3).

For $f \in \mathcal{D}(X)$, $a \neq 0$ it holds $(\mathcal{F}[T^a f])(\boldsymbol{\xi}) = |a|^{-d}\,\widehat{f}\,(\boldsymbol{\xi}/a)$, where $T^a f(\boldsymbol{x}) \coloneqq f(a\boldsymbol{x})$ is the dilation operator. Since $e_\gamma^k(\boldsymbol{x}) = \frac{1}{\gamma^2} e_1^k(\frac{\boldsymbol{x}}{\gamma})$, we have for the Fourier Transform of e_γ^k (using $a = 1/\gamma$)

$$\widehat{e_\gamma^k}(\boldsymbol{\xi}) = \frac{1}{\gamma^2}\left(\mathcal{F}\left[T^{\frac{1}{\gamma}} e_1^k\right]\right)(\boldsymbol{\xi}) = \widehat{e_1^k}(\gamma\boldsymbol{\xi}). \tag{4.1.1}$$

Using the FIO representation of $(-\Delta)^{1/2}$, we get

$$\begin{aligned}
(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) &= \frac{1}{2\pi}\int_{\mathbb{R}^2} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}}\,\|\boldsymbol{\xi}\|\,\widehat{e_\gamma^k}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\
&= \frac{1}{2\pi}\int_0^\infty \int_{\partial B_1(\mathbf{0})} \mathrm{e}^{\mathrm{i}r\boldsymbol{x}\cdot\boldsymbol{w}} r\,\widehat{e_\gamma^k}(r\boldsymbol{w}) r\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}r && \text{(polar coord.)} \\
&= \frac{1}{2\pi}\int_0^\infty \int_{\partial B_1(\mathbf{0})} \mathrm{e}^{\mathrm{i}r\boldsymbol{x}\cdot\boldsymbol{w}} r^2 \widehat{e_1^k}(\gamma r\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}r && (4.1.1) \\
&= \frac{1}{2\pi}\int_0^\infty r^2 \frac{2^k(k+1)!}{\pi}(\gamma r)^{-(k+1)} J_{k+1}(\gamma r)\int_{\partial B_1(\mathbf{0})} \mathrm{e}^{\mathrm{i}r\boldsymbol{x}\cdot\boldsymbol{w}}\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}r && (A.3.1) \\
&= \int_0^\infty r^2 \frac{2^k(k+1)!}{\pi}(\gamma r)^{-(k+1)} J_{k+1}(\gamma r) J_0(\|\boldsymbol{x}\|\,r)\,\mathrm{d}r && (A.3.2) \\
&= \frac{2^k(k+1)!}{\pi\gamma^{k+1}}\int_0^\infty r^{1-k} J_{k+1}(\gamma r) J_0(\|\boldsymbol{x}\|\,r)\,\mathrm{d}r.
\end{aligned} \tag{4.1.2}$$

Here, $\partial B_1(\mathbf{0})$ is the unit circle in $\mathbb{R}^2$, and J_k denotes the *Bessel function of the first kind* of order k

$$J_k(r) \coloneqq \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{e}^{\mathrm{i}(r\sin(\varphi)-k\varphi)}\,\mathrm{d}\varphi.$$

The type of integral in formula (4.1.2) is in general also known as a *Weber-Schafheitlin integral*. To evaluate it, we use the following theorem.

Theorem 4.2. (see [ES79, equation (2.8)])
Let x_1, x_2 be positive real numbers such that $x_2 > x_1$. Let $v_1, v_2, \mu \in \mathbb{R}$, $M \coloneqq \mu + v_1 + v_2$ so that $1 + v_1 + v_2 > 1 - \mu > -1$. Then

$$\int_0^\infty r^{\mu-1} J_{v_1}(x_1 r) J_{v_2}(x_2 r)\,\mathrm{d}r = \frac{2^{\mu-1} x_1^{v_1} x_2^{v_2-M} \Gamma\left(\frac{M}{2}\right)}{\Gamma(v_1+1)\Gamma\left(v_2 - \frac{M}{2} + 1\right)}\, {}_2F_1\left[\frac{M}{2}, \frac{M}{2} - v_2; v_1 + 1; \frac{x_1^2}{x_2^2}\right],$$

where Γ is Euler's Gamma function and ${}_2F_1$ is Gauss' hypergeometric function. *For $a, b, c \in \mathbb{R}$ and $z \in \mathbb{C}, |z| < 1$ it is given by the power series (see [Gau12; Gau88])*

$${}_2F_1(a, b; c; z) \coloneqq \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!},$$

with the Pochhammer symbol

$$(q)_n \coloneqq \begin{cases} 1, & n = 0, \\ q(q+1)\cdots(q+n-1), & n > 0. \end{cases}$$

Applying Therorem 4.2 to equation (4.1.2), where $\mu = 2-k$, $M = 3$, and after a rather extensive calculation in A.4, we get

$$(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) = \begin{cases} \frac{4^k (k+1)!\, k!}{\pi \gamma^3 (2k)!}\, {}_2F_1\left[\frac{3}{2}, \frac{1}{2} - k; 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right], & \|\boldsymbol{x}\| \le \gamma \\ -\frac{1}{2\pi \|\boldsymbol{x}\|^3}\, {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; \frac{\gamma^2}{\|\boldsymbol{x}\|^2}\right], & \|\boldsymbol{x}\| > \gamma. \end{cases}$$

To evaluate the Gauss' hypergeometric function, depending on different cases, transformation formulas are used to reduce it to a simpler form. Here, we rely on scipy's implementation scipy.special.hyp2f1[1] to calculate the necessary values. See their references in the documentation for details.

The normalized radial parts of $(-\Delta)^{1/2} e_\gamma^k$ and $(-\Delta) e_\gamma^k$ together with e_γ^k itself are displayed in Figure 4.2 for the parameters $\gamma = 1$ and $k = 3$. Notice that $(-\Delta)^{1/2} e_\gamma^k$ is no longer compactly supported. For example, this leads to the kernel-matrix not being sparse

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.hyp2f1.html

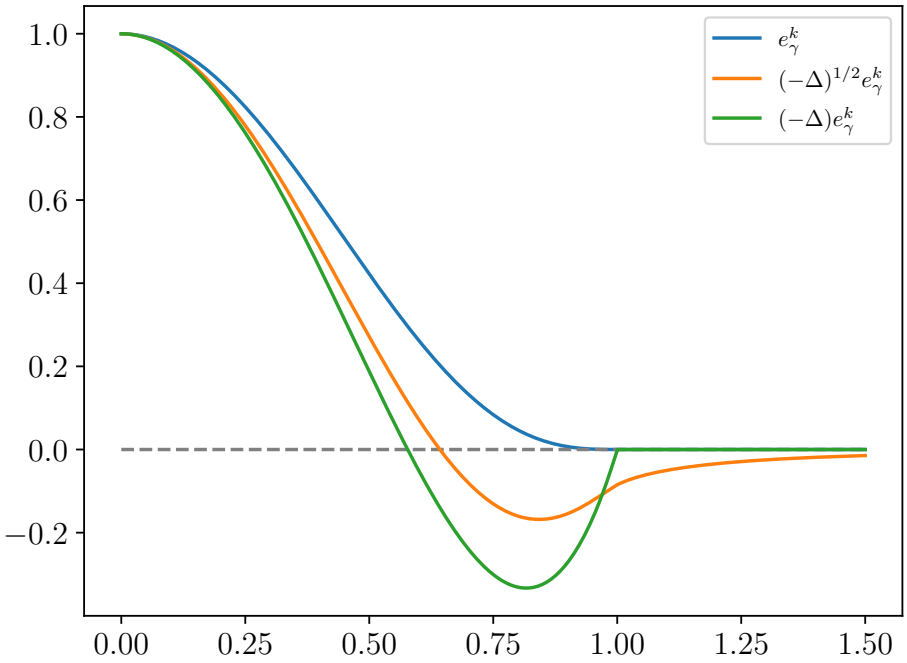


Fig. 4.2: Radial parts of the mollifier e_γ^k and the evaluated terms $(-\Delta)^{1/2}e_\gamma^k$ and $(-\Delta)e_\gamma^k$ for $\gamma = 1$ and $k = 3$. Each curve was normalized to one for better comparability.

anymore. In fact, it is dense, which leads to a much bigger memory footprint and can cause problems when the kernel-array size gets too large. Furthermore, since the evaluation of a non-trivial function in the Gauss' hypergeometric function is required, the runtime for a reconstruction is substantially longer.

4.2 Traditional Kirchhoff operators

The 2D Kirchhoff migration schemes described by [BCS01] in formulas (5.1.40) and (5.1.41) can be written as the following imaging operators, see [GQR24, Section 3.3]:

$$\Lambda_{\mathrm{K}}^{(1)} \coloneqq \frac{1}{2\pi} F_{|B|/A}^{\dagger} \left(\mathrm{Id} \otimes (\mathrm{H}\,\partial_t)\right) \psi F \quad \text{and} \quad \Lambda_{\mathrm{K}}^{(2)} \coloneqq \frac{1}{2\pi} F_{|B|/(c^2 A \|\nabla_{\boldsymbol{x}} \varphi\|)}^{\dagger} \left(\mathrm{Id} \otimes (\mathrm{H}\,\partial_t^2)\right) \psi F.$$

Here, H denotes the Hilbert transform ([Pan95, Section 3.2]) acting along the time-axis, and $\otimes$ the tensor product. Again, the cutoff function ψ is introduced to allow for the composition with F.

Under the Bolker condition, these operators are general pseudodifferential operators with top order symbols (see [GQR24, equations (34), (35)])

$$\sigma\left(\Lambda_{\mathrm{K}}^{(1)}\right)(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{(s,\omega)\in J(\boldsymbol{x},\boldsymbol{\xi})} \psi(s),$$
$$\sigma\left(\Lambda_{\mathrm{K}}^{(2)}\right)(\boldsymbol{x},\boldsymbol{\xi}) = -\mathrm{i}\frac{\|\boldsymbol{\xi}\|}{c^2(\boldsymbol{x})} \sum_{(s,\omega)\in J(\boldsymbol{x},\boldsymbol{\xi})} \operatorname{sgn}(\omega)\frac{\psi(s)}{\|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})\|^2},$$

and orders 0 and 1 respectively.

Note that H is not a local operator, meaning $(\mathrm{Id}\otimes\mathrm{H})u \notin \mathcal{D}(Y)$ for $u\in\mathcal{D}(Y)$. Therefore, it would be necessary to implement an additional cutoff function between $F_W^\dagger$ and the respective $M = \mathrm{Id}\otimes(\mathrm{H}\,\partial_t)$ or $M = \mathrm{Id}\otimes(\mathrm{H}\,\partial_t^2)$. As in the last section, we keep above representation of $\Lambda_{\mathrm{K}}^{(i)}$, $i=1,2$, since we only apply M^* to $\dot{v}_{\boldsymbol{p},\gamma,W}$ in equation (2.2.19). Nevertheless, when H is applied to $\dot{v}_{\boldsymbol{p},\gamma,W}$, the result is a non-sparse reconstruction kernel representation of $v_{\boldsymbol{p},\gamma,W}$, similar to earlier for the operator $\Lambda_{\mathrm{FN}}^{(1)}$, meaning we could potentially run into memory problems when storing the reconstruction kernels. Despite that, opposed to the filtered normal operator, the application of M^* in equation (2.2.19) can be regarded as a post-processing step on $\dot{v}_{\boldsymbol{p},\gamma,W}$, executed only immediately before the calculation of the scalar product (2.2.16). This allows us to still benefit from the sparseness of $\dot{v}_{\boldsymbol{p},\gamma,W}$. In our implementation, we use the scipy-implementation scipy.signal.hilbert[2] to apply the Hilbert transform along the time axis of the discrete kernels.

The factor $\frac{1}{c^2(\boldsymbol{x})}$ in $\sigma\left(\Lambda_{\mathrm{K}}^{(2)}\right)$ means that $\Lambda_{\mathrm{K}}^{(2)}$ de-emphasizes features where the background velocity c is large, see also [GQR24, Remark 3.10]. For the affine linear velocity $c(\boldsymbol{x}) = b + mx_2$, $b,m>0$, this means that the intensity[3] of singularities in the reconstruction decreases with depth. It can therefore be reasonable to eliminate the factor $\frac{1}{c^2(\boldsymbol{x})}$ from the weight of $\Lambda_{\mathrm{K}}^{(2)}$ and instead use

$$\boxed{\widetilde{\Lambda}_{\mathrm{K}}^{(2)} := \frac{1}{2\pi} F^\dagger_{|B|/(A\|\nabla_{\boldsymbol{x}}\varphi\|)}\left(\mathrm{Id}\otimes(\mathrm{H}\,\partial_t^2)\right)\psi F.}$$

For the Kirchhoff imaging operators the mollifier evaluation $K^* e_\gamma^k$ is trivial, as K is simply the identity operator. For M^* we mention that $H^* = -H$, see [Tit37, Theorem 102] and $(\partial_t)^* = -\partial_t$, see A.2.4.

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.hilbert.html

[3] By the intensity of a singularity $(\boldsymbol{x},\boldsymbol{\xi})\in \mathrm{WF}(n)$ in a reconstruction, we understand the value of the jump $\max\left\{\left|(\Lambda n * e_\gamma^k)(\boldsymbol{x}) - (\Lambda n * e_\gamma^k)(\boldsymbol{p})\right| : \boldsymbol{p} \text{ is a pixel in } B_\gamma(\boldsymbol{x})\right\}$, see also [Gra+20, Footnote p. 2271].

CHAPTER 5

Numerical realization/Implementation

Here, we give an overview of the required implementation steps for a reconstruction. We start with some notation for the discrete case and then explain how to solve the eikonal (2.2.7) and transport equations (2.2.8). Making use of those solvers, we then point out the involved steps to efficiently compute the reconstruction kernels (2.2.20) in case of a layered background velocity and the common-offset acquisition geometry. These reconstruction kernels then directly lead to our reconstruction. Finally, we present two other commonly used acquisition geometries and describe how we can utilize multiple measurements for a single reconstruction.

5.1 Notation

All of our discretizations will operate on a cartesian grid with equidistant spacing along each axis.

By $\mathcal{I}([a,b],n)$, $a<b$, we denote the discrete interval $[a,b]$ with number of grid-points $n\in\mathbb{N}$, including the end-points, and equidistant spacing with step size $h \coloneqq \frac{b-a}{n-1}$. To distinguish between different discrete intervals, we will write n_V and h_V for $V \coloneqq \mathcal{I}([a,b],n_V)$.

We will further write

$$\mathcal{V} \coloneqq \mathcal{M}\left([a,b]\times[c,d], \boldsymbol{n}^{\mathcal{V}}\right)$$

for a 2D mesh $\mathcal{V}$ with domain $[a,b]\times[c,d]\subset\mathbb{R}^2$, number of grid points $\boldsymbol{n}^{\mathcal{V}} = (n_1^{\mathcal{V}}, n_2^{\mathcal{V}})^\top \in \mathbb{N}^2$ and spacing $\boldsymbol{h}^{\mathcal{V}} = (h_1^{\mathcal{V}}, h_2^{\mathcal{V}})^\top \coloneqq \left(\frac{b-a}{n_1^{\mathcal{V}}-1}, \frac{d-c}{n_2^{\mathcal{V}}-1}\right)^\top \in \mathbb{R}^2_{>0}$. Define points

$$\boldsymbol{x}_{i,j}^{\mathcal{V}} \coloneqq (a,c)^\top + \left(ih_1^{\mathcal{V}}, jh_2^{\mathcal{V}}\right)^\top, \qquad i,j\in\mathbb{Z}.$$

The grid-points of $\mathcal{V}$ are then given by

$$\mathcal{V} = \left\{\boldsymbol{x}_{i,j}^{\mathcal{V}} \colon i,j\in\mathbb{N}_0,\ i<n_1^{\mathcal{V}},\ j<n_2^{\mathcal{V}}\right\}.$$

We will drop the superscripts of $\boldsymbol{n}$, $\boldsymbol{h}$ and $\boldsymbol{x}_{i,j}$ if it is clear to which mesh they belong.

The neighborhood of a grid-point $\boldsymbol{x}_{i,j}$ in $\mathcal{V}$ will then be

$$\mathcal{N}_{i,j} = \mathcal{N}(\boldsymbol{x}_{i,j}) \coloneqq \{\boldsymbol{x}_{i-1,j}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i,j-1}, \boldsymbol{x}_{i,j+1}\} \cap \mathcal{V}.$$

We will use the compact notation $f_{i,j} \coloneqq f(\boldsymbol{x}_{i,j})$ when evaluating some function f on a grid-point or when accessing the value of the array f at this grid-point. Sometimes even $f_{\mathcal{N}_{i,j}}$ will be used to refer to the set $f_{\mathcal{N}_{i,j}} \coloneqq \{f(\boldsymbol{x}) \colon \boldsymbol{x} \in \mathcal{N}_{i,j}\}$. Finally, by $f_{\mathcal{M}}$ we refer to an array (here, an array is a matrix) containing the values of f calculated on all grid-points of the mesh $\mathcal{M}$.

Array slicing

In the discrete case, the restriction of a function is a slicing operation on the corresponding computed array. For a 1D array arr, len(arr) specifies the number of elements in arr. The notation arr[ind] serves to define a sub-array, where ind denotes either a range i:j with $i \in \{0, \dots, \text{len(arr)} - 1\}$, $j \in \{-1, \dots, \text{len(arr)}\}$, $i \neq j$, or a sequence of indices. If $i < j$, the range i:j is an increasing sequence, while otherwise, it is a decreasing one. Note that we are using zero-based numbering for array indices. For a range ind = i:j, the starting index is always inclusive, while the end index is exclusive: ind $= (i, i+1, \dots, j-1)$ if $i < j$, and ind $= (i, i-1, \dots, j+1)$ if $i > j$.

Example 5.1.
Consider the one-dimensional array arr = [1, 2, 3, 4, 5, 6]. Then,

$$\text{arr[1:4]} = [2, 3, 4], \qquad \text{arr[2:}-1\text{]} = [3, 2, 1] \qquad \text{and} \qquad \text{arr[\{2, 5\}]} = [3, 6].$$

To include all array values, we use the abbreviation arr[:] as a shorthand for arr[0:len(arr)].

Slicing can be extended to higher dimensions by treating each dimension separately denoted by a comma-seperated list:

$$\text{arr} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \qquad \text{arr[1:3, :]} = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \qquad \text{arr[\{0, 2\}, 1:3]} = \begin{pmatrix} 2 & 3 \\ 8 & 9 \end{pmatrix}.$$

5.2 Solving the eikonal equation

To solve the eikonal equation (2.2.7) we use the *(factored) Fast Marching* scheme (see [Set99]).

Let $X_h := \mathcal{M}([a,b] \times [c,d], \boldsymbol{n})$ be the discrete mesh of interest and $\boldsymbol{x_s} \in X_h$ the source position. The general idea is to use an *upwind discretization* of the gradient operator to construct τ away from the source $\boldsymbol{x_s}$, where we know the traveltime to be zero, $\tau(\boldsymbol{x_s}) = 0$. This is possible because information propagates in one direction, from small values of τ to larger values, known as *causality*.

Setting $\tau_{i,j} = \tau(\boldsymbol{x}_{i,j}, \boldsymbol{x_s})$, we define the (one-sided) backward and forward finite differences for each space dimension as

$$\begin{aligned} D_{i,j}^{-x_1}\tau &:= \frac{\tau_{i,j} - \tau_{i-1,j}}{h_1}, \qquad & D_{i,j}^{+x_1}\tau &:= \frac{\tau_{i+1,j} - \tau_{i,j}}{h_1}, \\ D_{i,j}^{-x_2}\tau &:= \frac{\tau_{i,j} - \tau_{i,j-1}}{h_2}, \qquad & D_{i,j}^{+x_2}\tau &:= \frac{\tau_{i,j+1} - \tau_{i,j}}{h_2}. \end{aligned} \tag{5.2.1}$$

Now the *Godunov upwind discretization*, see, e.g. [RT92], is given by

$$\partial_{x_k}\tau_{i,j} \approx \max\left\{D_{i,j}^{-x_k}\tau,\ -D_{i,j}^{+x_k}\tau,\ 0\right\}, \quad k = 1,2,$$

which allows us to approximate

$$\|\nabla\tau_{i,j}\|^2 = \sum_{k=1}^{2}(\partial_{x_k}\tau_{i,j})^2 \approx \sum_{k=1}^{2}\max\left\{D_{i,j}^{-x_k}\tau,\ -D_{i,j}^{+x_k}\tau,\ 0\right\}^2.$$

Therefore, we need to solve the quadratic equation

$$\sum_{k=1}^{2}\max\left\{D_{i,j}^{-x_k}\tau, -D_{i,j}^{+x_k}\tau, 0\right\}^2 = \frac{1}{c_{i,j}^2} \tag{5.2.2}$$

for $\tau_{i,j}$.

The Fast Marching Algorithm now works as follows. We split the domain X_h into three disjoint sets:

- points $\boldsymbol{x}_{i,j}$ are *known* if they have been accepted and will not be recalculated,
- *front* for points that have received at least one update, but have not yet been approved to be *known*, and
- *hidden* for points that have not been touched yet.

In each iteration of the algorithm we then consider the set of previously updated grid points *front*, initially only containing the source point $\boldsymbol{x_s}$. We choose the point in this set with the smallest traveltime to continue "marching" outwards, and set it to *known*, while updating all its unknown neighbors according to the quadratic equation (5.2.2). This ensures the correct flow of information. Further details are given in Algorithm 1.

Algorithm 1: fast_marching($\boldsymbol{x_s}$)

 # initialize

$$\tau_{i,j} \leftarrow \begin{cases} 0, & \boldsymbol{x}_{i,j} = \boldsymbol{x_s}, \\ \infty, & \text{otherwise.} \end{cases}$$

known $\leftarrow \emptyset$

front $\leftarrow \{\boldsymbol{x_s}\}$

while front $\neq \emptyset$ **do**

 # find the minimal entry in front

$\boldsymbol{x}_{l,m} \leftarrow \arg\min \{\tau_{i,j} : \boldsymbol{x}_{i,j} \in \text{front}\}$

 # add $\boldsymbol{x}_{l,m}$ to known and take it out of front

known $\leftarrow$ known $\cup \{\boldsymbol{x}_{l,m}\}$

front $\leftarrow$ front $\setminus \{\boldsymbol{x}_{l,m}\}$

 # determine unknown neighborhood of $\boldsymbol{x}_{l,m}$

$\mathcal{N} \leftarrow X_h \cap \mathcal{N}_{l,m} \setminus$known

 # update front

front $\leftarrow$ front $\cup \mathcal{N}$

 # update the values at each unknown neighbor using the quadratic equation

foreach $\boldsymbol{x}_{i,j} \in \mathcal{N}$ **do**

$\tau_{i,j} \leftarrow$ solve_quadratic($\boldsymbol{x}_{i,j}$)

end

end

return τ

In order to efficiently identify the minimal entry in the *front* set, a minimum-heap data structure is used to store the elements of *front*. This structure allows insertions in complexity $\mathcal{O}(\log(m))$, where m is the current number of entries in *front*, and finding the minimal entry has complexity $\mathcal{O}(1)$.

The function solve_quadratic is responsible for solving the quadratic equation (5.2.2) at $\boldsymbol{x}_{i,j}$. To this end, the summands are rewritten as

$$\max\left\{D_{i,j}^{-x_k}\tau, -D_{i,j}^{+x_k}\tau, 0\right\}^2 = \max\left\{\alpha_k(\tau_{i,j} - \beta_k), 0\right\}^2, \quad k = 1, 2,$$

with $\alpha_k = \frac{1}{h_k}$, and $\beta_1 = \min\{\tau_{i-1,j}, \tau_{i+1,j}\}$, $\beta_2 = \min\{\tau_{i,j-1}, \tau_{i,j+1}\}$ for the first order operators (5.2.1). If we assume that all terms in equation (5.2.2) are positive, we get a simple quadratic equation

$$a\tau_{i,j}^2 + b\tau_{i,j} + \gamma = 0,$$

where

$$a = \sum_{k=1}^{2} \alpha_k^2, \qquad b = \sum_{k=1}^{2} -2\alpha_k^2 \beta_k, \qquad \gamma = -\frac{1}{c_{i,j}^2} + \sum_{k=1}^{2} \alpha_k^2 \beta_k^2. \tag{5.2.3}$$

When this result is invalid, i.e. the solution to the quadratic equation is complex, our assumption was wrong and not all terms of equation (5.2.2) were positive. In this case, we eliminate the corresponding terms in equation (5.2.3) one by one, beginning with the smallest value of $\alpha_k(\tau_{i,j} - \beta_k)$, $k = 1, 2$, and repeat the computation until we get a valid result. This outcome is guaranteed, since at least one of the terms $\alpha_k(\tau_{i,j} - \beta_k)$, $k = 1, 2$ is positive. Equivalently, as $\alpha_k \geq 0$, $k = 1, 2$, we can consider the term with the biggest β_k value as the candidate for elimination. See Algorithm 2 for more details (in the calculation for β_k, we set $\min \emptyset = 0$).

Algorithm 2: solve_quadratic($x_{i,j}$)

 # calculate coefficients for (5.2.3)
foreach dimension $k = 1, 2$ **do**
  $\alpha_k \leftarrow \frac{1}{h_k}$
  # if both forward and backward neighbors are in known, choose the direction with smaller
  # neighboring τ. If none of the neighbors are in known, we use the value 0
  $\beta_k \leftarrow \min\left\{\tau_{(i,j)+\lambda \boldsymbol{e_k}} : \lambda \in \{-1, 1\}, x_{(i,j)+\lambda \boldsymbol{e_k}} \in \text{known}\right\}$
end

calculate a, b, γ using equation (5.2.3)
$\tau_{i,j} \leftarrow \frac{-b+\sqrt{b^2-4a\gamma}}{2a}$

 # it's possible that some of the terms weren't positive
while solution $\tau_{i,j}$ is not valid **do**
  remove the term with the largest β_k from equation (5.2.2)
  calculate $\tau_{i,j}$ again with the remaining terms
end

return $\tau_{i,j}$

Note that, in place of the first order finite difference operators (5.2.1) we can also use second order (one-sided) finite differences to improve the approximation:

$$\widetilde{D}_{i,j}^{-x_k}\tau := \frac{3\tau_{i,j} - 4\tau_{(i,j)-\boldsymbol{e_k}} + \tau_{(i,j)-2\boldsymbol{e_k}}}{2h_k}, \qquad \widetilde{D}_{i,j}^{+x_k}\tau = \frac{-3\tau_{i,j} + 4\tau_{(i,j)+\boldsymbol{e_k}} - \tau_{(i,j)+2\boldsymbol{e_k}}}{2h_k},$$

for $k \in \{1, 2\}$. In this case, the coefficients of equation (5.2.3) change to

$$\alpha_k = \frac{3}{2h_k}, \quad \beta_k = \frac{1}{3}\min\left\{4\tau_{(i,j)-\boldsymbol{e_k}} - \tau_{(i,j)-2\boldsymbol{e_k}},\ 4\tau_{(i,j)+\boldsymbol{e_k}} - \tau_{(i,j)+2\boldsymbol{e_k}}\right\}.$$

To preserve desirable properties of the solution, e.g. monotonicity, we additionally have to check the following conditions to permit the use of second order operators:

- there are enough *known* points for the stencil,
- $\tau_{(i,j)-\boldsymbol{e_k}} \geq \tau_{(i,j)-2\boldsymbol{e_k}}$ for the backward and $\tau_{(i,j)+\boldsymbol{e_k}} \geq \tau_{(i,j)+2\boldsymbol{e_k}}$ for the forward operators, $k = 1, 2$.

If those are not fulfilled, we have to revert to first order. We refer to [TH16, Section 3.2] for an explanation why these conditions are required.

Further, instead of solving the eikonal equation (2.2.7), we rewrite it into a factored version

$$\|\tau_0 \nabla \tau_1 + \tau_1 \nabla \tau_0\|^2 = \frac{1}{c^2}, \qquad \tau_0(\boldsymbol{x}, \boldsymbol{x_s}) = \|\boldsymbol{x} - \boldsymbol{x_s}\|,$$

where $\tau = \tau_0 \tau_1$. This is profitable since the original eikonal equation is singular at the source, which is now captured by the term τ_0, leading to more accurate results around the source. The quantity τ_1 can be calculated with the same algorithm as in the original case, with a few adjustments to the finite difference operators and therefore the coefficients of the quadratic equation, and the initialization $\tau_1(\boldsymbol{x_s}) = \frac{1}{c(\boldsymbol{x_s})}$. For more details, we refer to [TH16]. Our implementation of the *(factored) Fast Marching* scheme is publicly available at https://github.com/kevinganster/eikonalfm or on the Python Packaging Index (PyPi) https://pypi.org/project/eikonalfm/.

5.3 Solving the transport equation

Again, let X_h and $\boldsymbol{x_s}$ as in the last section and $X = \mathrm{conv}(X_h)$ (convex hull). The next step is to compute a solution to the transport equation (2.2.8) on X_h. For this, we must first introduce a broader class of so-called (static) *Hamilton-Jacobi equations*. Here, we consider equations of the form

$$H(\boldsymbol{x}, u(\boldsymbol{x}), \partial_{x_1} u(\boldsymbol{x}), \partial_{x_2} u(\boldsymbol{x})) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in X,$$

where $f \colon X \to \mathbb{R}$, $u \colon X \to \mathbb{R}$, and the *Hamiltonian* H is a continuous scalar function on $X \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$. From now on, H will not depend on $\boldsymbol{x}$ directly, and we will write $H(u, p, q)$ as a shorthand for $H(u(\boldsymbol{x}), \partial_{x_1} u(\boldsymbol{x}), \partial_{x_2} u(\boldsymbol{x}))$. As an approximation, we use the *Lax-Friedrichs numerical Hamiltonian* ([OS91, equation (2.3)])

$$\begin{aligned} H^{LF}(u_{i,j}, u_{\mathcal{N}_{i,j}}) = H\left(u_{i,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2h_1}, \frac{u_{i,j+1} - u_{i,j-1}}{2h_2}\right) \\ - \alpha_1 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{2h_1} - \alpha_2 \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{2h_2}, \end{aligned} \tag{5.3.1}$$

where the *artificial viscosities* α_k are chosen in a way to ensure monotonicity, consistency and a differenced form of the flux H^{LF}, which in turn guarantees the convergence of the scheme, see [CL84; OS91]. We follow the reccomendation by [LQZ12, equation (A-4)] and set

$$\begin{aligned}\alpha_1 &= \max_{A\le u\le B,\, C\le p\le D,\, E\le q\le F} \{|\partial_u H(u,p,q)| + |\partial_p H(u,p,q)|\}\,,\\ \alpha_2 &= \max_{A\le u\le B,\, C\le p\le D,\, E\le q\le F} \{|\partial_u H(u,p,q)| + |\partial_q H(u,p,q)|\}\,,\end{aligned} \tag{5.3.2}$$

where $[A,B]$, $[C,D]$ and $[E,F]$ are the (numerical) ranges of the expressions $u_{i,j}$, $\frac{u_{i+1,j}-u_{i-1,j}}{2h_1}$ and $\frac{u_{i,j+1}-u_{i,j-1}}{2h_1}$ respectively.

By reformulating $H^{LF} = f$ as a fixed point problem for $u_{i,j}$ we get

$$\begin{aligned}u_{i,j} = \frac{1}{\frac{\alpha_1}{h_1}+\frac{\alpha_2}{h_2}} \Bigg[f_{i,j} - H\left(u_{i,j}, \frac{u_{i+1,j}-u_{i-1,j}}{2h_1}, \frac{u_{i,j+1}-u_{i,j-1}}{2h_2}\right)\\ +\alpha_1 \frac{u_{i+1,j}+u_{i-1,j}}{2h_1} + \alpha_2 \frac{u_{i,j+1}+u_{i,j-1}}{2h_2}\Bigg].\end{aligned}$$

We use a Gauss-Seidel iteration to obtain an update rule for the Lax-Friedrichs sweeping scheme depending on the name-giving sweeping direction. The number of sweeping directions depends on the space dimension; in 2D, there are 4. For example, when sweeping from lower left $(i,j=0)$ to upper right $(i=n_1-1,\, j=n_2-1)$, we get

$$\begin{aligned}u^{new}_{i,j} = \frac{1}{\frac{\alpha_1}{h_1}+\frac{\alpha_2}{h_2}} \Bigg[f_{i,j} - H\left(u^{old}_{i,j}, \frac{u^{old}_{i+1,j}-u^{new}_{i-1,j}}{2h_1}, \frac{u^{old}_{i,j+1}-u^{new}_{i,j-1}}{2h_2}\right)\\ +\alpha_1 \frac{u^{old}_{i+1,j}+u^{new}_{i-1,j}}{2h_1} + \alpha_2 \frac{u^{old}_{i,j+1}+u^{new}_{i,j-1}}{2h_2}\Bigg].\end{aligned}$$

According to [ZZQ06] we obtain a higher order scheme by replacing all appearances of $u_{\mathcal{N}_{i,j}}$ in equation (5.3.1) with the WENO approximations $(u_k)^{\pm}_{i,j}$:

$$\begin{aligned}u_{i-1,j} &= u_{i,j} - h_1 (u_1)^-_{i,j}, \qquad & u_{i+1,j} &= u_{i,j} - h_1 (u_1)^+_{i,j},\\ u_{i,j-1} &= u_{i,j} - h_2 (u_2)^-_{i,j}, \qquad & u_{i,j+1} &= u_{i,j} - h_2 (u_2)^+_{i,j}.\end{aligned}$$

For instance, $(u_1)^-_{i,j}$ is given by

$$(u_1)^-_{i,j} = (1-\omega_-)\left(\frac{u_{i+1,j}}{2h_1}\right) + \omega_-\left(\frac{3u_{i,j}-4u_{i-1,j}+u_{i-2,j}}{2h_1}\right),$$

with

$$\omega_- = \frac{1}{1+2\gamma_-^2}, \qquad \gamma_- = \frac{\epsilon + (u_{i,j}-2u_{i+1,j}+u_{i-2,j})^2}{\epsilon + (u_{i+1,j}-2u_{i,j}+u_{i-1,j})^2},$$

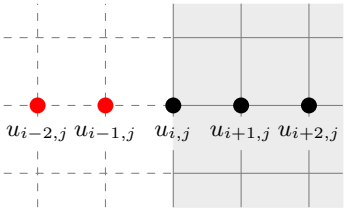
where ϵ is a small positive number to avoid division by zero. It is again important to use the most up-to-date values of the appearing terms here. Therefore the update formula using WENO approximations is now

$$u_{i,j}^{new} = u_{i,j}^{old} + \frac{1}{\frac{\alpha_1}{h_1} + \frac{\alpha_2}{h_2}} \left[f_{i,j} - H\left(u_{i,j}^{old}, \frac{(u_1)_{i,j}^{+} + (u_1)_{i,j}^{-}}{2}, \frac{(u_2)_{i,j}^{+} + (u_2)_{i,j}^{-}}{2} \right) + \alpha_1 \frac{(u_1)_{i,j}^{+} - (u_1)_{i,j}^{-}}{2} + \alpha_2 \frac{(u_2)_{i,j}^{+} - (u_2)_{i,j}^{-}}{2} \right]. \tag{5.3.3}$$

Since $u_{i\pm1,j\pm1}$, and for the WENO approximations even $u_{i\pm2,j\pm2}$, are needed for the update step (5.3.3), we have to perform extrapolation for ghost points lying outside the discrete domain. The authors of [ZZQ06, Remark 8] suggest the use of linear extrapolation since higher order extrapolations may cause instability. When looking for example at the left boundary there are 3 possible cases. The illustration on the right-hand side of each case visually shows the situation. Red dots lie outside the discrete domain and require extrapolation, while black dots do not.

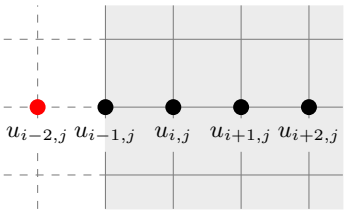
$\underline{i = 0}$: The ghost points are $u_{i-2,j}$ and $u_{i-1,j}$. Linear extrapolation leads to

$$u_{i-2,j} = 3u_{i,j} - 2u_{i+1,j}, \quad u_{i-1,j} = 2u_{i,j} - u_{i+1,j}.$$

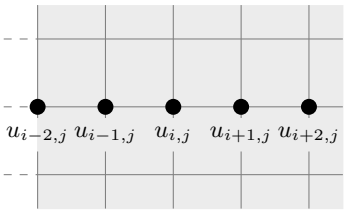


$\underline{i = 1}$: In this case the only ghost point is $u_{i-2,j}$. It is calculated by

$$u_{i-2,j} = 2u_{i-1,j} - u_{i,j}.$$



$\underline{i = 2}$: Here all required points are inside the grid, therefore no extrapolation is necessary.



Analoguously, we can perform the same extrapolation on all other boundaries.

Since this is an iterative scheme, we need a stopping criterion to determine the convergence of the numerical algorithm. A typical one in this case is when the residual, i.e. the norm of the difference between the current and last iterate, is smaller than some tolerance $\delta > 0$. This works well when the residual is monotonically decreasing, but could fail when

the residual is oscillatory. Therefore [CCK13, Chapter 4] and [SQ10, (3.16)] suggest using an additional stopping criterion, namely to stop when

$$\left\|u^{(n+1)} - u^{(n)}\right\| \leq \delta \quad \text{and} \quad \frac{\left\|u^{(n+1)} - u^{(n)}\right\| + h^p}{\left\|u^{(n)} - u^{(n-1)}\right\|} > 1,$$

where $u^{(n)}$ is the n-th iterate of the scheme, $h = \max\{h_1, h_2\}$ and p depends on the finite difference operators used. In the case of WENO approximations, $p = 3$.

The general algorithm from [LQZ12] is outlined in Algorithm 3.

Algorithm 3: fast_sweeping($\boldsymbol{x_s}$,δ)

for $i = 0 : n_1$, $j = 0 : n_2$ **do**
  $u_{i,j} \leftarrow$ initialize with suitable starting values
end
 # fix values in a 3x3 square around the source
fixed $\leftarrow \{\boldsymbol{x_s} + (ih_1, jh_2),\ i, j \in \{-1, 0, 1\}\} \cap X_h$
 # list of sweeping directions
directions $\leftarrow$ [
  $i = 0 : n_1,\quad j = 0 : n_2,\quad$ *# lower left to upper right*
  $i = n_1 - 1 : -1,\ j = 0 : n_2,\quad$ *# lower right to upper left*
  $i = 0 : n_1,\quad j = n_2 - 1 : -1,$ *# upper left to lower right*
  $i = n_1 - 1 : -1,\ j = n_2 - 1 : -1$ *# upper right to lower left*
]
$i_{dir} \leftarrow 0$ *# index for the current sweeping direction*

last_residual, residual $\leftarrow \infty$
$h \leftarrow \max\{h_1, h_2\}$
while residual $> \delta$ **and** $\frac{\text{residual}+h^3}{\text{last\_residual}} \leq 1$ **do** *# stop criterion for given tolerance $\delta > 0$*
  last_residual $\leftarrow$ residual
  for i, j in directions$[i_{dir}]$ **do**
    if $\boldsymbol{x}_{i,j} \notin$ fixed **then**
      $u_{i,j} \leftarrow$ update using (5.3.3), extrapolating when needed
    end
    $i_{dir} = (i_{dir} + 1) \mod 4$ *# cycle to next sweeping direction*
  end
  residual $\leftarrow \left\|u^{new} - u^{old}\right\|_\infty$
end

In case of the transport equation (2.2.8), similar to the eikonal equation, we factorize $a = a_0 a_1$, with $a_0 := \frac{1}{2\sqrt{2\pi}\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}$ ([LQB14, equation (8)]) being a solution of (2.2.8) with $c \equiv 1$. This is again done to capture the source singularity in the analytically known term a_0, so that a_1 is smooth near the source. Given the numerical solution τ_1 for the factored

eikonal equation, we now use fast sweeping to solve for a_1. Therefore the Hamiltonian in question reads (see A.1.2, $u \mathrel{\widehat{=}} a_1$, $\nabla_{\boldsymbol{x}} a_1 \mathrel{\widehat{=}} (p, q)$)

$$H(u,p,q) = \begin{pmatrix} p \\ q \end{pmatrix} \cdot \left[\frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}} \nabla_{\boldsymbol{x}} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) + \frac{(\boldsymbol{x}-\boldsymbol{x_s})\,\tau_1(\boldsymbol{x},\boldsymbol{x_s})}{\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}} \right] + u \left[\frac{\boldsymbol{x}-\boldsymbol{x_s}}{2\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}} \cdot \nabla_{\boldsymbol{x}} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) + \frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}} \Delta_{\boldsymbol{x}} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) \right].$$

We note that $\nabla_{\boldsymbol{x}} \tau_1$ and $\Delta_{\boldsymbol{x}} \tau_1$ can be calculated using standard second order central (in the interior) and one-sided (at the boundaries) finite differences. The artificial viscosities are set according to equation (5.3.2), for example

$$\alpha_1 = \max_{\boldsymbol{x} \in X_h} \left\{ \epsilon + \left| \frac{\boldsymbol{x}-\boldsymbol{x_s}}{2\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}} \cdot \nabla_{\boldsymbol{x}} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) + \frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}} \Delta_{\boldsymbol{x}} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) \right| + \left| \frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}} \partial_{x_1} \tau_1(\boldsymbol{x},\boldsymbol{x_s}) + \frac{(x_1 - (\boldsymbol{x_s})_1)\,\tau_1(\boldsymbol{x},\boldsymbol{x_s})}{\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}} \right| \right\},$$

where ϵ is a small positive number to account for numerical inaccuracies (and prevent division by zero in equation (5.3.3)). The remaining artificial viscosity α_2 is set accordingly.

The initial condition in equation (2.2.8) suggests to initialize $u_{i,j} = \sqrt{c_{i,j}}$.

5.4 Computing reference kernels

We now explain how we can efficiently calculate the reconstruction kernels using the numerical solvers for the eikonal and transport equation just presented in the case of the common-offset geometry. Let c be layered ($c(\boldsymbol{x}) = c(x_2)$), $\alpha \geq 0$ be the common offset, $\mathcal{T} = \mathcal{I}([t_{\min}, t_{\max}], n_{\mathcal{T}})$ the discrete time interval and $\mathcal{S} = \mathcal{I}([s_{\min}, s_{\max}], n_{\mathcal{S}})$ the discrete parameters for the source and receiver positions in equation (2.2.9). From here on out, we assume that $n_{\mathcal{S}}$ is odd, which is not a restriction in general, but eases the presentation of the reference kernels since $s_{\text{mid}} = \frac{s_{\min}+s_{\max}}{2}$ is contained in $\mathcal{S}$.

We first need to start with the calculation of $\mathcal{L}_{(s,t)}$ (and therefore φ) and the integrand in equation (2.2.20). To make this efficient, using the layered background velocity and the common-offset geometry, we enjoy the following invariances (see A.5.1, A.5.2, A.5.3)

$$\begin{aligned} \tau\left(\boldsymbol{x}, (r,0)^\top\right) &= \tau\left((x_1 - r, x_2)^\top, \mathbf{0}\right), \\ a\left(\boldsymbol{x}, (r,0)^\top\right) &= a\left((x_1 - r, x_2)^\top, \mathbf{0}\right), \end{aligned} \qquad \boldsymbol{x} \in \mathbb{R}^2_+, \quad r \in \mathbb{R},$$

$$\mathcal{L}_{(s,t)} = \mathcal{L}_{(0,t)} + \begin{pmatrix} s \\ 0 \end{pmatrix} \qquad s \in \mathcal{S},\ t \in \mathcal{T}. \tag{5.4.1}$$

This means that if we know τ and a with respect to the origin, we basically know them for every source/receiver position $(r,0)^\top$ at the surface, and with that especially also $\mathcal{L}_{(s,t)}$. When we actually want to use these relations in the case of a finite (and discrete) domain though, we have to be careful about the mesh on which the respective arrays have to be calculated. Let $\tau_{-\alpha}(\boldsymbol{x}) := \tau\left(\boldsymbol{x},(-\alpha,0)^\top\right)$ and $\tau_{+\alpha}(\boldsymbol{x}) := \tau\left(\boldsymbol{x},(+\alpha,0)^\top\right)$. To get the reflection isochrones (2.2.18), we specifically need (set $r=\pm\alpha$ in equation (5.4.1))

$$\begin{aligned}\varphi(0,\boldsymbol{x}) &= \tau_{-\alpha}(\boldsymbol{x}) + \tau_{+\alpha}(\boldsymbol{x}) \\ &= \tau\left(\boldsymbol{x}+(\alpha,0)^\top,\mathbf{0}\right) + \tau\left(\boldsymbol{x}+(-\alpha,0)^\top,\mathbf{0}\right)\end{aligned}, \qquad \boldsymbol{x}\in\mathcal{M},$$

on the *"traveltime/amplitude" mesh* $\mathcal{M}$. Set $i_\alpha := \lceil \alpha/h_1^{\mathcal{M}} \rceil$[1] as the number of points on the x_1-axis of $\mathcal{M}$ corresponding to length α rounded up to the next grid point. The consequence of using equation (5.4.1) is that the traveltime and amplitude with respect to the origin then have to be calculated on a mesh $\mathcal{M}^{+\alpha}$, which is $\mathcal{M}$ enlarged by i_α grid points on both sides of the x_1-axis, meaning $\boldsymbol{n}^{\mathcal{M}^{+\alpha}} = \left(\boldsymbol{n}_1^{\mathcal{M}} + 2i_\alpha, \boldsymbol{n}_2^{\mathcal{M}}\right)^\top$, see Figure 5.1. Given the arrays $\tau_{\mathcal{M}^{+\alpha}}$ and $a_{\mathcal{M}^{+\alpha}}$ for $\boldsymbol{x_s}=\mathbf{0}$ from Algorithms 1 and 3, we then calculate

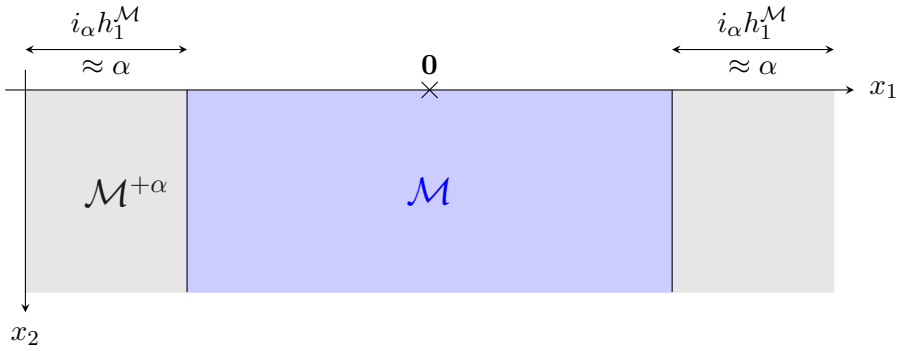


Fig. 5.1: traveltime/amplitude mesh $\mathcal{M}$ and its expanded version $\mathcal{M}^{+\alpha}$.

$$(\tau_{-\alpha})_{\mathcal{M}} = \tau_{\mathcal{M}^{+\alpha}}[2i_\alpha : 2i_\alpha + n_1^{\mathcal{M}},\ :\ ], \qquad (\tau_{+\alpha})_{\mathcal{M}} = \tau_{\mathcal{M}^{+\alpha}}[0 : n_1^{\mathcal{M}},\ :\ ],$$

and the same holds for $(a_{\pm\alpha})_{\mathcal{M}^{+\alpha}}$. Note that if the weight W fulfills the shift invariance

$$W(s-p_1,\boldsymbol{x}-\hat{\boldsymbol{p}_1}) = W(s,\boldsymbol{x}), \qquad p_1\in\mathbb{R},\ \hat{\boldsymbol{p}_1} = \begin{pmatrix} p_1 \\ 0 \end{pmatrix},$$

we only need values of W for $s=0$. From the calculations in A.5.1, A.5.2, A.5.3 one can see that this is the case for all weights we use. Therefore let $W_{\mathcal{M}}$ be the weight evaluated for $s=0$ on the mesh $\mathcal{M}$. For example $W=A$ would lead to $W_{\mathcal{M}} = \frac{(a_{-\alpha})_{\mathcal{M}}(a_{+\alpha})_{\mathcal{M}}}{c^2|_{\mathcal{M}}}$, where

[1] For $x\in\mathbb{R}$ we have $\lceil x\rceil = \min\{m\in\mathbb{Z}\colon x\le m\}$

the multiplication and division have to be understood element-wise. Now, $\varphi_{\mathcal{M}}$ and the integrand factor from equation (2.2.20) can be expressed as

$$\varphi_{\mathcal{M}} = (\tau_{-\alpha})_{\mathcal{M}} + (\tau_{+\alpha})_{\mathcal{M}}, \qquad I_{\mathcal{M}} := \frac{c|_{\mathcal{M}} W_{\mathcal{M}}}{\sqrt{1 + c^2|_{\mathcal{M}} \nabla(\tau_{-\alpha})_{\mathcal{M}} \cdot \nabla(\tau_{+\alpha})_{\mathcal{M}} + \epsilon}},$$

where, again, the operations have to be understood element-wise and the gradients of $(\tau_{-\alpha})_{\mathcal{M}}$ and $(\tau_{-\alpha})_{\mathcal{M}}$ are calculated using finite differences. The small value $\epsilon > 0$ is added to help with numerical inaccuracies.

Example 5.2.
Let $\mathcal{M} = \mathcal{M}\left([-5,5] \times [0,5], (101,51)^\top\right)$ and $\alpha = 5$.
Then $\boldsymbol{h}^{\mathcal{M}} = (0.1, 0.1)^\top$, $i_\alpha = \lceil 5/0.1 \rceil = 50$ and we have to calculate $\tau(\cdot, \mathbf{0})$ on the expanded mesh $\mathcal{M}^{+\alpha} = \mathcal{M}\left([-10,10] \times [0,5], (201,51)^\top\right)$ so that the required slicing operations are possible.

To then compute the approximated reflection isochrones for a given $t \in \mathcal{T}$ and $s = 0$, which are simply the contour lines $\varphi(0, \boldsymbol{x}) = t$, we use our own implementation of the Marching Squares algorithm on $\varphi_{\mathcal{M}}$ to find the corresponding indices. The method of Marching Squares is a well known algorithm to find contour lines of a function, and is simply the 2D version of the Marching Cubes algorithm from [LC87]. We write those indices resulting from the Marching Squares algorithm as list of 2-tuples and denote them by ind. The corresponding list of real positions, which are simply obtained by multiplying each entry in ind with $\boldsymbol{h}^{\mathcal{M}}$, are then denoted by points. The same indices can then also be used to access the values of the integrand array $I_{\mathcal{M}}$. However, as an entry in ind actually lies on an edge connecting points in $\mathcal{M}$, see Figure 5.2, we access the values in $I_{\mathcal{M}}$ by linear interpolation along the respective edge.

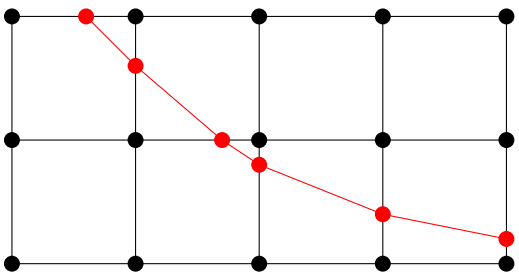


Fig. 5.2: Illustration of the contour indices ind returned by the Marching Squares algorithm. The black dots correspond to the mesh $\mathcal{M}$, while the red dots indicate the entries in ind of a contour calculated by the Marching Squares algorithm.

Note that the mesh $\mathcal{M}$ has to contain the origin and should be symmetric with respect to the x_2-axis. We obviously also need to make sure that $\mathcal{M}$ is big enough so that the

reflection isochrones for the largest t-value in $\mathcal{T}$ are fully contained inside it: $\mathcal{L}_{(0,t_{\max})} \subset \mathcal{M}$. Otherwise some parts of the reflection isochrones are missing, see Figure 5.3, and it is very likely that values in the integral from equation (2.2.20) are left out, which in turn leads to invalid results. This problem can be easily fixed by choosing a first guess for

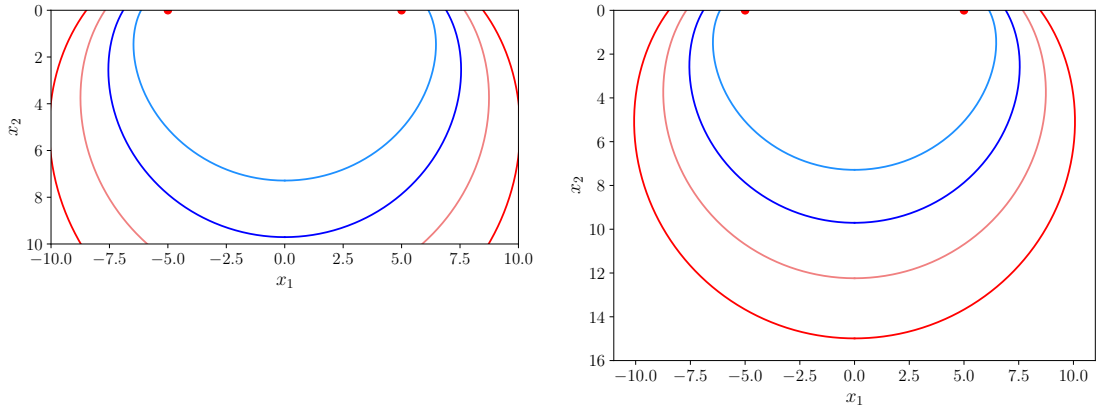


Fig. 5.3: Some reflection isochrones plotted on two different meshes $\mathcal{M}$. On the left side, the mesh is too small to fully contain the reddish isochrones. The mesh on the right however is large enough and is therefore the only sensible choice in this example.

$\mathcal{M}$, calculating $\varphi_{\mathcal{M}}$ as discussed earlier via Fast Marching, and plotting the reflection isochrones for $t = t_{\max}$. If the isochrones $\mathcal{L}_{(0,t_{\max})}$ are not contained in the mesh, we adjust it as needed.

Remark 5.3. In practice we only need the right half of the mesh $\mathcal{M}$, since the background velocity is only depth-dependent and therefore the traveltime and amplitude are symmetric with respect to the x_2-axis. Thus, we can simply calculate all the quantities we established in this section on the right half of $\mathcal{M}$ and then mirror them to the negative x_1 side. By doing this, we reduce the computation time by half.

For example, given the mesh $\mathcal{M} = \mathcal{M}\left([-5,5]\times[0,5],(101,51)^\top\right)$, the right half of it would then be $\mathcal{M}\left([0,5]\times[0,5],(51,51)^\top\right)$ and the corresponding extended mesh $\mathcal{M}\left(\left[0,\ 5+i_\alpha \boldsymbol{h}_1^{\mathcal{M}}\right]\times[0,5],(51+i_\alpha,51)^\top\right)$. Later, when we perform numerical experiments in Chapter 6, we will therefore only specify this half-mesh, but still denote it by $\mathcal{M}$.

Remark 5.4. It is possible that the program is given a $t < t_{\text{first}}$, where $t_{\text{first}} = t_{\text{first}}(s)$ is the minimal time required for a wave to travel from its source to the receiver (the first arrival time), see beginning of Chapter 4. In this case, no reflection isochrones exist and we should return the value 0 for the reconstruction kernel as (2.2.20) is an integral over the empty set. Due to numerical inaccuracies in $\varphi_{\mathcal{M}}$, the resulting contours might also be malformed for values close to t_{first}, meaning $t < t_{\text{first}} + \epsilon$ for some relatively small

value $\epsilon > 0$. For this reason, we check the condition $t < (1+\epsilon_1)\min(\varphi_{\mathcal{M}}) + \epsilon_2$ for small constants $\epsilon_1, \epsilon_2 > 0$. When this condition is met, we simply return the value 0 without the calculation of any reflection isochrones, and continue as normal otherwise.

Now, let $\mathcal{M}_{\boldsymbol{p}}$ be our reconstruction mesh, whose range for the x_1-coordinate is contained in $[s_{\min}, s_{\max}]$. For a fixed $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$, we can compute $\dot{v}_{\boldsymbol{p},\gamma,W}$ for every $s \in \mathcal{S}$ and $t \in \mathcal{T}$ using Algorithm 4, assuming $K^* e_\gamma$ is explicitly available, and with that the reconstruction kernel $v_{\boldsymbol{p},\gamma,W}(s,t)$ by applying M^* afterwards. Note that the factor $\frac{1}{\sqrt{2}}$ in the returned value is the scalar in front of the integral in equation (2.2.20), and the factor $\frac{1}{2}$ originates from the trapezoidal rule.

Algorithm 4: $\dot{v}_{\boldsymbol{p},\gamma,W}(s,t)$

ind, points ← use Marching Squares on $\varphi_{\mathcal{M}}$
 # fill an array with the values of the integrand at each point
values = [ ]
for $k = 0 : \text{len(ind)}$ **do**
  # note that the array access $I_{\mathcal{M}}$[ind[k]] actually requires interpolation
  values[k] ← $I_{\mathcal{M}}$[ind[k]] $K^* e_\gamma(\text{points}[k] + (s,0)^\top - \boldsymbol{p})$
end

 # (chained) trapezoidal rule
res ← 0
for $k = 1 : \text{len(ind)}$ **do**
  res ← res + $\|\text{points}[k] - \text{points}[k-1]\| \cdot (\text{values}[k] + \text{values}[k-1])$
end
return $(2\sqrt{2})^{-1} \cdot$ res

Remark 5.5. By replacing $K^* e_\gamma^k(\cdot - \boldsymbol{p})$ with any function ρ in equation (2.2.20), we can apply the same concepts to calculate the expression $I_{\mathcal{M}}$ and finally $\dot{v}_{\boldsymbol{p},\gamma,W}$ using Algorithm 4, provided that we have an explicit expression for ρ. In this case $\dot{v}_{\boldsymbol{p},\gamma,W} = \dot{v}_W$ is independent of $\boldsymbol{p}$ and γ. Therefore by setting $W = A$, in which case $\dot{v}_A = F\rho$, we basically described an implementation for the operator F applied to ρ. This is useful for generating synthetic data from a given function n, since we can calculate $g(s,t) = Fn(s,t)$ for any $s \in \mathcal{S}$, $t \in \mathcal{T}$.

In principle we could calculate $v_{\boldsymbol{p},\gamma,W}$ for every point $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$ on $\mathcal{S} \times \mathcal{T}$, but the number of kernels then scales quadratically with the mesh size, namely with $\boldsymbol{n}_1^{\mathcal{M}_{\boldsymbol{p}}} \times \boldsymbol{n}_2^{\mathcal{M}_{\boldsymbol{p}}}$. Instead, it is much more efficient to utilize a further shift invariance for the reconstruction

kernels, similar to the traveltime and amplitude (see A.5.3), and compute much less of them on a larger grid $\mathcal{S}_{\text{ref}} \times \mathcal{T}$:

$$v_{\boldsymbol{p},\gamma,W}(s,t) = v_{(r,p_2)^\top,\gamma,W}(s - p_1 + r, t), \qquad r \in \mathbb{R}. \tag{5.4.2}$$

It is sufficient to calculate what we call *reference kernels* $\left(v_{(s_{\text{mid}},p_2)^\top,\gamma,W}\right)_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}$ along the vertical axis $p_1 = s_{\text{mid}}$ (applying the shift operation (5.4.2) with $r = s_{\text{mid}}$). Hereby an appropriate number $n_{P_{\text{ref}}}$ of kernels with p_2 spanning accross the x_2-axis of $\mathcal{M}_{\boldsymbol{p}}$ is chosen. This set of p_2-coordinates is denoted by

$$P_{\text{ref}} := \mathcal{I}\left(\left[\min_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_2, \max_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_2\right], n_{P_{\text{ref}}}\right).$$

Further, the following set of parameters is necessary to enable the required shift operations

$$\mathcal{S}_{\text{ref}} := \mathcal{I}([s_{\min} - p_{\text{ref}}, s_{\max} + p_{\text{ref}}], n_{\mathcal{S}_{\text{ref}}}), \tag{5.4.3}$$

where p_{ref} will be defined later.

As the calculation of $\dot{v}_{\boldsymbol{p},\gamma,W}(s,t)$ is completely independent of t and $\boldsymbol{p}$, we can use parallel computing with shared arrays $\varphi_{\mathcal{M}}$, $I_{\mathcal{M}}$ and $\mathcal{S}_{\text{ref}}$. Hereby each worker is given a single $t \in \mathcal{T}$ and $p_2 \in P_{\text{ref}}$ to calculate one row $\left(\dot{v}_{(s_{\text{mid}},p_2)^\top,\gamma,W}\right)_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}[:, i_t]$, where $\mathcal{T}[i_t] = t$. Further parallelization in $s \in \mathcal{S}_{\text{ref}}$ is avoided as this would require unnecessary repeated computation of the t-contour for the reflection isochrone.

5.5 Getting reconstruction kernels from reference kernels

Now we explain how to obtain the (approximated) reconstruction kernel for any point $\boldsymbol{p} = (p_1, p_2)^\top \in \mathcal{M}_{\boldsymbol{p}}$ using the previously computed reference kernels. When the number of reference kernels is smaller than the number of reconstruction points along the p_2-axis, meaning $n_{P_{\text{ref}}} < (\boldsymbol{n}_{\mathcal{M}_{\boldsymbol{p}}})_2$, we perform linear interpolation of the two neighboring reference kernels

$$(v_{(s_{\text{mid}},p_2),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}} = \frac{p_r - p_2}{p_r - p_l}(v_{(s_{\text{mid}},p_l),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}} + \frac{p_2 - p_l}{p_r - p_l}(v_{(s_{\text{mid}},p_r),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}, \tag{5.5.1}$$

where $p_l, p_r \in P_{\text{ref}}$ correspond to the adjacent values $p_l = \max\{q \in P_{\text{ref}} : q \le p_2\}$ and $p_r = \min\{q \in P_{\text{ref}} : q > p_2\}$, and therefore $p_l \le p_2 < p_r$. In the edge case that p_2 is the biggest entry in P_{ref}, meaning p_r does not exist, we simply use the corresponding reference kernel $v_{(s_{\text{mid}},p_2),\gamma,W}$ since no interpolation is needed. Note that this interpolation might introduce horizontal artefacts in the reconstruction image, therefore $n_{P_{\text{ref}}} = (\boldsymbol{n}_{\mathcal{M}_{\boldsymbol{p}}})_2$

would be desirable, but is significantly more computationally intensive. In almost all of our numerical experiments, a much lower number of reference kernels $n_{P_{\text{ref}}}$ did not introduce any visible of this kind and is therefore sufficient.

Now that we obtained the reference kernel array $(v_{(s_{\text{mid}},p_2),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}$ corresponding to the p_2-coordinate, we have to cut out the correct part of it according to the shift relation (5.4.2). In general the shifted argument $s - p_1 + s_{\text{mid}}$ in equation (5.4.2) is not contained in the discrete set $\mathcal{S}_{\text{ref}}$, even for $s \in \mathcal{S}$. Therefore we start by replacing p_1 with the next smaller entry $\underline{p_1} = \mathcal{S}[i_{p_1}]$ in $\mathcal{S}$, meaning

$$i_{p_1} \coloneqq \left\lfloor \frac{p_1 - s_{\min}}{h_{\mathcal{S}}} \right\rfloor,$$

where for $x \in \mathbb{R}$, $\lfloor x \rfloor = \max\{y \in \mathbb{Z} \colon y \leq x\}$ is the floor function.

Example 5.6.
Let $\mathcal{S} = \mathcal{I}([-10, 10], 201) = [-10, -9.9, \ldots, 9.9, 10]$. Then we have $\underline{2.3} = 2.3$ with index $i_{2.3} = \left\lfloor \frac{2.3+10}{0.1} \right\rfloor = 123$, and $\underline{-6.84} = -6.9$ with $i_{-6.84} = \lfloor 31.6 \rfloor = 31$.

These calculations can only be carried out if $s - \underline{p_1} + s_{\text{mid}} \in \mathcal{S}_{\text{ref}}$ for all $s \in \mathcal{S}$ and $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$. To ensure this, we set

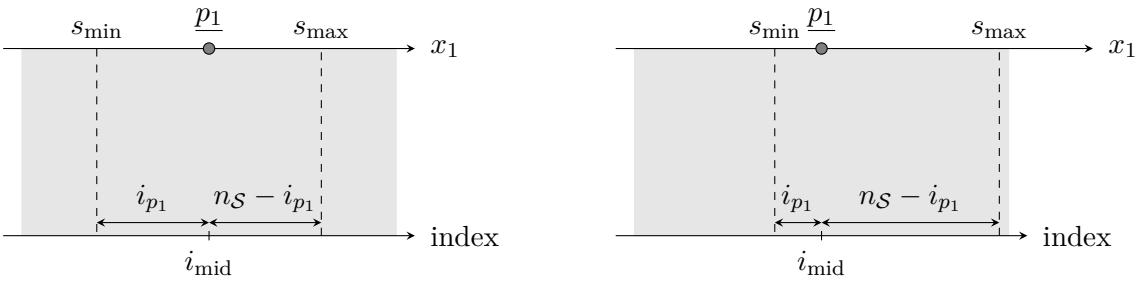
$$p_{\text{ref}} \coloneqq \max_{\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}} \left| s_{\text{mid}} - \underline{p_1} \right|$$

in the definition of $\mathcal{S}_{\text{ref}}$ (5.4.3). Therefore $n_{\mathcal{S}_{\text{ref}}} = n_{\mathcal{S}} + 2p_{\text{ref}}/h_{\mathcal{S}}$ (note that $p_{\text{ref}}/h_{\mathcal{S}} \in \mathbb{N}$). Finally, we have

$$(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}} = (v_{(s_{\text{mid}},p_2),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}[i_{\text{mid}} - i_{p_1} \colon i_{\text{mid}} - i_{p_1} + n_{\mathcal{S}},\ :]$$

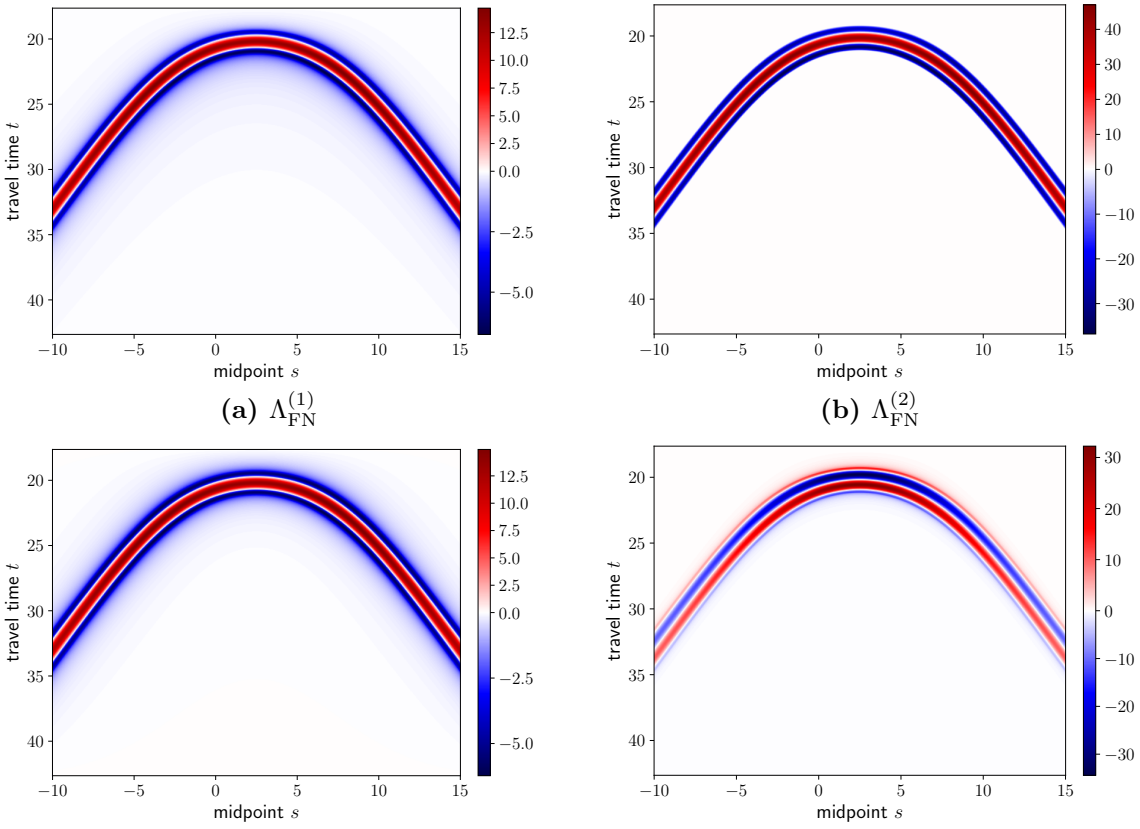
with $i_{\text{mid}} \coloneqq \frac{n_{\mathcal{S}}-1}{2}$. This is the array slicing operation implementing the shift (5.4.2). In Figure 5.4 a few examples of this slicing operation for different p_1 are illustrated.

Different reconstruction kernels $(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}$ for the imaging operators from Chapter 4 are displayed in Figure 5.5. Note that depending on the imaging operator, the corresponding kernels are sparse, but may even be dense in certain cases. This can become problematic when the kernel resolution $n_{\mathcal{S}} \times n_{\mathcal{T}}$ and the number of reference kernels $n_{P_{\text{ref}}}$ are both large. For example, assuming we use 64 bit floating point values to represent the reference kernels, then with $n_{P_{\text{ref}}} = 201$, $n_{\mathcal{S}_{\text{ref}}} = 3501$, and $n_{\mathcal{T}} = 2501$, the resulting dense reference kernel arrays would already occupy approximately 13 GB of memory. If possible, we therefore use a sparse matrix representation for the reference kernels.



(a) $\underline{p_1} = s_{\text{mid}}$ **(b)** $\underline{p_1} < s_{\text{mid}}$

Fig. 5.4: Examples for the slicing operation to obtain $(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}$ from the reference kernel $(v_{(s_{\text{mid}},p_2),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}$. The light-gray box represents the reference kernel $(v_{(s_{\text{mid}},p_2),\gamma,W})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}$, while the area enclosed in the dashed lines is the correct slice for $(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}$.



(a) $\Lambda^{(1)}_{\text{FN}}$ **(b)** $\Lambda^{(2)}_{\text{FN}}$

(c) $\Lambda^{(1)}_{\text{K}}$ **(d)** $\Lambda^{(2)}_{\text{K}}$

Fig. 5.5: Reconstruction kernels $v_{\boldsymbol{p},\gamma,W}$ for the different imaging operators. Here, the parameters are $\boldsymbol{p} = (2.5, 6)^\top$, $\gamma = 1$ and the background velocity is $c(\boldsymbol{x}) = 0.5 + 0.1x_2$.

A potential solution to this problem, which is not covered in this work, might be to utilize a low-rank or curvelet representation of the reconstruction kernels, as, e.g., proposed in [DH07; CDY09; De +09]. This would inevitably require some additions and adjustments in the implementation.

5.6 Evaluating the scalar product

We finally reached the last step in the reconstruction algorithm. Given the calculated reconstruction kernels $(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}$ we now describe the implementation of the approximate inverse in (2.2.16). We assume that the right-hand side g of equation (2.2.11) is measured on a Cartesian grid $\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}$ with $\mathcal{S}_{\text{data}} = \mathcal{I}([s_{\min}, s_{\max}], n_{\mathcal{S}_{\text{data}}})$ and $\mathcal{T}_{\text{data}} = \mathcal{I}([t_{\min}, t_{\max}], n_{\mathcal{T}_{\text{data}}})$, meaning $g_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}} \in \mathbb{R}^{n_{\mathcal{S}_{\text{data}}}\times n_{\mathcal{T}_{\text{data}}}}$. Ways to obtain these data $g_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}$ will be presented in Chapter 6. Further, we assume that the reconstruction kernels have been calculated with the same or a higher resolution than the data: $n_{\mathcal{S}_{\text{data}}} \leq n_{\mathcal{S}}$ and $n_{\mathcal{T}_{\text{data}}} \leq n_{\mathcal{T}}$. This covers the case that often occurs in practice when only a relatively small number of measurements can be made. The higher resolution of the reconstruction kernel can result in a much better reconstruction image in this case, which we will later see in a numerical experiment (see Figure 6.18). Since the data $g_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}$ and kernels $(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}$ are therefore given on different Cartesian meshes, we need an interpolation operator $\Pi\colon \mathbb{R}^{n_{\mathcal{S}}\times n_{\mathcal{T}}} \to \mathbb{R}^{n_{\mathcal{S}_{\text{data}}}\times n_{\mathcal{T}_{\text{data}}}}$ that maps the kernels onto the coarser data mesh. Here, Π will perform bilinear interpolation on the kernels.

The trapezoidal rule for the integral in equation (2.2.16) then yields

$$\Lambda n(\boldsymbol{p}) \approx h_{\mathcal{S}_{\text{data}}} h_{\mathcal{T}_{\text{data}}} \sum_{\mathcal{S}_{\text{data}}} \sum_{\mathcal{T}_{\text{data}}} \psi|_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}} \odot g_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}} \odot \Pi(v_{\boldsymbol{p},\gamma,W})_{\mathcal{S}\times\mathcal{T}}, \tag{5.6.1}$$

where $\odot$ is the elementwise multiplication (note that the integrand is 0 at the boundaries). We utilize parallelization over $\boldsymbol{p}$ to speed up the calculation on the whole reconstruction mesh $\mathcal{M}_{\boldsymbol{p}}$. To this end, we could share the list of all (reference) kernels and each worker is assigned a single point $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$ to then calculate the correct kernel from the list of reference kernels according to the last section, and finally evaluate the scalar product (5.6.1). Unfortunately, the list of all reference kernels is often quite large in terms of memory. Sharing this list among all workers is a significant overhead, to the extent that it is even slower than the non-parallel approach. However, as the calculation at point $\boldsymbol{p}$ only requires the two "p_2-neighboring" reference kernels (see (5.5.1)), this can be done more efficient in terms of parallelization overhead. We divide P_{ref} into $m \in \mathbb{N}$, $m < \text{len}(P_{\text{ref}})$ (discrete) subsets

$$[q_i, q_{i+1}] \subset P_{\text{ref}}, \quad q_i < q_{i+1},\ i = 0, \ldots, m-1, \quad q_0 = \min P_{\text{ref}},\ q_m = \max P_{\text{ref}},$$

which overlap only at the endpoint of one segment and the startpoint of the next one. In practice, m will often be the number of available processor cores. The worker for the point $\boldsymbol{p} = (p_1, p_2)^\top$ is then assigned to his respective group fulfilling $p_2 \in [q_i, q_{i+1})$.

Now, we can share only the (reference) kernels corresponding to $[q_i, q_{i+1}]$ among this group of workers, which significantly reduces the overhead and thereby accelerates the parallelization process.

In [Gan25] we provide a software package that contains our implementation of the scheme outlined in this chapter. This package also includes installation instructions and guidance on how to use the provided scripts in a README.md file.

5.7 Other acquisition geometries

So far, our implementation is based on the common-offset geometry (2.2.9) with half offset $\alpha \geq 0$, see Figure 5.6 for a visual illustration.

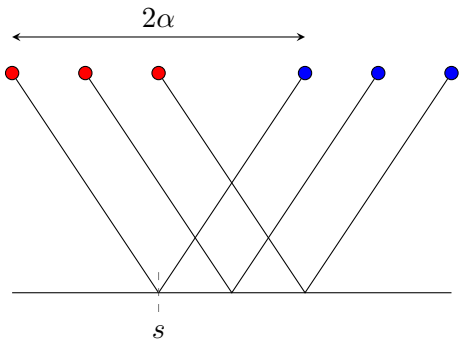


Fig. 5.6: Illustration of the common-offset acquisition geometry. Red dots correspond to source positions, while blue dots depict receiver positions.

In geophysics, this geometry is known as *common-offset gather*. Note that source and receiver definitions in (2.2.9) can also be swapped for symmetry reasons.

In this section, let F^α denote the operator F from equation (2.2.12) and $v^\alpha_{\boldsymbol{p},\gamma,W}$ the reconstruction kernel corresponding to the common-offset geometry. Now we look into two other commonly used acquisition geometries and how we can trace their respective operators and kernels back to the common-offset case, providing an inplementation for those geometries.

5.7.1 Common-midpoint geometry

As the name suggests, in this acquisition geometry the midpoint between a source-receiver pair is fixed, while their distance varies. For $x_{cm} \in \mathbb{R}$ we have

$$x_s = (x_{cm} - s, 0)^\top, \qquad x_r = (x_{cm} + s, 0)^\top, \qquad s \geq 0.$$

In geophysics, this geometry is known as *common-midpoint gather* (CMP). See Figure 5.7 for a visual illustration.

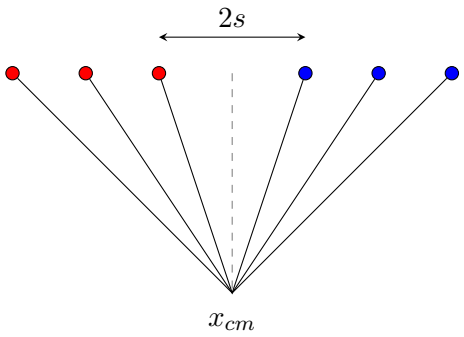


Fig. 5.7: Illustration of the common-midpoint acquisition geometry. Red dots correspond to source positions, while blue dots depict receiver positions.

The relation of the common-midpoint operator F^{cm} to the common-offset operator is

$$(F^{cm})_W \, n(s,t) = (F^s)_W \, n\,(x_{cm}, t)$$

Applying this relation to the reconstruction kernel $v^{cm}_{\boldsymbol{p},\gamma,W}$ for the common-midpoint geometry gives us

$$\begin{aligned} v^{cm}_{\boldsymbol{p},\gamma,W}(s,t) &= \left(M^\star (F^{cm})_W K^\star e(\cdot - \boldsymbol{p})\right)(s,t) \\ &= \left(M^\star (F^s)_W K^\star e(\cdot - \boldsymbol{p})\right)(x_{cm},t) = v^s_{\boldsymbol{p},\gamma,W}\,(x_{cm},t) \\ &= v^s_{(0,p_2)^\top,\gamma,W}\,(x_{cm} - p_1, t) \end{aligned}$$

Let $\mathcal{S} = [s_{\min}, s_{\max}]$, $s_{\min} \geq 0, s_{\max} > s_{\min}$ be the desired discrete interval replacing $s \geq 0$ in the definition of the the common-midpoint geometry. Further let $\mathcal{M}_{\boldsymbol{p}}$ be the reconstruction mesh and P_{ref} the x_2-coordinates for the reference kernels.

The idea is to define α_{cm} as a subset of $\mathcal{S}$ and calculate common-offset reference kernels $v^\alpha_{(0,p_2)^\top,\gamma,W}$ for $p_2 \in P_{\text{ref}}$ and common offsets $\alpha \in \alpha_{cm}$. Those reference kernels have to be calculated on the common-offset interval

$$\mathcal{S}_{co} = [x_{cm} - p_{\max}, x_{cm} - p_{\min}],$$

where $p_{\max} = \max_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_1$ and $p_{\min} = \min_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_1$.

For $s \in \mathcal{S}$, $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$ we can use bilinear interpolation of (s, p_2) on $\alpha_{cm} \times P_{\text{ref}}$ to get the corresponding reference kernel. Then we calculate the discrete index of $x_{cm} - p_1$ in $\mathcal{S}_{co}$ and obtain the correct row of the kernel.

5.7.2 Common-source geometry

Here, the source is fixed at a single point and the parameter s describes the distance of a receiver to the source. A typical example for this acquisition geometry is a marine streamer,

a ship towing an array of receivers/hydrophones along the sea surface, see Figure 5.8 (left), or in land acquisition with a vibrator truck, see Figure 5.8 (right).

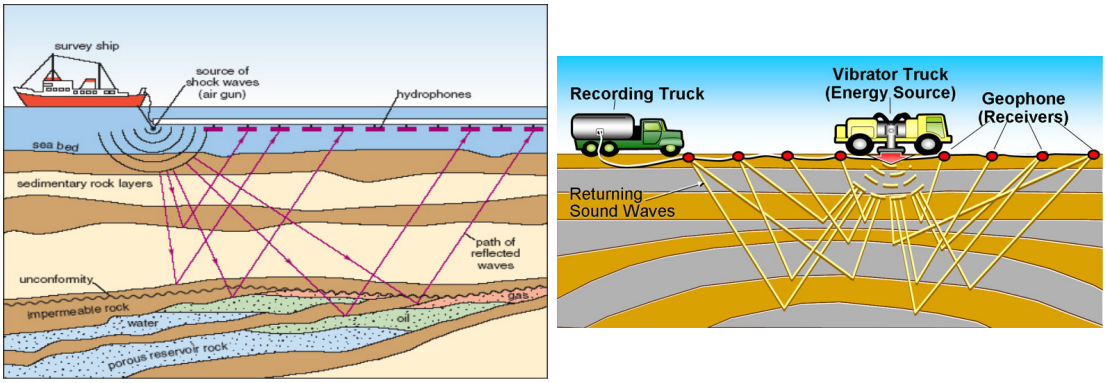


Fig. 5.8: Common-source gathers in seismic. Left: marine streamer sea acquisition, right: vibrator truck land acquisition. Original images taken from http://geologylearn.blogspot.com/2015/06/marine-and-land-seismic-aquisition.html.

For $x_{cs} \in \mathbb{R}$ we have

$$\boldsymbol{x_s}(s) = (x_{cs}, 0)^\top, \qquad \boldsymbol{x_r}(s) = (x_{cs} + s, 0)^\top, \qquad s \in \mathbb{R}.$$

See Figure 5.9 for a visual illustration.

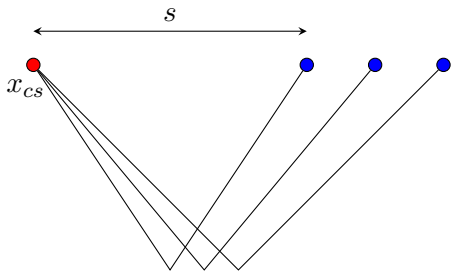


Fig. 5.9: Illustration of the common-source acquisition geometry. The red dot corresponds to the source position, while blue dots depict receiver positions.

The relation of the common-source operator F^{cs} to the common-offset operator is

$$(F^{cs})_W \, n(s,t) = \left(F^{|s/2|}\right)_W n\left(x_{cs} + \frac{s}{2}, t\right),$$

and the common-source kernel is

$$\begin{aligned} v^{cs}_{\boldsymbol{p},\gamma,W}(s,t) &= \left(M^\star \left(F^{cs}\right)_W K^\star e(\cdot - \boldsymbol{p})\right)(s,t) \\ &= \left(M^\star \left(F^{|s/2|}\right)_W K^\star e(\cdot - \boldsymbol{p})\right)\left(x_{cs} + \frac{s}{2}, t\right) = v^{|s/2|}_{\boldsymbol{p},\gamma,W}\left(x_{cs} + \frac{s}{2}, t\right) \\ &= v^{|s/2|}_{(0,p_2)^\top,\gamma,W}\left(\frac{s}{2} - p_1 + x_{cs}, t\right). \end{aligned}$$

Let $\mathcal{S} = [s_{\min}, s_{\max}]$, $s_{\max} > s_{\min}$ be the desired discrete interval replacing $s \in \mathbb{R}$ in the definition of the the common-source geometry. Further let $\mathcal{M}_{\boldsymbol{p}}$ be the reconstruction mesh and P_{ref} the x_2-coordinates for the reference kernels.

The idea is to define α_{cs} as a discrete subset of $\left[\min_{s\in\mathcal{S}} \left|\frac{s}{2}\right|, \max_{s\in\mathcal{S}} \left|\frac{s}{2}\right|\right]$ and calculate common-offset reference kernels $v^{\alpha}_{(0,p_2)^\top,\gamma,W}$ for $p_2 \in P_{\text{ref}}$ and common offsets $\alpha \in \alpha_{cs}$. Those reference kernels have to be calculated on common-offset intervals depending on $\alpha \in \alpha_{cs}$ as follows. If $\mathcal{S}$ has negative parts, which is the case if $s_{\min} < 0$, we need

$$\tilde{\mathcal{S}}^{-\alpha}_{co} = [x_{cs} - \alpha - p_{\max}, x_{cs} - \alpha - p_{\min}]\,.$$

where $p_{\max} = \max_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_1$ and $p_{\min} = \min_{\boldsymbol{p}\in\mathcal{M}_{\boldsymbol{p}}} p_1$. On the other hand, if $\mathcal{S}$ has positive parts, i.e. $s_{\max} > 0$, then we need

$$\tilde{\mathcal{S}}^{+\alpha}_{co} = [x_{cs} + \alpha - p_{\max}, x_{cs} + \alpha - p_{\min}]\,.$$

Therefore, overall we have

$$\mathcal{S}^{\alpha}_{co} = \begin{cases} \tilde{\mathcal{S}}^{-\alpha}_{co} \cup \tilde{\mathcal{S}}^{+\alpha}_{co}, & s_{\min} < 0, s_{\max} > 0 \\ \tilde{\mathcal{S}}^{-\alpha}_{co}, & s_{\min} < s_{\max} \leq 0 \\ \tilde{\mathcal{S}}^{+\alpha}_{co}, & 0 \leq s_{\min} < s_{\max}. \end{cases}$$

For $s \in \mathcal{S}$, $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$ we can use bilinear interpolation of (α_s, p_2) on $\alpha_{cs} \times P_{\text{ref}}$ to get the corresponding reference kernel, where $\alpha_s = \frac{|s|}{2}$. For the reference kernel $v^{\alpha_s}_{(0,p_2)^\top,\gamma,W}$ we calculate the discrete index of $x_{cs} \pm \alpha_s - p_1$ in $\mathcal{S}^{\alpha}_{co}$, depending on the sign of s, and obtain the correct row of the kernel.

5.8 Multiple Measurements

So far we have only considered reconstructions using a single set of measurements. One such measurement and the corresponding setting for the reconstruction can be described by the involved choice of parameters. We will call such a collection of parameters an *ensemble*. For example, in the common-offset geometry an ensemble consists of the intervals $\mathcal{S}$ and $\mathcal{T}$, and the common offset α. In case of the common-midpoint geometry, one ensemble instead consists of the common midpoint x_{cm} and the invervals $\mathcal{S}$ and $\mathcal{T}$.

Now, we want to expand our scheme to allow for a reconstruction using multiple ensembles. Let $m \in \mathbb{N}$ be the number of ensembles. For $i \in \{1, \cdots, m\}$ let F_i and

g_i describe ensemble i with $F_i n = g_i$. Using vector notation, we can write the whole equation as

$$\mathfrak{F}\mathfrak{n} = \mathfrak{g}, \qquad \text{where} \quad \mathfrak{F} \coloneqq \begin{pmatrix} F_1 \\ \vdots \\ F_m \end{pmatrix}, \ \mathfrak{g} \coloneqq \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}.$$

With inner product

$$\langle \boldsymbol{v}\,,\, \boldsymbol{w} \rangle \coloneqq \frac{1}{m} \sum_{i=1}^{m} \langle v_i\,,\, w_i \rangle\,,$$

the generalized backprojection $\mathfrak{F}_W^\dagger$ is given by $\mathfrak{F}_W^\dagger = \left((F_1)_W^\dagger \cdots (F_m)_W^\dagger \right)$.

Let ψ_i be the smooth cutoff function with respect to the component F_i. By setting $\Psi\mathfrak{F} \coloneqq (\psi_1 F_1, \ldots, \psi_m F_m)^\top$ and with some minor abuse of notation, we can write the reconstruction operator as $\boldsymbol{\Lambda} \coloneqq K\mathfrak{F}_W^\dagger M \Psi\mathfrak{F}$ and we have

$$\mathfrak{F}_W^\dagger M \Psi\mathfrak{F} = \left((F_1)_W^\dagger \cdots (F_m)_W^\dagger \right) \begin{pmatrix} M\psi_1 F_1 \\ \vdots \\ M\psi_m F_m \end{pmatrix} = \frac{1}{m} \sum_{i=1}^{m} (F_i)_W^\dagger M \psi_i F_i.$$

For a family of mollifiers $\{e_\gamma\}_{\gamma>0}$ it then holds

$$\begin{aligned} \boldsymbol{\Lambda} n * e_\gamma(\boldsymbol{p}) &= \langle \boldsymbol{\Lambda} n\,,\, e_\gamma(\cdot - \boldsymbol{p}) \rangle = \left\langle K \left(\sum_{i=1}^{m} (F_i)_W^\dagger M \psi_i F_i \right) n\,,\, e_\gamma(\cdot - \boldsymbol{p}) \right\rangle \\ &= \frac{1}{m} \sum_{i=1}^{m} \left\langle K (F_i)_W^\dagger M \psi_i F_i n\,,\, e_\gamma(\cdot - \boldsymbol{p}) \right\rangle \\ &= \frac{1}{m} \sum_{i=1}^{m} \langle \psi_i F_i n\,,\, M^\star (F_i)_W K^\star e_\gamma(\cdot - \boldsymbol{p}) \rangle \\ &= \frac{1}{m} \sum_{i=1}^{m} \left\langle \psi_i g_i\,,\, v^i_{\boldsymbol{p},\gamma,W} \right\rangle, \end{aligned}$$

where $v^i_{\boldsymbol{p},\gamma,W}$ is the reconstruction kernel for ensemble i. Therefore the whole reconstruction consists of the mean value of all reconstructions of the corresponding ensemble, meaning we can simply calculate a reconstruction independently for each ensemble and later add them all together.

CHAPTER 6

Numerical Experiments

To perform a numerical experiment, we require some form of data, i.e. a right-hand side of equation (2.2.11). We will present three different ways to obtain a right-hand side g for our experiments and show the results of the corresponding inversion procedures. In the first section, we also compare the different imaging operators among each other and reproduce some theoretical predictions. Before all of that, we state the cutoff function used for the numerical experiments.

For the cutoff function ψ we used in our numerical experiments, we begin with the 1D function

$$f(r) = \begin{cases} \mathrm{e}^{\left(-\frac{1}{r}\right)}, & r > 0, \\ 0, & r \leq 0. \end{cases}$$

Then, the function

$$g(r) = \frac{f(r)}{f(r) + f(1-r)}, \qquad r \in \mathbb{R}$$

is a smooth transition from value 0 to 1 in the unit interval $[0, 1]$ with $g(r) = 0$ for $r \leq 0$ and $g(r) = 1$ for $r \geq 1$. To obtain a smooth transition for an arbitrary interval $[a, b]$, $a < b$, we simply use

$$p(r, a, b) = g\left(\frac{r-a}{b-a}\right) = \frac{f\left(\frac{r-a}{b-a}\right)}{f\left(\frac{r-a}{b-a}\right) + f\left(\frac{b-r}{b-a}\right)}.$$

The same can be done for a smooth transition from value 1 to 0 in the interval $[c, d]$, $c < d$

$$q(r, c, d) = g\left(\frac{d-r}{d-c}\right) = \frac{f\left(\frac{d-r}{d-c}\right)}{f\left(\frac{d-r}{d-c}\right) + f\left(\frac{r-c}{d-c}\right)}.$$

An example can be seen in Figure 6.1.

This can be extended to a cutoff function in multiple dimensions by specifying a lower and upper transition for each dimension, for example, in the 2D case we have

$$\psi(s, t;\ \underline{L}_s, \overline{L}_s, \underline{R}_s, \overline{R}_s; \underline{L}_t, \overline{L}_t, \underline{R}_t, \overline{R}_t) = \underbrace{p(s, \underline{L}_s, \overline{L}_s)\ q(s, \underline{R}_s, \overline{R}_s)}_{s\text{-cutoff}}\ \underbrace{p(t, \underline{L}_t, \overline{L}_t)\ q(t, \underline{R}_t, \overline{R}_t)}_{t\text{-cutoff}}.$$

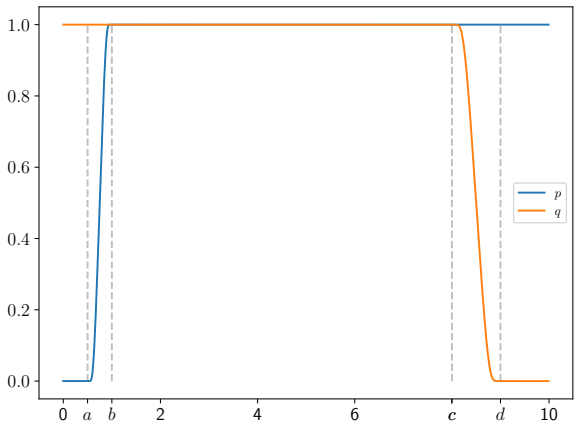


Fig. 6.1: Transition functions p and q for $a = 0.5$, $b = 1$, $c = 8$ and $d = 9$.

The parameters $\underline{L}_s, \overline{L}_s, \underline{R}_s, \overline{R}_s, \underline{L}_t, \overline{L}_t, \underline{R}_t, \overline{R}_t$ of ψ define the lower and upper intervals of the corresponding axis where the smooth cutoff is performed. Note that

$$\operatorname{supp}\psi \subset [\underline{L}_s, \overline{R}_s] \times [\underline{L}_t, \overline{R}_t] \quad \text{and} \quad \psi|_{[\overline{L}_s, \underline{R}_s] \times [\overline{L}_t, \underline{R}_t]} = 1.$$

As an abbreviation, we use tuples $L_s = (\underline{L}_s, \overline{L}_s)$, $R_s = (\underline{R}_s, \overline{R}_s)$, $L_t = (\underline{L}_t, \overline{L}_t)$ and $R_t = (\underline{R}_t, \overline{R}_t)$ to denote the corresponding transition intervals of ψ. We may also use relative values for L and R, meaning

$$\begin{aligned}\psi_{rel}(s, t; L_s, R_s; L_t, R_t) \coloneqq \psi(s, t; s_{\min} + \underline{L}_s, s_{\min} + \overline{L}_s, s_{\max} + \underline{R}_s, s_{\max} + \overline{R}_s;\\ t_{\min} + \underline{L}_t, t_{\min} + \overline{L}_t, t_{\max} + \underline{R}_t, t_{\max} + \overline{R}_t).\end{aligned}$$

For example, given $\mathcal{S}_{\text{data}} = \mathcal{I}([-5, 10], n_{\mathcal{S}_{\text{data}}})$ and $\mathcal{T}_{\text{data}} = \mathcal{I}([3, 7.5], n_{\mathcal{T}_{\text{data}}})$ and relative cutoffs $L_s = (0, 0.5)$, $R_s = (-0.8, 0)$, $L_t = (0, 0)$, $R_t = (-0.5, 0)$. Then we have $\psi_{rel}(s, t; L_s, R_s; L_t, R_t) = \psi(s, t;\ -5, -4.5,\ 9.2, 10;\ 3, 3,\ 7, 7.5)$.

Theoretically, only a smooth cutoff along the s-axis is required. However, the authors of [FQ15] mention that artifacts can arise from hard truncation of the data. To reduce these kinds of artifacts, we still include a smooth cutoff along the t-axis for our numerical experiments.

Similar experiments to those that follow have been done in [GR23] and [GQR24]. In [GR23], we also experimented with velocities c that do not fulfill the geometric optics assumption (recall Section 2.2).

Before we start with the first experiment, we will briefly discuss the choice of the regularization parameter γ in the following Remark 6.1.

Remark 6.1. The selection of the regularization parameter γ is related to the common offset α, discretization step sizes of $\mathcal{S}$ and $\mathcal{T}$, and noise level of the right-hand side g. However, we do not know a general strategy to select the value γ for practical situations. Here, we determine γ largely based on trial and error, but one useful condition is the following: γ should be chosen such that the kernel $v_{\boldsymbol{p},\gamma,W}$ is well resolved by its discrete samples. In case of the zero-offset geometry ($\alpha = 0$), a more rigorous choice of γ is outlined in [Gra+17, Remark 4.1].

Let us now begin with the first type of data acquisition, where we generate synthetic data as the right-hand side of equation (2.2.11).

6.1 Consistent data

In this case, the right-hand side g is obtained by choosing a "phantom" reflectivity n and applying the operator F to it, see Remark 5.5. As we discussed in Section 5.6, we replace $\mathcal{S}$ and $\mathcal{T}$ by $\mathcal{S}_{\text{data}}$ and $\mathcal{T}_{\text{data}}$ respectively with $n_{\mathcal{S}_{\text{data}}} \leq n_{\mathcal{S}}$, $n_{\mathcal{T}_{\text{data}}} \leq n_{\mathcal{T}}$ in this "data generation". This way we calculate a synthetic right-hand side g on $\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}$. We can then use our implementation of the chosen imaging operator to reconstruct n from these synthetic data. Note however, that this experiment setup suffers from the *inverse crime*, since we use the same implementation for the data generation and inversion. It is still reasonable to look at this case as an indicator for the functionality of the different imaging operators and we will show later that our method works properly in other scenarios as well.

For our phantom function of choice, we set

$$\begin{aligned} n \coloneqq\ & \chi_{B_\infty((0,5)^\top;2)} \chi_{B_1((0,5)^\top;2\sqrt{2})} - \chi_{B_2((0,5)^\top;1)} \\ & + 2\, \chi_{B_\infty((3,6)^\top;1.25)} + \chi_{\left\{\boldsymbol{x}\in\mathbb{R}^2_+ \,:\, x_2 \geq 6.5+\sin(\pi x_1/2)\right\}}, \end{aligned} \tag{6.1.1}$$

where we recall that $B_p(\boldsymbol{x};r) \coloneqq \left\{\boldsymbol{v}\in\mathbb{R}^2_+ :\ \|\boldsymbol{v}-\boldsymbol{x}\|_p \leq r\right\}$ is the ball around $\boldsymbol{x}$ with radius r with respect to the p-norm, and χ_A is the characteristic function on the set A. Therefore, n is made up of characteristic functions of circular and rectangular disks and a half space with a sine boundary, see Figure 6.2.

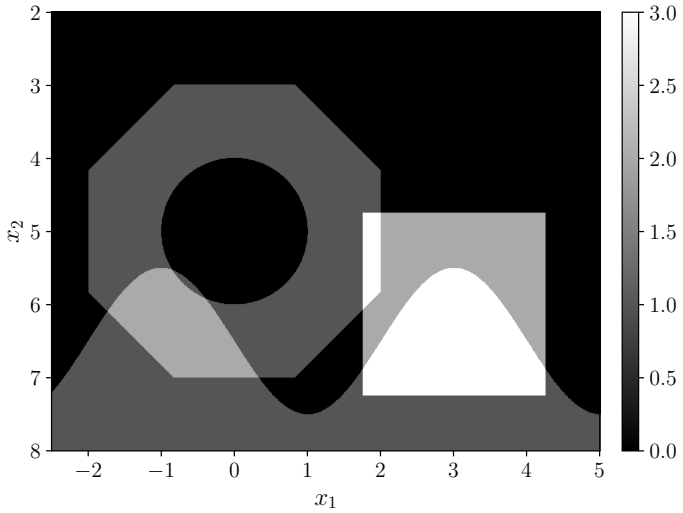


Fig. 6.2: Phantom function n from equation (6.1.1), where the colors black, darkgray, lightgray and white correspond to the numerical values 0, 1, 2 and 3 respectively.

If not explicitly stated otherwise, we use the following set of parameters

$$
\begin{aligned}
&\mathcal{M} = \mathcal{M}\left([0,28]\times[0,49],(2801,4901)^\top\right) && (\boldsymbol{h}_{\mathcal{M}} = (0.01,0.01)^\top),\\
&\mathcal{M}_{\boldsymbol{p}} = \mathcal{M}\left([-2.5,5]\times[2,8],(751,601)^\top\right) && (\boldsymbol{h}_{\mathcal{M}_{\boldsymbol{p}}} = (0.01,0.01)^\top),\\
&\mathcal{S} = \mathcal{S}_{\text{data}} = \mathcal{I}([-10,15],2501) && (h_{\mathcal{S}} = 0.01),\\
&\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([17.64,42.64],2501) && (h_{\mathcal{T}} = 0.01),\\
&P_{\text{ref}} = \mathcal{I}([2,8],201) && (h_{P_{\text{ref}}} = 0.03),\\
&c(\boldsymbol{x}) = 0.5 + 0.1x_2,\\
&\text{common-offset geometry with } \alpha = 5.0,\\
&\text{mollifier: } \gamma = 0.2,\ k = 3,\\
&\text{fast sweeping tolerance } \delta = 0.005,\\
&\text{cutoff } \psi_{rel} \text{ with } L_s = (0,0.5), R_s = (-0.5,0),\ L_t = (0,0), R_t = (-0.5,0).
\end{aligned}
\tag{6.1.2}
$$

An example of the processed consistent data for the above parameters and phantom function can be seen in Figure 6.3.

We start with a comparison of the different imaging operators.

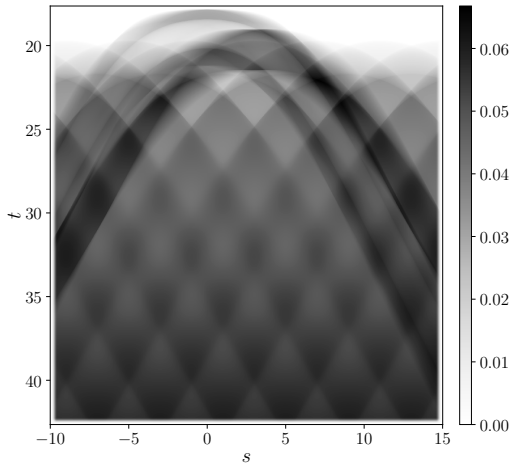


Fig. 6.3: Consistent data $\psi|_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}\, g_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}$ for $\alpha = 5$ and the affine linear velocity model $c(\boldsymbol{x}) = 0.5 + 0.1x_2$.

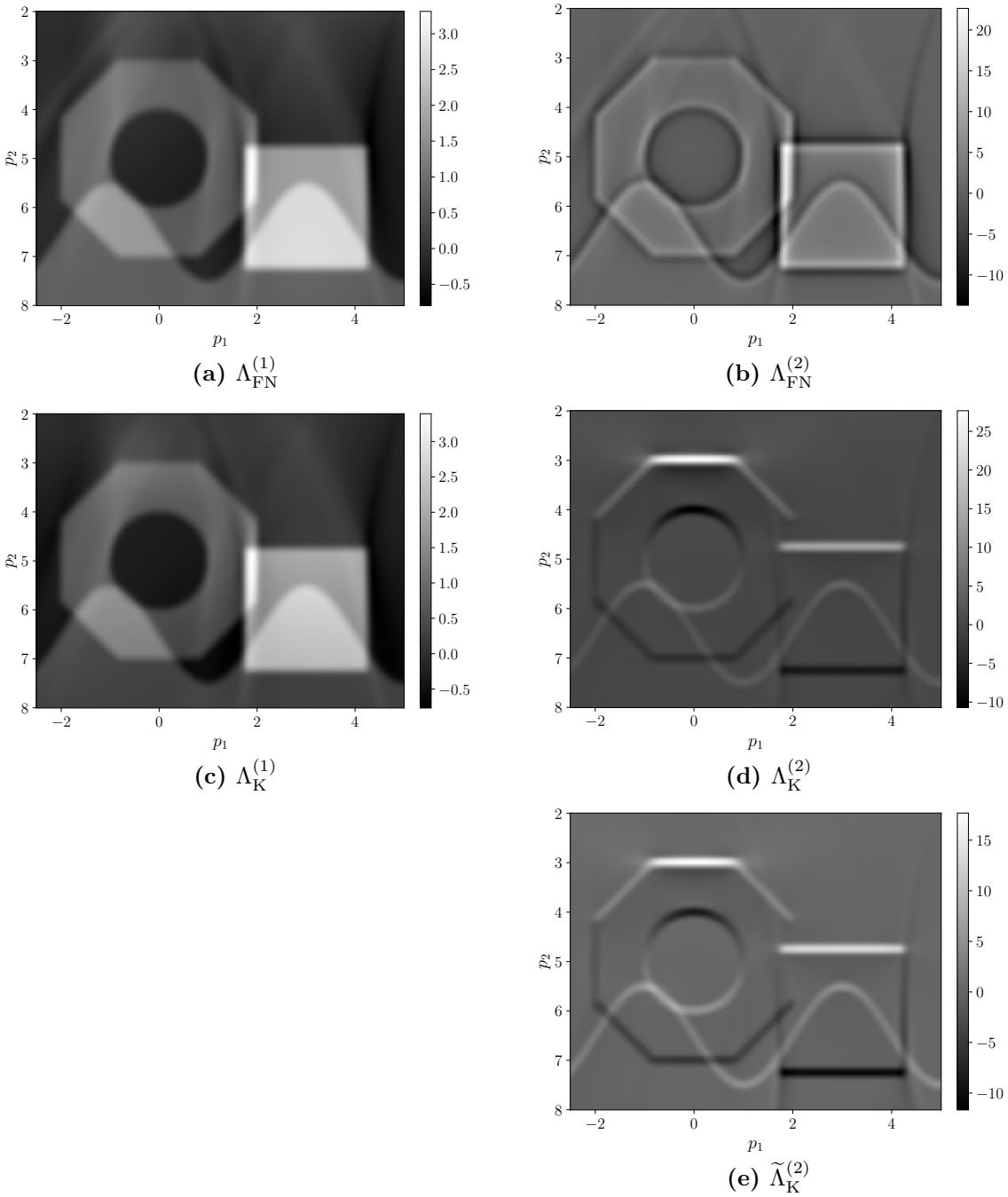
6.1.1 Comparison of the presented imaging operators

We will now show reconstructions for the various imaging operators presented in Chapter 4 using the parameters (6.1.2).

In all reconstructions in Figure 6.4, we can observe multiple qualitatively small artifacts of a curved "v" shape. Most noticeable are those running along the left and right side of the square and then bending outwards. These types of artifacts are very common in this setting, and are known as limited data artifacts, see [FQ15].

Since $\Lambda^{(1)}_{\text{FN}}$ and $\Lambda^{(1)}_{\text{K}}$ are of order zero, we expect that the singular support of n is neither emphasized nor smoothed in the reconstruction. This is confirmed in Figure 6.4 (a) and (c), as these images are quite accurate reconstructions of the phantom n, both qualitatively and quantitatively. In contrast, the imaging operators $\Lambda^{(2)}_{\text{FN}}$ and $\Lambda^{(2)}_{\text{K}}$ have order 1, meaning the singular support of n is emphasized in the corresponding reconstruction, see Figure 6.4 (b) and (d).

By looking very closely at Figure 6.4 (d), one can see that the contrast at singular points decreases with depth. For instance, the lower edge of the square is reconstructed with a similar intensity as the circle, even though the value of the phantom function is twice as high at the square. This effect will become more noticeable the further down the reconstruction domain lies. This behavior is expected and the reason for it was outlined in Section 4.2. For the adjusted operator $\widetilde{\Lambda}^{(2)}_{\text{FN}}$ this is not the case, which can be observed in Figure 6.4 (e).



(a) $\Lambda_{\mathrm{FN}}^{(1)}$ **(b)** $\Lambda_{\mathrm{FN}}^{(2)}$

(c) $\Lambda_{\mathrm{K}}^{(1)}$ **(d)** $\Lambda_{\mathrm{K}}^{(2)}$

(e) $\widetilde{\Lambda}_{\mathrm{K}}^{(2)}$

Fig. 6.4: Reconstructions using different imaging operators with parameters (6.1.2).

6.1.2 Mollifier parameters

Next, we investigate the influence of the parameters γ and k of the mollifier e_γ^k. In Figure 6.5 we can observe the effect of varying the parameters of the mollifier e_γ^k in the reconstructed image using the reconstruction operator $\Lambda_{\mathrm{FN}}^{(2)}$. On the left we used the parameters from (6.1.2) as a reference image. One can see that increasing k or decreasing

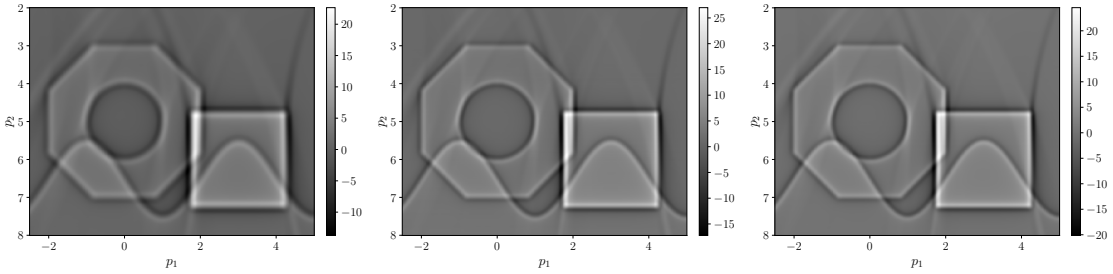


Fig. 6.5: Reconstructions using $\Lambda_{\mathrm{FN}}^{(2)}$ with different parameters for the mollifier e_γ^k. Left: $\gamma = 0.2$, $k = 3$, middle: $\gamma = 0.15$, $k = 3$, right: $\gamma = 0.2$, $k = 7$.

γ both have a similar effect in that the resulting reconstruction becomes sharper. Since the convolution $\Lambda n * e_\gamma$ was introduced in Section 2.2.1 as a smoothed version of Λn, this is exactly what we expect.

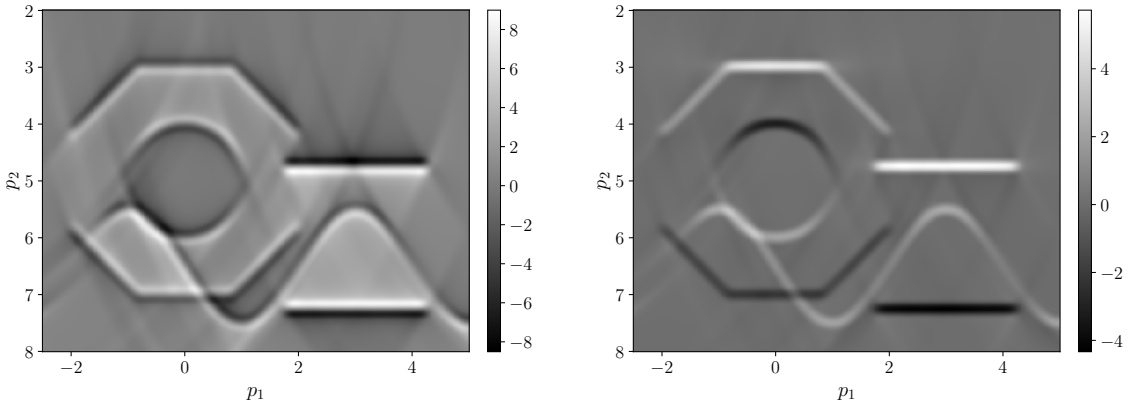
6.1.3 Constant vs affine linear background velocity

In this numerical experiment, we want to show the advantages of using an affine linear background velocity in comparison to a constant one. The distinguishing parameters from (6.1.2) for the constant case will be the following

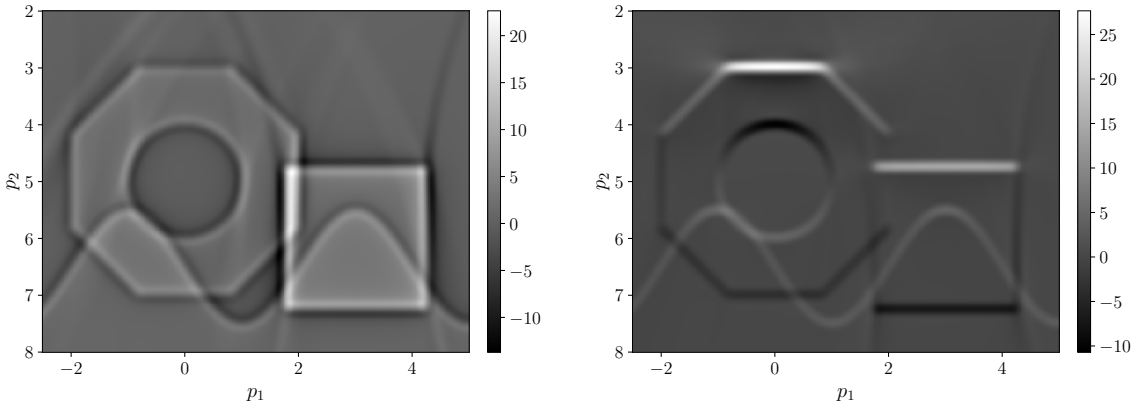
$$\mathcal{M} = \mathcal{M}\left([0,21]\times[0,20],(2101,2001)^\top\right) \qquad (\boldsymbol{h}_{\mathcal{M}} = (0.01,0.01)^\top)$$
$$\mathcal{T} = \mathcal{I}([10.05,35.05],2501) \qquad (h_{\mathcal{T}} = 0.01)$$
$$c(\boldsymbol{x}) = 1.$$

As one can observe in Figure 6.6, the operators in the constant case fail to reconstruct those singularities with a horizontal normal direction. In fact, it is known that those singularities are not visible in the reconstruction, regardless of the choice of ψ or $\mathcal{S}$, see [Gra+20]. By using the affine linear velocity however, those features are reconstructed properly.

The reason for the missing singularities in the constant case can be illustrated by looking at the reflection isochrones. A necessary condition to be able to reconstruct a point of the singular support is, that there is a reflection isochrone tangent to it (the normal direction of the isochrone at the singularity must be linearly dependent on the



(a) $\Lambda^{(2)}_{\mathrm{FN}}$ (left), $\Lambda^{(2)}_{\mathrm{K}}$ (right) for $c(\boldsymbol{x}) = 1$.



(b) $\Lambda^{(2)}_{\mathrm{FN}}$ (left), $\Lambda^{(2)}_{\mathrm{K}}$ (right) for $c(\boldsymbol{x}) = 0.5 + 0.1x_2$.

Fig. 6.6: Comparison of reconstructions with constant and affine linear background velocity using imaging operators $\Lambda^{(2)}_{\mathrm{FN}}$ and $\Lambda^{(2)}_{\mathrm{K}}$.

direction of the singularity). More precicely, if there is no $s \in \mathcal{S}$ so that $\boldsymbol{\xi} = \omega \nabla_{\boldsymbol{x}} \varphi(s, \boldsymbol{x})$ for a $\omega \in \mathbb{R}\backslash\{\mathbf{0}\}$ and the normal direction $\boldsymbol{\xi}$, then the symbol of the reconstruction operator vanishes there, see [Gra+20, Theorem 3.9] or [Gra20, Theorem 3.21]. For the constant background velocity, it is not possible to find such a reflection isochrone for a vertical feature, since the reflection isochrones are half ellipses with foci at the source and receiver positions. Therefore the only points at which those isochrones have a horizontal normal direction lie at $x_2 = 0$, regardless of t and s, see Figure 6.7. Further, this argument shows that there cannot be an inversion formula for F in the case of a constant background velocity.

In contrast, in the case of the affine linear background velocity $c(\boldsymbol{x}) = 0.5 + 0.1x_2$, it is possible to find a reflection isochrone for a vertical feature, as demonstrated in Figure 6.8. However, this does not mean the singularity will be recovered in a reconstruction, since this condition is not sufficient. A sufficient condition for a so-called *visible singularity*

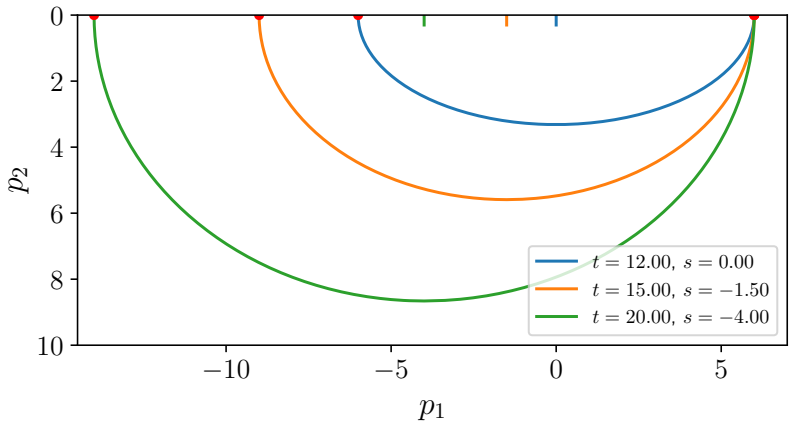


Fig. 6.7: Reflection isochrones for $c(\boldsymbol{x}) = 1$. The points with a horizontal normal direction are indicated by a red dot. Colored ticks at the top of the image show the parameters s of each isochrone.

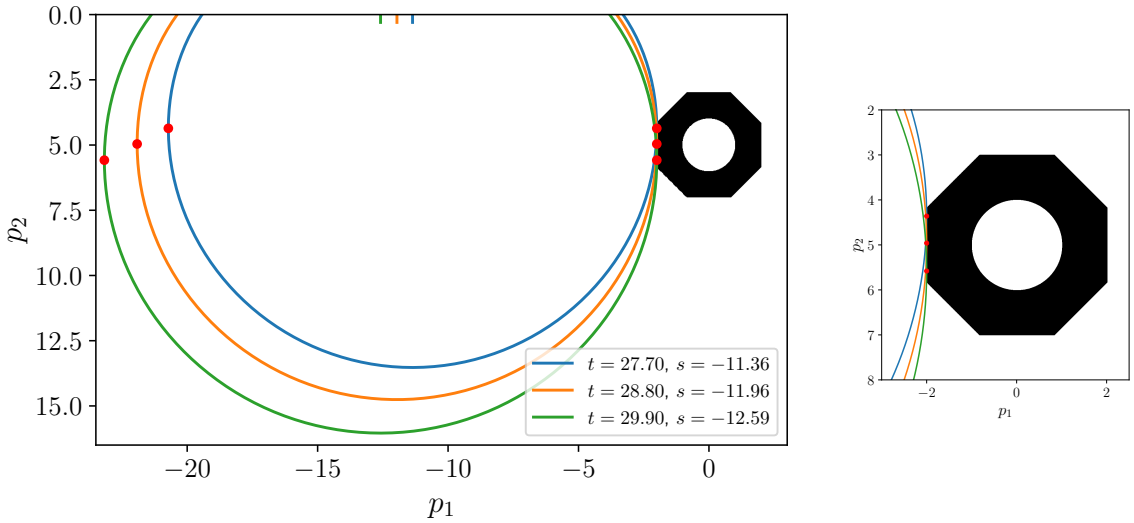


Fig. 6.8: Left: a single part of the phantom n from (6.1.1) and reflection isochrones for $c(\boldsymbol{x}) = 0.5+0.1x_2$ touching this part with a horizontal normal direction. The points with a horizontal normal direction are indicated by a red dot. The parameters s of each isochrone are marked with their respective color at the top of the image. Right: zoomed-in section on the phantom part.

would be, that the reconstruction operator is microlocally elliptic in the direction of the singularity at that point, which is the case here, see [KQR23, Proposition 3.5].

6.1.4 Consequences of incorrect common offset

Next, we want to analyze how a wrong common offset in the inversion procedure affects the reconstructions. To this end, we generate the synthetic right-hand side g as before using common offset $\alpha_{\text{data}} = 3$. For the kernel calculation we then set the common offset to a lower or higher value than α_{data}, more specifically $\alpha \in \{2.5, 3.5\}$. The parameters different from (6.1.2) are

$$\begin{aligned} &\mathcal{M} = \mathcal{M}\left([0,16]\times[0,26],(1601,2601)^\top\right) && (\boldsymbol{h}_{\mathcal{M}} = (0.01,0.01)^\top),\\ &\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([11.5,36.5],2501) && (h_{\mathcal{T}} = 0.01),\\ &\alpha_{\text{data}} = 3.0, \qquad \alpha \in \{2.5,3.5\}\,. \end{aligned}$$

If we choose α for the reconstruction lower than α_{data}, then the singularities are reconstructed further down compared to the real ones (left of Figure 6.9). The reason for this is that the traveltime τ for any fixed point increases with the common offset, meaning the traveltime to a point $\boldsymbol{p} \in \mathcal{M}_{\boldsymbol{p}}$ is larger for the data then for the kernels. If we fix a time $t \in \mathcal{T}_{\text{data}}$ for a reflection isochrone in the data generation, then the reflection isochrone in the reconstruction for the same time t lies further down, see Figure 6.10. This effect is much more pronounced the closer we are to the surface, as the influence of the common offset to φ diminishes with depth. Accordingly, if $\alpha > \alpha_{\text{data}}$, the reconstructed singularities are shifted upwards in comparison to the real singularities (see right of Figure 6.9).

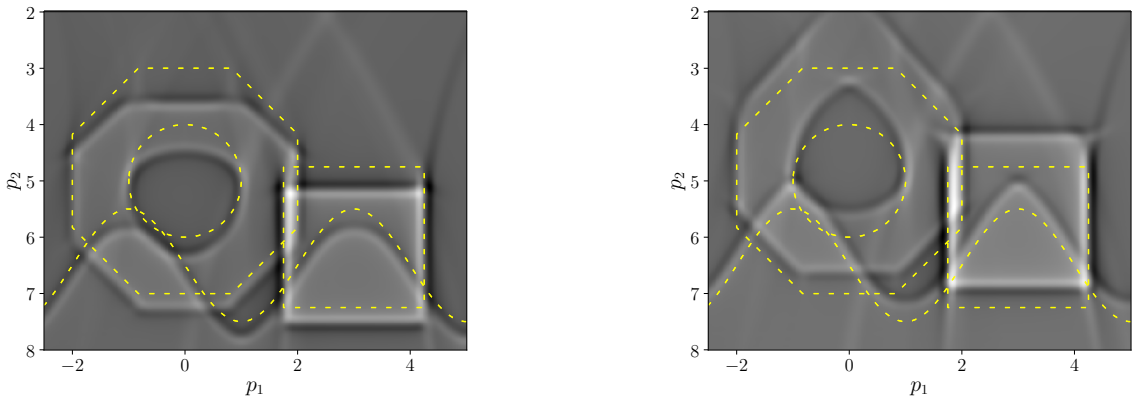


Fig. 6.9: Reconstructions with wrong common offsets α for the reconstruction. The reconstruction was done with $\alpha = 2.5$ (left) and $\alpha = 3.5$ (right), while the data were generated with $\alpha_{\text{data}} = 3$. The dotted lines indicate the actual singular support of n.

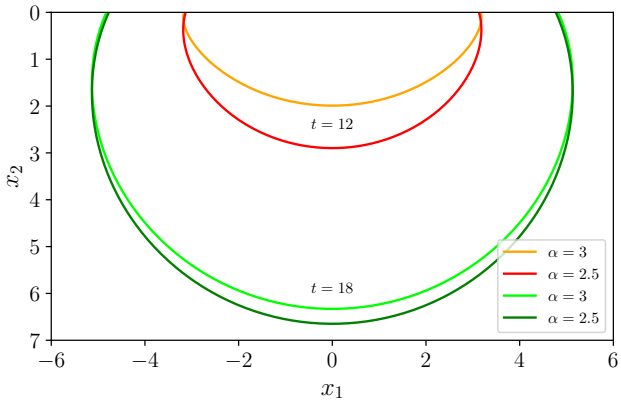


Fig. 6.10: Reflection isochrones for two different common offsets α.

6.1.5 Upwards-shifted phantom function

Here, we shift the phantom function n from (6.1.1) closer towards the surface, specifically

$$\widetilde{n}(x_1, x_2) \coloneqq n(x_1, x_2 + 1).$$

We use the imaging operator $\Lambda^{(2)}_{\mathrm{FN}}$, along with the usual parameters (6.1.2). By doing this, the support of $\widetilde{n}$ is no longer contained in the set $X = \{\boldsymbol{x} \in \mathbb{R}^2_+ : x_2 > x_{\min} \approx 2.07\}$ on which $\Lambda^{(2)}_{\mathrm{FN}}$ is a ΨDO, see beginning of Chapter 4. Consequently, the reconstruction operator is now "only" a FIO, meaning it could potentially add artifacts to the reconstruction. This suspicion is confirmed by the numerical experiment. In Figure 6.11, the reconstruction exhibits a pronounced horizontal artifact at approximately $x_2 = 2.6$, which was not present in $\widetilde{n}$.

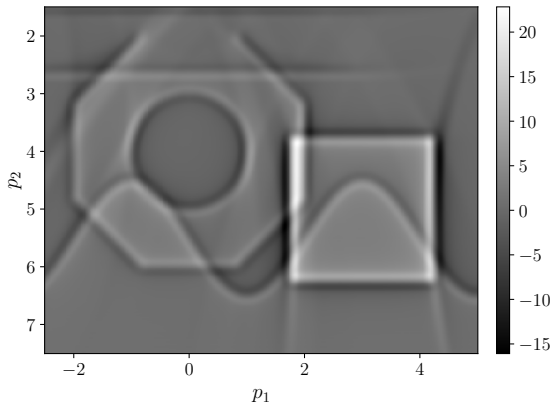


Fig. 6.11: Reconstruction of the upwards-shifted phantom function $\widetilde{n}$.

6.1.6 Smooth parts of the zero-order operators

Concluding the experiments on consistent data, we want to investigate the setting of Corollary 3.7 and Corollary 3.9 in [GQR24] by a numerical experiment. We set $c(\boldsymbol{x}) = 1$ and use the zero-offset geometry (common-offset geometry with $\alpha = 0$).

In this setting, those corollaries state the following

Corollary 6.2. (see [GQR24, Corollaries 3.7 and 3.9])
To any open $X \subset \mathcal{X}$, $\mathrm{Cl}(X)$ compact, and any cone $V = \{\eta \in \mathbb{R}^2 \setminus \{\mathbf{0}\} : |\eta_1| \leq M\,|\eta_2|\}$, $M > 0$, there are an open $S \subset \mathbb{R}$, $\mathrm{Cl}(S)$ compact, a cutoff function $\psi \in \mathcal{D}(S)$, and general pseudodifferential operators O_{FN} and O_{K}, such that

$$\Lambda^{(1)}_{\mathrm{FN}} u = u + O_{\mathrm{FN}}\, u, \qquad \text{and} \qquad \Lambda^{(1)}_{\mathrm{K}} u = u + O_{\mathrm{K}}\, u,$$

for all $u \in \mathcal{E}'(X)$ with $\mathrm{WF}(u) \subset X \times V$. For those u, $O_{\mathrm{FN}} u$ and $O_{\mathrm{K}} u$ are one degree smoother in Sobolev scale than u.

Our goal in this experiment is to calculate the reconstructions $\Lambda^{(1)}_{\mathrm{FN}} n$ and $\Lambda^{(1)}_{\mathrm{K}} n$, and subtract the phantom function n afterwards, to obtain approximations of $O_{\mathrm{FN}}\, n$ and $O_{\mathrm{K}}\, n$, respectively. Since our reconstruction scheme does not calculate Λn, $\Lambda = \Lambda^{(1)}_{\mathrm{FN}}$ or $\Lambda = \Lambda^{(1)}_{\mathrm{K}}$, but instead the approximation $\Lambda n * e^k_\gamma$, the best we can do is to choose a small value for γ. We therefore expect the resulting reconstruction $\Lambda n * e^k_\gamma$ to not have a perfectly sharp edge at the singular support of n, meaning the difference $(\Lambda n * e^k_\gamma) - n$ will not be smooth there.

Here, we choose a simple triangle and a half space with the sine boundary as phantom functions, see also Figure 6.12:

$$n_{sin} \coloneqq \chi_{\left\{\boldsymbol{x} \in \mathbb{R}^2_+ \,:\, x_2 \geq 4 + \sin(x_1)\right\}},$$
$$n_{tri} \coloneqq \chi_{\mathrm{conv}(\{(-2,5),(2,5),(0,3)\})},$$

where conv denotes the convex hull.

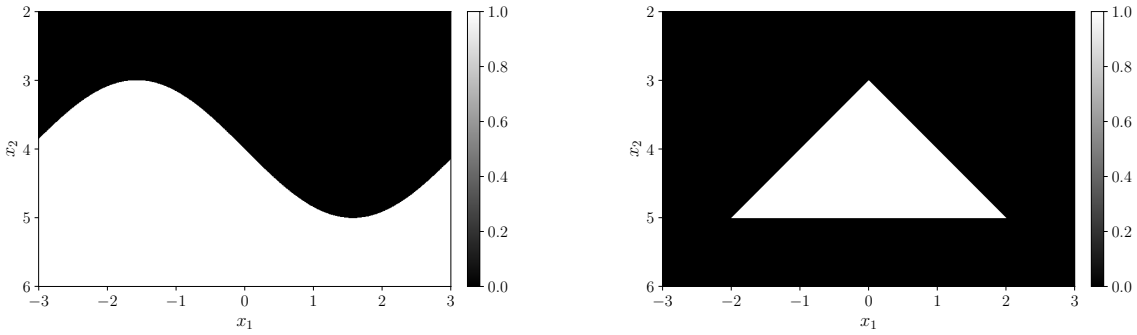


Fig. 6.12: Phantom functions n_{sin} (left) and n_{tri} (right)

The further parameters are as follows

$$\begin{aligned}
\mathcal{M} &= \mathcal{M}\left([0,15]\times[0,15],(6001,6001)^\top\right) && (\boldsymbol{h}_{\mathcal{M}} = (0.0025,0.0025)^\top),\\
\mathcal{M}_{\boldsymbol{p}} &= \mathcal{M}\left([-3,3]\times[2,6],(1201,801)^\top\right) && (\boldsymbol{h}_{\mathcal{M}_{\boldsymbol{p}}} = (0.005,0.005)^\top),
\end{aligned}$$

$$\begin{aligned}
\mathcal{S} = \mathcal{S}_{\text{data}} &= \mathcal{I}([-10,10],5401) && (h_{\mathcal{S}} = 0.005),\\
\mathcal{T} = \mathcal{T}_{\text{data}} &= \mathcal{I}([2.05,29.05],5401) && (h_{\mathcal{T}} = 0.005),\\
P_{\text{ref}} &= \mathcal{I}([2,6],401) && (h_{P_{\text{ref}}} = 0.01),
\end{aligned}$$

$c(\boldsymbol{x}) = 1$

common-offset geometry with $\alpha = 0$,

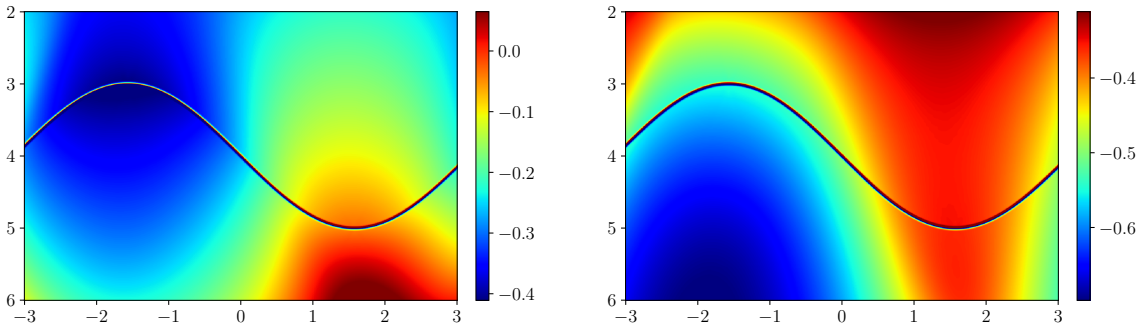
mollifier: $\gamma = 0.05,\ k = 3$,

fast sweeping tolerance $\delta = 0.001$,

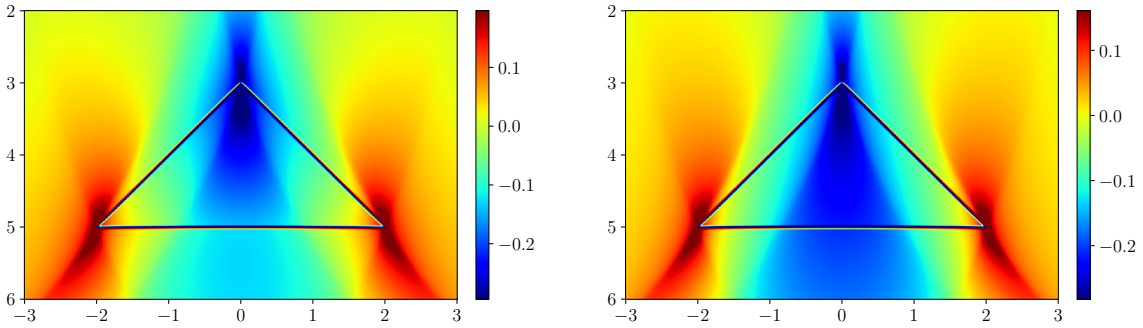
cutoff ψ: $\underline{L}_s = -10, \overline{L}_s = -9.5,\quad \underline{R}_s = 9.5, \overline{R}_s = 10,$
$\underline{L}_t = 0, \overline{L}_t = 0,\quad \underline{R}_s = 28.55, \overline{R}_s = 29.05.$

In Figure 6.13, we show the "smooth part" $(\Lambda n * e_\gamma^k) - n$ for $\Lambda = \Lambda_{\text{FN}}^{(1)}$ and $\Lambda = \Lambda_{\text{K}}^{(1)}$, and the phantom functions $n = n_{sin}$ and $n = n_{tri}$. As mentioned earlier, the approximate inverse gives us a smoothened reconstruction, therefore the boundaries of the inclusion (triangle or sine-halfspace) are still distinctly visible in the difference $(\Lambda n * e_\gamma^k) - n$. For $\gamma \to 0$ this outline will get smaller and smaller, and would theoretically dissapear in the limit, but this is not possible in practice as this would require the spacing to go to zero as well and with that the runtime to infinity. Since the values at this boundary are very large compared to the rest, we clip all values below the 1-percentile or above the 99-percentile, otherwise the remaining structures would not be recognizable. Aside from that, the remaining pictures are very smooth overall, except on the corners of the triangle. This

is expected though, as the wavefront set at those points does contain horizontal normal directions, and therefore does not fulfill the assumptions of Corollary 6.2.



(a) $(\Lambda^{(1)}_{\mathrm{FN}} n_{sin} * e^3_{0.05}) - n_{sin}$ (left) and $(\Lambda^{(1)}_{\mathrm{K}} n_{sin} * e^3_{0.05}) - n_{sin}$ (right)



(b) $(\Lambda^{(1)}_{\mathrm{FN}} n_{tri} * e^3_{0.05}) - n_{tri}$ (left) and $(\Lambda^{(1)}_{\mathrm{K}} n_{tri} * e^3_{0.05}) - n_{sin}$ (right)

Fig. 6.13: Smooth parts $O_{\mathrm{FN}}\, n$ and $O_{\mathrm{FN}}\, n$ for $n = n_{sin}$ and $n = n_{tri}$. Values are clipped to the 1- and 99-percentiles to make the remaining structure visible.

Further, in Figure 6.14 we can observe the difference of the reconstructions from the operators $\Lambda^{(1)}_{\mathrm{K}}$ and $\Lambda^{(1)}_{\mathrm{FN}}$ for both the sine-halfspace and the triangle example.

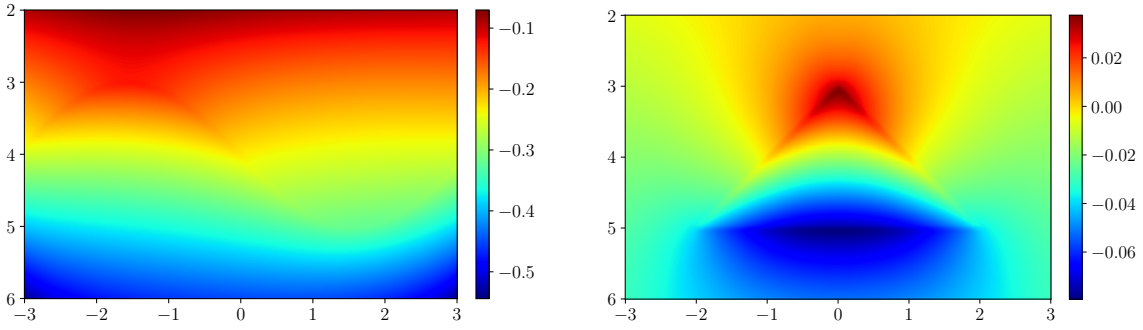


Fig. 6.14: Pointwise difference of the reconstructions: $(\Lambda^{(1)}_{\mathrm{K}} n_{sin} * e^3_{0.05}) - (\Lambda^{(1)}_{\mathrm{FN}} n_{sin} * e^3_{0.05})$ (left) and $(\Lambda^{(1)}_{\mathrm{K}} n_{tri} * e^3_{0.05}) - (\Lambda^{(1)}_{\mathrm{FN}} n_{tri} * e^3_{0.05})$ (right).

6.2 Data from the wave equation

Here, we generate data by using a numerical solver for the wave equation (2.2.1), namely the forward solver of the open source toolbox PySIT ([DH]). To use this solver, we first have to replace the Dirac pulse in the wave equation (2.2.1) with a corresponding source function $f(t)$, which models a seismic source. The wave equation then reads

$$\begin{aligned} \frac{1}{\nu_{\mathrm{pr}}^2}\partial_t^2 u - \Delta_{\boldsymbol{x}} u &= \delta(\boldsymbol{x}-\boldsymbol{x_s})f(t), \qquad \boldsymbol{x}\in\mathbb{R}^2_+,\ t>0, \\ u(0;\cdot,\boldsymbol{x_s}) &= \partial_t u(0;\cdot,\boldsymbol{x_s}) = 0. \end{aligned} \tag{6.2.1}$$

We use a *Gaussian pulse* with peak frequency $\mu > 0$ and time shift $t_0 > 0$ to model an approximated Dirac pulse

$$f(t) = \mathrm{e}^{-\left(\frac{t-t_0}{\sqrt{2}\sigma}\right)^2} = \mathrm{e}^{-((t-t_0)\pi\mu)^2}, \quad \text{where} \quad \sigma = \frac{1}{\pi\mu\sqrt{2}}.$$

The time shift is needed to capture the whole signal, as the signal extends to the negative time axis and would be missing otherwise. This time shift has to be reversed after computing the seismograms. We used the parameters $t_0 = 0.2$ and $\mu = 200$ in our experiments.

As in the consistent data case, we choose the reflectivity n from (6.1.1) and set the pressure velocity ν_{pr} for the wave solver in view of (2.2.2) as

$$\nu_{\mathrm{pr}}^2(\boldsymbol{x}) = \frac{c^2(\boldsymbol{x})}{1+\lambda n(\boldsymbol{x})},$$

where the parameter $\lambda > 0$ is needed to scale the comitted linearization error.

Then the forward solver for the wave equation (6.2.1) can be used with the correct measurement geometry, and the inputs v_{pr} and c to generate the corresponding seismograms $u(t,\boldsymbol{x_r}(s),\boldsymbol{x_s}(s))$ and $\widetilde{u}(t,\boldsymbol{x_r}(s),\boldsymbol{x_s}(s))$ for $s\in\mathcal{S}_{\mathrm{data}} = \mathcal{I}([s_{\min},s_{\max}],n_{\mathcal{S}_{\mathrm{data}}})$, $t\in\mathcal{T}_{\mathrm{data}}$, where the choice of $\mathcal{T}_{\mathrm{data}}$ will be explained in the following. Note that the time interval the wave solver is actually provided with starts at zero, meaning t-range $= [0,t_{\max}]$. Since the time step $h_{\mathcal{T}_{\mathrm{data}}}$ is determined by the wave solver to satisfy its CFL-condition, it may not align with the $t_{\min}$ and $t_{\max}$ values. The seismograms are therefore cropped to $\mathcal{T}_{\mathrm{data}} = \mathcal{I}([\,\overline{t_{\min}},\underline{t_{\max}}\,],n_{\mathcal{T}_{\mathrm{data}}})$, where $n_{\mathcal{T}_{\mathrm{data}}} = \frac{t_{\max}-\overline{t_{\min}}}{h_{\mathcal{T}_{\mathrm{data}}}}+1$. Here, $\overline{t_{\min}}$ is the smallest time in the array generated by the wave solver that is bigger or equal to $t_{\min}$, and $\underline{t_{\max}}$ is set accordingly:

$$\begin{aligned} \overline{t_{\min}} &:= \min\left\{i\cdot h_{\mathcal{T}_{\mathrm{data}}} : i\cdot h_{\mathcal{T}_{\mathrm{data}}} \geq t_{\min}, i\in\mathbb{N}_0\right\}, \\ \underline{t_{\max}} &:= \max\left\{i\cdot h_{\mathcal{T}_{\mathrm{data}}} : i\cdot h_{\mathcal{T}_{\mathrm{data}}} \leq t_{\max}, i\in\mathbb{N}_0\right\}. \end{aligned}$$

Further, $h_{\mathcal{T}_{\text{data}}}$ depends on the maximum input-velocity, which means that $n_{\mathcal{T}_{\text{data}}}$ is most likely larger for u than $\widetilde{u}$. Therefore we configure the wave solver to use the smaller of both time steps for both the actual velocity ν_{pr} and the background velocity c to ensure the same discrete time interval. Another solution would be to perform piecewise linear interpolation of $\widetilde{u}$ on the time grid of u, or any other discrete time interval, provided no information is lost. In our experiments though, the first method lead to better results.

The right-hand side g is then finally obtained by equation (2.2.13) and a trapezoidal rule for the integral therein.

Let $\mathcal{M}_{\text{data}}$ be the mesh on which the wave solver is operating on. It needs to be big enough so that all the different source and receiver configurations are inside it. Since we do not want the wavefields to reflect on any boundaries of this domain, we use a *Perfectly Matched Layer* (PML) on each side of the mesh, see Figure 6.15. PMLs dampen the wave at the boundary of the domain through artificial absorbing boundary conditions. Thus, we can simulate an endless domain regarding the behavior of the wavefields. For more details, see [Ber94].

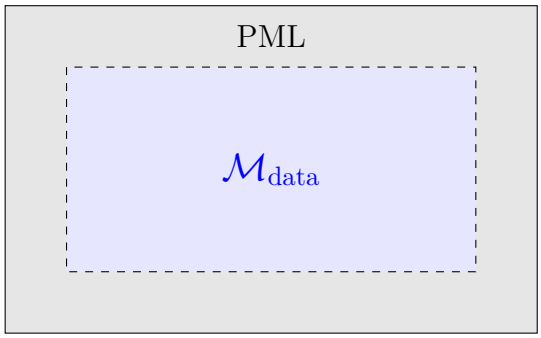


Fig. 6.15: The mesh of the wave solver, surrounded by a Perfectly Matched Layer.

The setup for the kernel and reconstruction calculation stays the same as in (6.1.2). For the data generation we use

$\mathcal{M}_{\text{data}} = \mathcal{M}\left([-15, 20] \times [0, 15], (841, 361)^\top\right)$ $\quad (\boldsymbol{h}_{\mathcal{M}} \approx (0.0416, 0.0416)^\top)$,

PML width $= 3$,

t-range $= [0, 42.64]$, $\quad \mathcal{T}_{\text{data}} = \mathcal{I}([\overline{17.64}, \underline{42.64}], n_{\mathcal{T}_{\text{data}}})$,

$\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 1001)$ $\quad (h_{\mathcal{S}} = 0.025)$,

$\lambda = 0.05$.

On the left of Figure 6.16 we see the processed data $\psi|_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}\, g_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}$ from the wave solver. Note that there is a decrease at about $t \approx 29$. This is due to reflections at the bottom of the computational domain of the wave solver, even though PML boundary

conditions were applied. The exact time of a wavefront reflected at the bottom of the domain to travel from source to receiver in this setting is 29.012. An obvious solution to this problem would be to increase the depth of the computational domain, but would result in much more computing time for the wave solver. Another option is to restrict the time interval for example to $[17.64, 29]$, but then we possibly lose some singularities of n in the reconstruction. As this problem did not impact our reconstructions, we did not use any of the mentioned solutions.

The reconstruction using $\Lambda_{\mathrm{FN}}^{(2)}$ can be seen on the right of Figure 6.16. The image clearly shows the singular support of n, and is qualitatively comparable to the consistent data case in Figure 6.6 (b) (left).

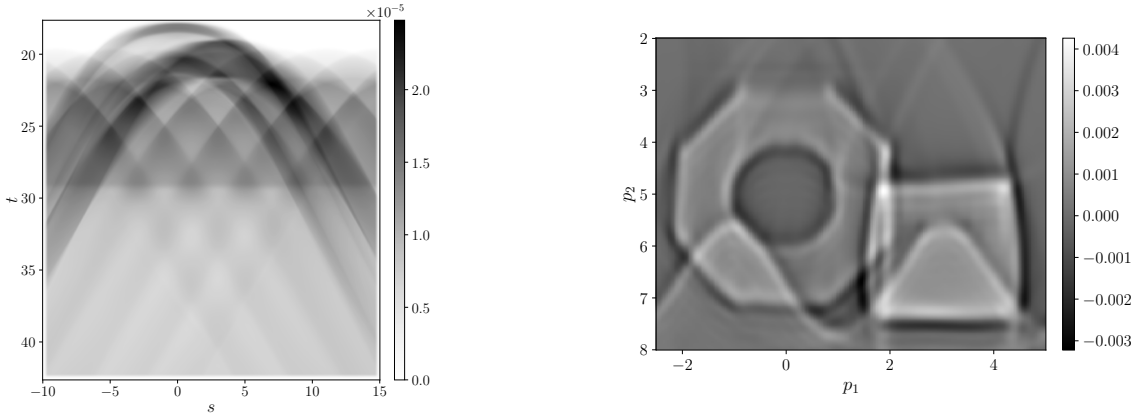


Fig. 6.16: Left: processed data $\psi|_{\mathcal{S}_{\mathrm{data}}\times\mathcal{T}_{\mathrm{data}}}\, g_{\mathcal{S}_{\mathrm{data}}\times\mathcal{T}_{\mathrm{data}}}$. Right: reconstruction using $\Lambda_{\mathrm{FN}}^{(2)}$ with $\mathcal{T}_{\mathrm{data}} = \mathcal{I}([17.64, 42.637], 7200)$ and $\lambda = 0.05$.

The impact of a larger linearization error can be observed in Figure 6.17. As the value of λ increases, the quality of the reconstructions declines.

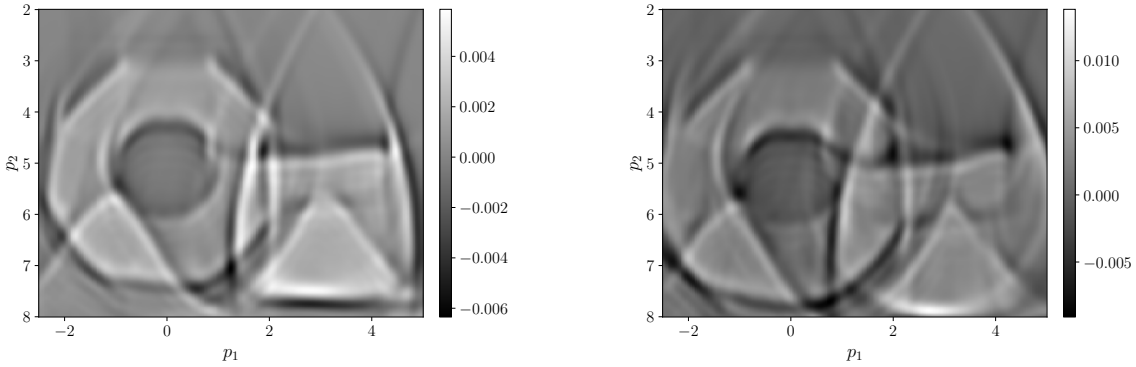


Fig. 6.17: Reconstructions using $\Lambda_{\mathrm{FN}}^{(2)}$ with $\mathcal{T}_{\mathrm{data}} = \mathcal{I}([17.64, 42.637], 7200)$ and $\lambda = 0.1$ (left) or $\lambda = 0.2$ (right).

6.2.1 Low resolution data

In practice, we may not have access to thousands of source and receiver pairs (parameters $s \in \mathcal{S}_{\text{data}}$). We will now show that even in an extreme case with just 101 parameters in $\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 101)$, we can still get good results. In this setting, we also show that it is advantageous to calculate the reconstruction kernels with higher resolution than the data. Even though the reconstruction on the left in Figure 6.18 clearly shows the phantom function, it is etremely pixelated. Here, we used $\mathcal{S} = \mathcal{S}_{\text{data}}$. Note that this is purely the effect of the low resolution data and kernels, since the resolution of the reconstruction mesh $\mathcal{M}_{\boldsymbol{p}}$ is still unchanged (751×601 points). The reconstruction on the right of Figure 6.18 using finer reconstruction kernels ($\mathcal{S} = \mathcal{I}([-10, 15], 1001)$) clearly results in a much better image. The low resolution of the data still introduces some high frequency artifacts, but it is a massive uplift in quality in this comparison nevertheless. By increasing γ we could mitigate those artifacts further, at the cost of blurring the image.

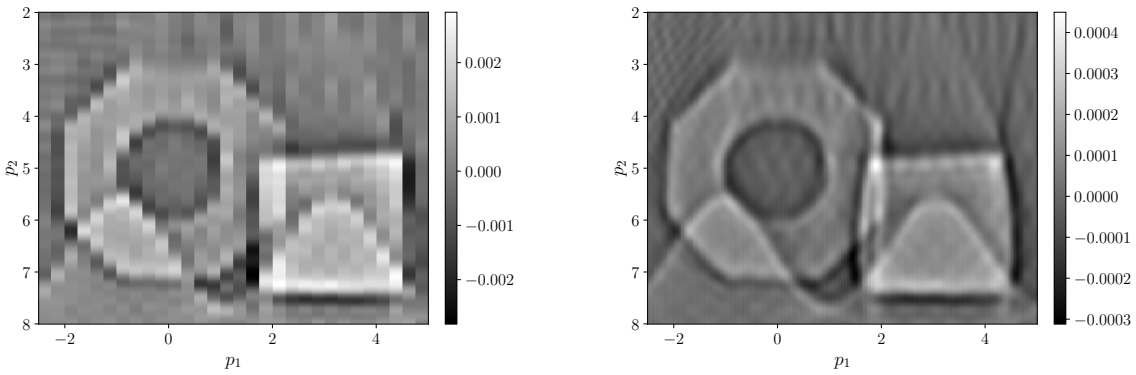


Fig. 6.18: Low resolution inconsistent data reconstructions using $\Lambda_{\text{FN}}^{(2)}$ and different resolutions of the reconstruction kernels. Left: $\mathcal{S} = \mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 101)$, right: $\mathcal{S} = \mathcal{I}([-10, 15], 1001)$.

6.2.2 Realistic measurement scenario

As an intermediate step between the last experiment and actual field data, we test the following setup. This time, we use the common-source geometry (recall Section 5.7.2) and noised source and receiver positions for a realistic measurement scenario that to some extent mimics an actual field data acquisition. To this end, we again generate data by using the forward solver for the wave equation, where the source and receiver positions are uniformly perturbed within the lower half-circle with radius ε_{pos} around their original positions for each ensemble, see Figure 6.19. The radius of this half-circle will be denoted by ε_{pos}. This is done to emulate some plausible alignment errors in the source and receiver positions. Here, one ensemble describes the measurement of a single source excitation,

or rather the involved parameters of the source position $\boldsymbol{x}_{\text{cs}}$ and the parameters $\mathcal{S}_{\text{data}}$, together with the measurement times $\mathcal{T}_{\text{data}}$. This collection of parameters in the case of the common-source geometry is more commonly known as one *shot*. Additionally, the taken measurements, or more precicely, the right-hand side g of (2.2.13), are also perturbed with artificial noise:

$$g_k^\varepsilon = g_k + \varepsilon \left\|g_k\right\|_F \frac{\mathrm{N_k}}{\left\|\mathrm{N_k}\right\|_F}, \qquad \varepsilon > 0, k \in \{1, \dots, n_{shots}\},$$

where g_k are the data (integrated seismogram) of the kth shot, $\mathrm{N_k}$ is an $n_{\mathcal{S}_{\text{data}}} \times n_{\mathcal{T}_{\text{data}}}$ array of uniformly distributed random numbers in the interval $[-1, 1]$ and $\left\|\cdot\right\|_F$ denotes the Frobenius norm. We have that

$$\frac{\left\|g_k - g_k^\varepsilon\right\|_F}{\left\|g_k\right\|_F} \leq \varepsilon, \quad k \in \{1, \dots, n_{shots}\},$$

meaning ε measures the *relative noise*. The parameters will differ significantly from the usual ones, as we now work with the common-source acquisition geometry:

$c(\boldsymbol{x}) = 0.5 + 0.1x_2$,
$\mathcal{M}_{\text{data}} = \mathcal{M}\left([-7, 9] \times [0, 10], (641, 401)^\top\right)$,
PML width $= 3$,
ensembles (shots):

$\boldsymbol{x}_{\text{cs}} = 0$,	$\mathcal{S}_{\text{data}} = \mathcal{I}([-5, 5], 41)$,	$\mathcal{S} = \mathcal{I}([-5, 5], 301)$,
	t-range $= [0, 25.5]$,	$\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([4.5, 25.497], 7560)$,
$\boldsymbol{x}_{\text{cs}} = 8$,	$\mathcal{S}_{\text{data}} = \mathcal{I}([-13, -3], 41)$,	$\mathcal{S} = \mathcal{I}([-13, -3], 301)$,
	t-range $= [0, 30]$,	$\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([9.5, 29.997], 7380)$,
$\boldsymbol{x}_{\text{cs}} = -5$,	$\mathcal{S}_{\text{data}} = \mathcal{I}([0, 10], 41)$,	$\mathcal{S} = \mathcal{I}([0, 10], 301)$,
	t-range $= [0, 28]$,	$\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([7, 27.997], 7560)$

positional noise radius $\varepsilon_{pos} = 0.1$,
relative noise $\varepsilon = 5\%$,
$\mathcal{M} = \mathcal{M}\left([0, 17] \times [0, 29], (2041, 3481)^\top\right)$,
$\mathcal{M}_{\boldsymbol{p}} = \mathcal{M}\left([-2.5, 5] \times [2, 8], (751, 601)^\top\right)$,
$P_{\text{ref}} = \mathcal{I}([2, 8], 101)$, $\quad n_{\alpha_{cs}} = 51$ (number of common offsets in the set α_{cs}),
mollifier: $\gamma = 0.2,\ k = 3$,
fast sweeping tolerance $\delta = 0.005$,
cutoff $\quad \psi_{rel}(s, t;\ 0, 0.5,\ -0.5, 0;\ 0, 0.1,\ -0.5, 0)$

In Figure 6.19 we display the scanning geometry, including the positional displacements, and the processed data $(g_2^\varepsilon)_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}$ for the second ensemble/shot with source position $\boldsymbol{x_s} = (8, 0)^\top$. We can observe some vertical stripes in $(g_2^\varepsilon)_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}$. These are caused by adjacent receivers having a relatively large vertical distance, which results in a difference of the amplitude of an incoming reflected wave.

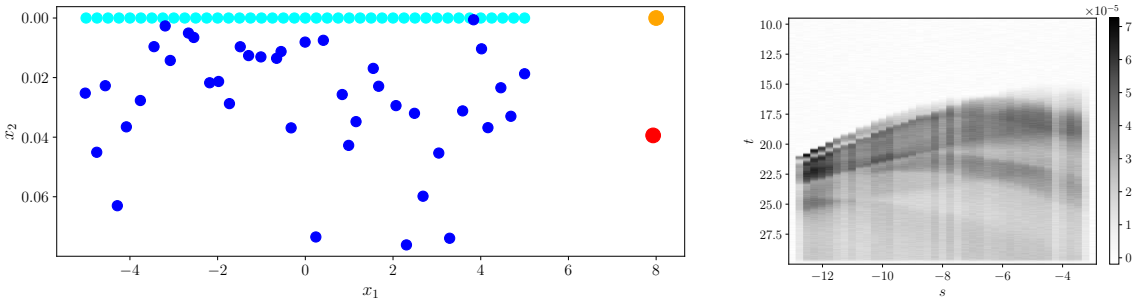
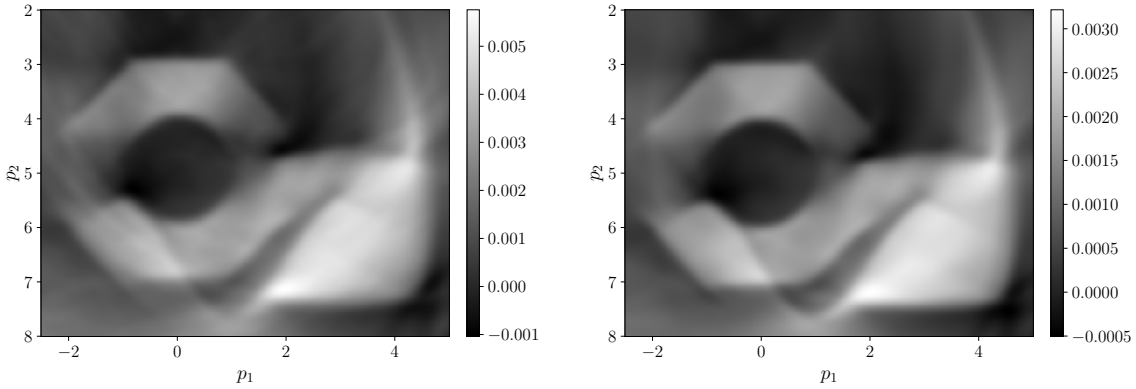
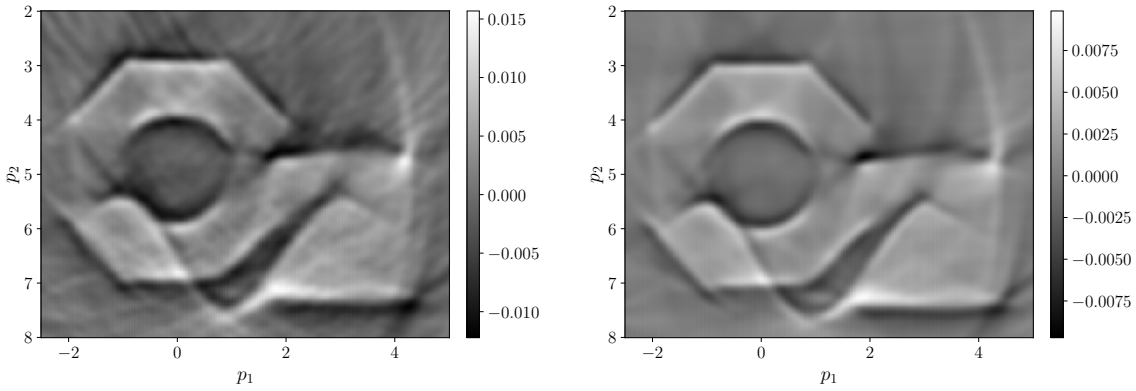


Fig. 6.19: Left: scanning geometry for the second shot with $\boldsymbol{x_s} = (8, 0)^\top$ and $\varepsilon_{pos} = 0.1$. The cyan spheres show the exact and the blue spheres the perturbed receiver positions. Similarly, the orange sphere is the exact and the red sphere the perturbed source position. Note that the scale of the coordinate axes vastly differ.
Right: processed data $g^\varepsilon_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}$ for this shot.

The resulting reconstructions using the operators $\Lambda_{\text{K}}^{(1)}$ and $\Lambda_{\text{FN}}^{(2)}$ are presented in Figure 6.20. The left column shows the reconstruction with perturbed source and receiver positions and noised data, while the right column displays reference reconstructions using the same operators, but from data without any added noise or perturbed positions. Clearly, we can observe by this comparison, that the reconstruction is very stable with respect to noise of this kind.



(a) Reconstruction operator $\Lambda_{\mathrm{K}}^{(1)}$. Left: with noise, right: no noise.



(b) Reconstruction operator $\Lambda_{\mathrm{FN}}^{(2)}$. Left: with noise, right: no noise.

Fig. 6.20: Reconstructions from data with relative data noise and perturbed source/receiver positions, and corresponding reference reconstructions where no noise or positional perturbation was added.

6.3 Venezuela field data

The data utilized in this experiment originates from a seismic project summarized as "Broadband Ocean-Land Investigation of Venezuela and the Antilles arc Region". Some general information can be found at https://doi.org/10.7284/901238.

In the timespan from 18th of April 2004 to 3rd of June 2004 the research vessel and its crew followed the route in Figure 6.21 (left) and recorded data along the lines in Figure 6.21 (right). The whole data set, along with in-depth documents of the cruise, is available on [Mar].

For our experiment, we exclusively used the data from line BOL-11. This prestack data set was preprocessed by the Geophysical Institute (GPI) of the KIT: A portion of the prestack data was cut off to remove a segment where the streamer was bent. The raw data underwent denoising and demultiple steps, including band-pass filters, swell

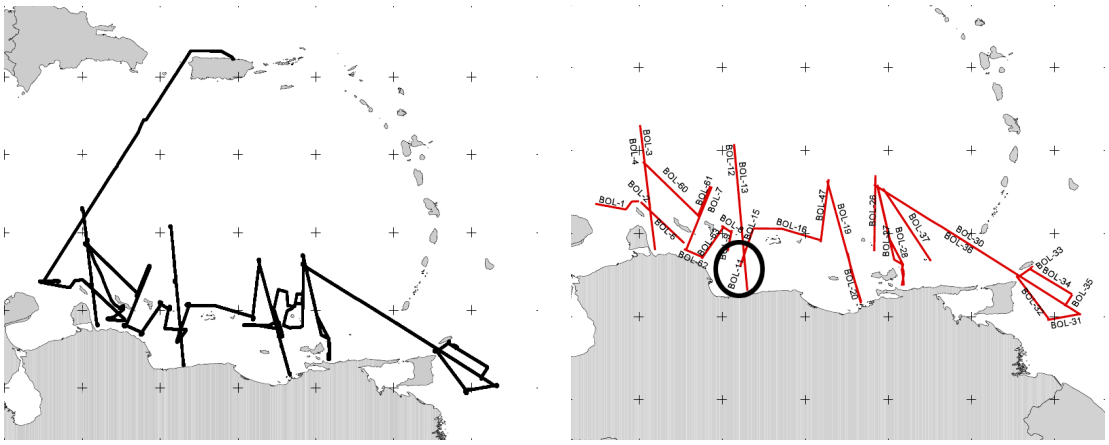


Fig. 6.21: Cruise ship route for the exploration project (left) and the lines, where seismic data were measured (right). Line BOL-11 is marked by the circle. The images are taken from the science report under the Documents section on [Mar].

noise removal, surface-related mutliple elimination and parabolic Radon demultiple. For general information on these processes, we refer the reader to [Yil01].

We were also provided with a reference migration result, which is time migrated and stretched to depth using standard time-to-depth conversion based on the given velocity model. While this is not equivalent to true depth migration, it should still allow for a comparison with a proper depth migration result.

Data summary

The seismograms were originally recorded in the common-source geometry, but subsequently saved in a data format where each trace from every shot is dumped into a single large list along with its trace information. A trace is the recorded signal of a single receiver. To perform a reconstruction using the common-offset geometry, we need to group those traces by their common offset first and then sort each group by the original ensemble number. Hereby the grouping allows for a small leeway of the grouping parameter, so that two traces with minor deviations in their grouping parameter are still considered in the same group. This process, known as *data binning*, is sensible because for field data there are often small deviations from the planned, perfectly regular geometry. Now that the raw data are expressed as multiple ensembles in common-offset form, we summarize important parameters in Table 6.1.

For the geophysicists: please note that our terminology differs from the one typically used in geophysics. The term “common offset”, which in geophysics is often referred to simply as “offset”, here only describes half the distance between source and receiver, see Section 5.7. The parameter s represents the (x_1-) position of the midpoint between a source and a receiver.

Table 6.1: Data parameters

number of ensembles	480
traces in each ensemble	701
sources/receivers depth	7 m below sea surface
average water depth	1890 m
common offsets	89.5 m (ensemble 0) - 3083.5 m (ensemle 479)
time interval (in seconds)	$\mathcal{T}_{\text{data}} = \mathcal{I}([0,9], 2251) \quad (h_{\mathcal{T}_{\text{data}}} = 0.004)$
midpoint parameters (in m)	$s_{\min} = 9593.74$, $s_{\max} = 44592.77$ (ensemble 0) $\vdots$ $s_{\min} = 6599.77$, $s_{\max} = 41598.8$ (ensemble 479)

The discrete parameters $s \in [s_{\min}, s_{\max}]$ for each ensemble are not quite equidistant, with the distance of neighboring s-values varying between approximately 47 m and 52.5 m. This is a problem, since the slicing operations of the reference kernels outlined in Section 5.5 would not align with the $\mathcal{S}$ grid. Consequently, the implementation of the inversion scheme requires equidistant $\mathcal{S}$. Additionally, the simplified trapezoidal rule in equation (5.6.1) relies on equidistant parameters $\mathcal{S}$, which could, however, be easily fixed by using the standard trapezoidal rule. In summary, we have to interpolate the seismograms for each ensemble to uniformly spaced parameters $\mathcal{S}_{\text{data}} = \mathcal{I}([s_{\min}, s_{\max}], 701)$.

Reconstruction summary

We denote this resampled array of each ensemble by $(seismo_k)_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}$, $k = 0, \ldots, 479$, and set $values_k[i,j] \coloneqq (seismo_k)_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}[i,j]$ for $i = 0, \ldots, 700$, $j = 0, 2250$. The seismograms in our data set already have their first arrival signal removed, therefore a reference solution $\widetilde{u}$ in equation (2.2.13) is not required. The approximation to the right-hand side for ensemble k then reads ($i = 0, \ldots, 700$ and $j = 0, \ldots, 2250$)

$$(g_k)_{\mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}}}[i,j] = -2\pi h_{\mathcal{T}_{\text{data}}} \sum_{l=1}^{j} \left(values_k[i,l] + values_k[i,l-1]\right).$$

Since we only consider depth-dependent velocities c, it suffices to specify the contour line $c_0 \coloneqq c\left((0,\cdot)^\top\right)$. The base contour $\widetilde{c_0}$ consists of a constant water layer with the known velocity $c_{\text{water}} = 1500\,\frac{m}{s}$ and an affine linear part below the sea depth:

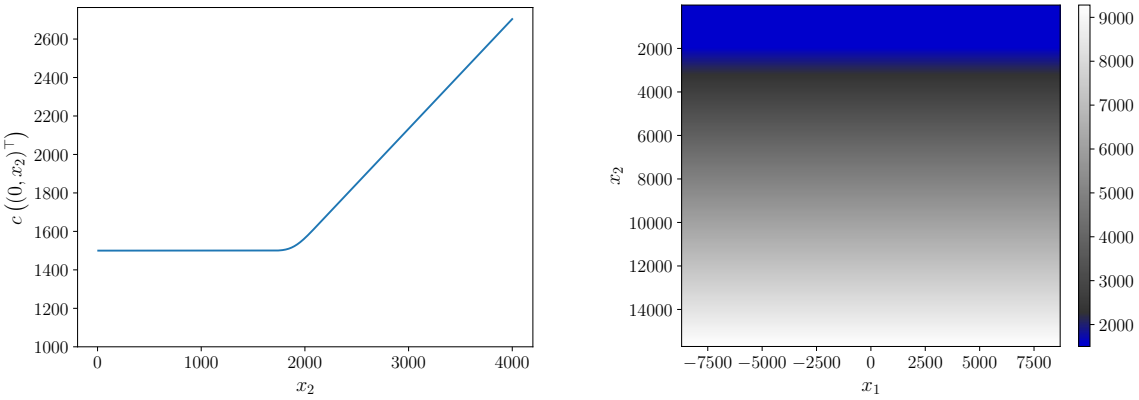
$$\widetilde{c_0}(y) \coloneqq \begin{cases} 1500, & y \leq 1890, \\ 1500 + 0.57071358\,(y - 1890), & y > 1890. \end{cases}$$

The slope 0.57071358 was chosen by a least squares approximation to fit the horizontal average of a provided initial model. Since c_0 is not differentiable in $y = 1890$ we smoothen this kink by convolving $\widetilde{c_0}$ with a Gaussian window

$$\widetilde{w}(y) \coloneqq \mathrm{e}^{-\frac{1}{2}\left(\frac{y}{\sigma}\right)^2}, \qquad w(y) \coloneqq \frac{\widetilde{w}(y)}{\int_{-\infty}^{\infty} \widetilde{w}(z)\,\mathrm{d}z} = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{1}{2}\left(\frac{y}{\sigma}\right)^2}.$$

We use the Python implementations scipy.signal.windows.gaussian[1] and scipy.signal.convolve[2] for the gaussian window and convolution respectively. The c_0-array is appropriately padded with reflecting boundaries before convolution, and sliced back to the original length afterwards so that the boundaries do not impact the result. We therefore obtain, see Figure 6.22

$$c_0(x_2) = (\widetilde{c_0} * w)(x_2), \qquad c(\boldsymbol{x}) = c_0(x_2).$$



(a) Depth contour section c_0 of the background velocity.

(b) Background velocity c.

Fig. 6.22: Background velocity for the venezuela data set.

The remaining reconstruction parameters are summarized in Table 6.2.

The results of this experiment are displayed in Figure 6.23. We also included the migration result from the Geophysical Institute for reference at the top. Note that by using the imaging operator $\Lambda_{\mathrm{FN}}^{(2)}$ of order 1 here, the actual values are not really meaningful. In the middle image, we display the result from a single ensemble, specifically ensemble 0. The reconstruction is somewhat dominated by noise, but one can cleary notice the seabed and some of the internal structures. In 6.23 (c) we aggregated the reconstructions from every ensemble, normalized by the total number of used ensembles (479). Here, the internal

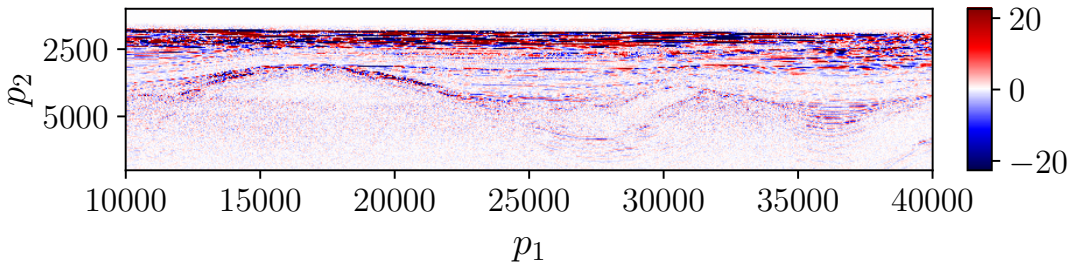
[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.windows.gaussian.html

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve.html

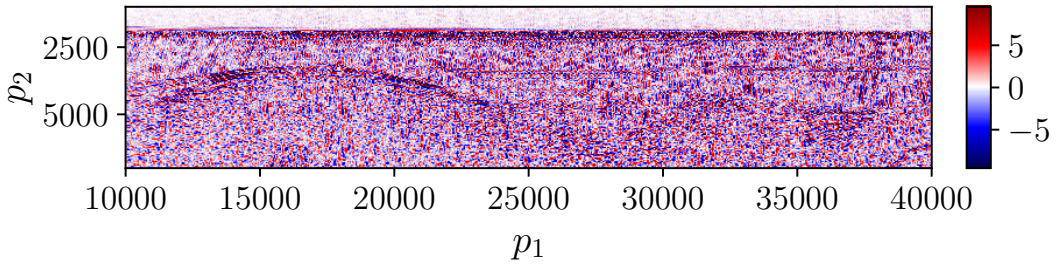
Table 6.2: Reconstruction parameters

geometry	common offset
traveltime/amplitude mesh	$\mathcal{M} = \mathcal{M}([-8700, 8700] \times [7, 15700], (4351, 3924))$ $(\boldsymbol{h}_{\mathcal{M}} = (4, 4.000255))$
fast sweeping tolerance	0.2
reconstruction mesh	$\mathcal{M}_{\boldsymbol{p}} = \mathcal{M}([10000, 40000] \times [1000, 7000], (4801, 1201))$ $(\boldsymbol{h}_{\mathcal{M}_{\boldsymbol{p}}} = (6.25, 5))$
time interval	$\mathcal{T} = \mathcal{T}_{\text{data}}$
midpoint parameters	$\mathcal{S} = \mathcal{I}([9593.74, 44592.77], 1001)$ (ensemble 0) $\vdots$ $\mathcal{S} = \mathcal{I}([6599.77, 41598.8], 1001)$ (ensemble 479)
x_2-interval for the reference kernels	$P_{\text{ref}} = \mathcal{I}([1000, 7000], 601)$
imaging operator	$\Lambda^{(2)}_{\text{FN}} = \frac{1}{2\pi}(-\Delta)F^{\dagger}_{\lvert B\rvert/(A\lVert\nabla_{\boldsymbol{x}}\varphi\rVert)}\psi F$
γ	75
k	3
cutoff function	$\psi_{rel}(s, t;\ 0, 100,\ -100, 0;\ 0, 0,\ 0, 0)$

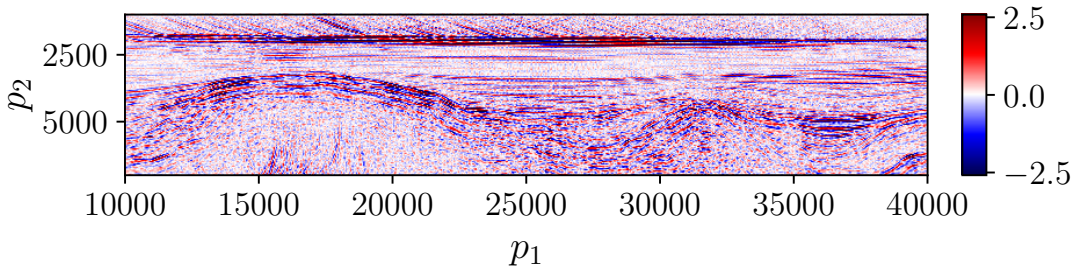
structures are substantially clearer to see and the noise below the seabed is significantly reduced. Although some noise was introduced in the water column, the seabed remains distinctly visible. Variations in the singular points among the different ensembles makes it look like some of the internal structures are duplicated slightly lower or higher. One reason for this could be the background velocity, since we obviously do not have access to the exact one and are limited by only depth-dependent velocities. The comitted linearization error is therefore potentially rather large. Another reason could possibly involve the data-binning. As we have seen in an earlier numerical experiment, the depth at wich features are reconstructed is affected by the common offset used in the reconstruction. Variations of the actual common offset in the different traces, which are inevitably introduced through the data binning process, could therefore explain the appearance of the same structure at slightly different depths over different ensembles.



(a) Reference migration result (GPI).



(b) Reconstruction from ensemble 0.



(c) Mean value of all reconstructions from the corresponding ensembles.

Fig. 6.23: Reference and own reconstructions of the Venezuela data set using $\Lambda_{\mathrm{FN}}^{(2)}$.

CHAPTER 7

Conclusion and outlook

This final chapter provides a recap of the thesis. We also discuss the differences and potential challenges introduced in a three-dimensional setting.

In this work, we presented a numerical scheme for the linearized problem of seismic imaging in two dimensions using the concept of approximate inverse. The setting was derived starting from the acoustic wave equation, resulting in a general form of reconstruction operators suitable for solving this inverse problem. This framework allows for great flexibility in the chosen reconstruction operators, depending on the desired objectives. It encompasses, for instance, the operators from the well-established Kirchhoff migration, which is still heavily used in some form to this day. We covered the necessary aspects of the theoretical background, which is microlocal analysis, that allows us to predict what singularities can be reconstructed and which types of artifacts we expect to appear. We delved into the intricacies of the implementation process, focusing on its efficiency within the context of a layered background velocity and the common-offset acquisition geometry. Key components of the implementation included solvers for the traveltime and amplitude functions. Finally, the conducted numerical experiments on synthetic data provided empirical evidence to support our theoretical predictions. The numerical scheme demonstrated reliability with respect to different types of noise, and further showcased considerable potential for reconstructions from field data.

7.1 A look at the three-dimensional case

Source and receiver positions are now parametrized by $\boldsymbol{s} = (s_1, s_2)^\top \in \mathcal{S} \subset \mathbb{R}^2$. For example, for the common-offset acquisition geometry with offset $\alpha \geq 0$ we have (see, e.g. [Fel+16])

$$\boldsymbol{x_s}(\boldsymbol{s}) = (s_1, s_2 - \alpha, 0)^\top \quad \text{and} \quad \boldsymbol{x_r}(\boldsymbol{s}) = (s_1, s_2 + \alpha_2, 0)^\top.$$

Recall Remark 2.1. In three dimensions, the fundamental solution of the wave equation with constant $c > 0$ is instead given by

$$\widetilde{u}(t; \boldsymbol{x}, \boldsymbol{x_s}) = \frac{\delta(ct - \|\boldsymbol{x} - \boldsymbol{x_s}\|)}{4\pi \|\boldsymbol{x} - \boldsymbol{x}_s\|}.$$

Similar to the two-dimensional case, we can approximate $\widetilde{u}$ by a progressing wave expansion when the background velocity is not constant

$$\widetilde{u}(t; \boldsymbol{x}, \boldsymbol{x_s}) \approx a(\boldsymbol{x}, \boldsymbol{x_s})\delta(t - \tau(\boldsymbol{x}, \boldsymbol{x_s})).$$

Plugging this into (2.2.5) we get (with abbreviations $\boldsymbol{x}_s(s) = \boldsymbol{x_s}$, $\boldsymbol{x_r}(s) = \boldsymbol{x_r}$)

$$\begin{aligned} Ln(s,t) &\approx \partial_t^2 \int_X \frac{n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \int_0^t a(\boldsymbol{x}, \boldsymbol{x_s})\delta(t - \eta - \tau(\boldsymbol{x}, \boldsymbol{x_s}))\; a(\boldsymbol{x_r}, \boldsymbol{x})\delta(\eta - \tau(\boldsymbol{x_r}, \boldsymbol{x}))\, \mathrm{d}\eta\, \mathrm{d}\boldsymbol{x} \\ &= \partial_t^2 \int_X n(\boldsymbol{x}) A(s, \boldsymbol{x})\; \delta(t - \varphi(s, \boldsymbol{x}))\, \mathrm{d}\boldsymbol{x}. \end{aligned}$$

Now, we integrate twice over time and get the same representation of F as in (2.2.12), and the 3D equivalent of the right-hand side (2.2.13) as

$$g(s,t) = \int_0^t \int_0^\zeta (\widetilde{u} - u)(\zeta, \boldsymbol{x_r}, \boldsymbol{x_s})\, \mathrm{d}\zeta\, \mathrm{d}\eta = \int_0^t (t - \eta)(\widetilde{u} - u)(\eta, \boldsymbol{x_r}, \boldsymbol{x_s})\, \mathrm{d}\eta,$$

where we used the Cauchy formula for repeated integration in the last step. The definition of the operator F itself stays the same, but the transport equation (2.2.8) needs some minor adjustments (see [LQB14, equation (8)]):

$$\begin{gathered} 2\nabla_{\boldsymbol{x}} a(\boldsymbol{x}, \boldsymbol{x_s}) \cdot \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_s}) + a(\boldsymbol{x}, \boldsymbol{x_s})\Delta_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_s}) = 0, \\ \lim_{\boldsymbol{x} \to \boldsymbol{x_s}} a(\boldsymbol{x}, \boldsymbol{x_s}) \|\boldsymbol{x} - \boldsymbol{x_s}\| = \frac{1}{4\pi}. \end{gathered}$$

Note that the solution of the transport equation for $c(\boldsymbol{x}) = 1$, and therefore a_0 for the factorization $a = a_0 a_1$ is now given by

$$a_0(\boldsymbol{x}, \boldsymbol{x_s}) = \frac{1}{4\pi \|\boldsymbol{x} - \boldsymbol{x_s}\|}.$$

Accordingly, the initialization for the fast sweeping scheme (Algorithm 3) should therefore be $u_{i,j} = 1$. The fast marching and fast sweeping algorithms for three dimensions have already been incorporated into our code.

For a depth-dependent velocity $c(\boldsymbol{x}) = c(x_3)$, the translation invariances (5.4.1) carry over to their respective three-dimensional versions:

$$\begin{aligned} \tau\left(\boldsymbol{x}, (r_1, r_2, 0)^\top\right) &= \tau\left((x_1 - r_1, x_2 - r_2, x_3)^\top, \mathbf{0}\right), \\ a\left(\boldsymbol{x}, (r_1, r_2, 0)^\top\right) &= a\left((x_1 - r_1, x_2 - r_2, x_3)^\top, \mathbf{0}\right), \end{aligned} \qquad \boldsymbol{x} \in \mathbb{R}^3_+, \quad (r_1, r_2) \in \mathbb{R}^2,$$

$$\mathcal{L}_{(\boldsymbol{s},t)} = \mathcal{L}_{(\mathbf{0},t)} + \begin{pmatrix} s_1 \\ s_2 \\ 0 \end{pmatrix} \qquad \boldsymbol{s} \in \mathcal{S},\ t \in \mathcal{T}.$$

We can therefore use the same concepts as in Section 5.4 to calculate the three-dimensional versions of $\varphi_{\mathcal{M}}$ and $W_{\mathcal{M}}$ with respect to $\boldsymbol{s} = \mathbf{0}$ and then obtain $I_{\mathcal{M}}$. The reflection isochrones can analogously be calculated using the Marching Cubes algorithm. See Figure 7.1 for an example of three-dimensional reflection isochrones.

The remaining calculations for the reference kernels, reconstruction kernels and the evaluation of the scalar product in Chapter 5 can also be extended to three dimensions by adjusting each step involving the parameter s.

One issue that may arise is the memory required to save the 3D reference kernels. It might not be feasible to store the reference kernels in memory without some form of compression or memory efficient representation for finer reconstructions.

The reconstruction operators for the three-dimensional case suggested in [GQR24, Chapter 6] are as follows. For the filtered normal operators, we have

$$\Lambda^{(1)}_{\mathrm{FN}} = \frac{1}{4\pi^2}(-\Delta)F^\dagger_{|B|/A\|\nabla_{\boldsymbol{x}}\varphi\|^2}\psi F \qquad \text{and} \qquad \Lambda^{(2)}_{\mathrm{FN}} = \frac{1}{4\pi^2}\partial_{x_3}(-\Delta)F^\dagger_{|B|/A\|\nabla_{\boldsymbol{x}}\varphi\|^2}\psi F.$$

These are again of order 0 and 1, respectively.

Note that the normalization factor of the mollifier e^k_γ from (4.0.1) differs in three dimensions, namely

$$C_{k,\gamma} = \left(4\pi \int_0^\gamma (\gamma^2 - r^2)^k r^2 \,\mathrm{d}r\right)^{-1} = \frac{\Gamma(k + 5/2)}{\pi^{3/2}\,\gamma^{2k+3}\,\Gamma(k+1)}.$$

The mollifier evaluation $K^* e^k_\gamma$ for an explicit expression does also have to be done again. Here, we have

$$(-\Delta)e^k_\gamma(\boldsymbol{x}) = C_{k,\gamma}\begin{cases} 2k\left(\gamma^2 - \|x\|^2\right)^{k-2}\left(3\gamma^2 - (2k+1)\|x\|^2\right), & \|\boldsymbol{x}\| < \gamma, \\ 0, & \|\boldsymbol{x}\| \geq \gamma, \end{cases}$$

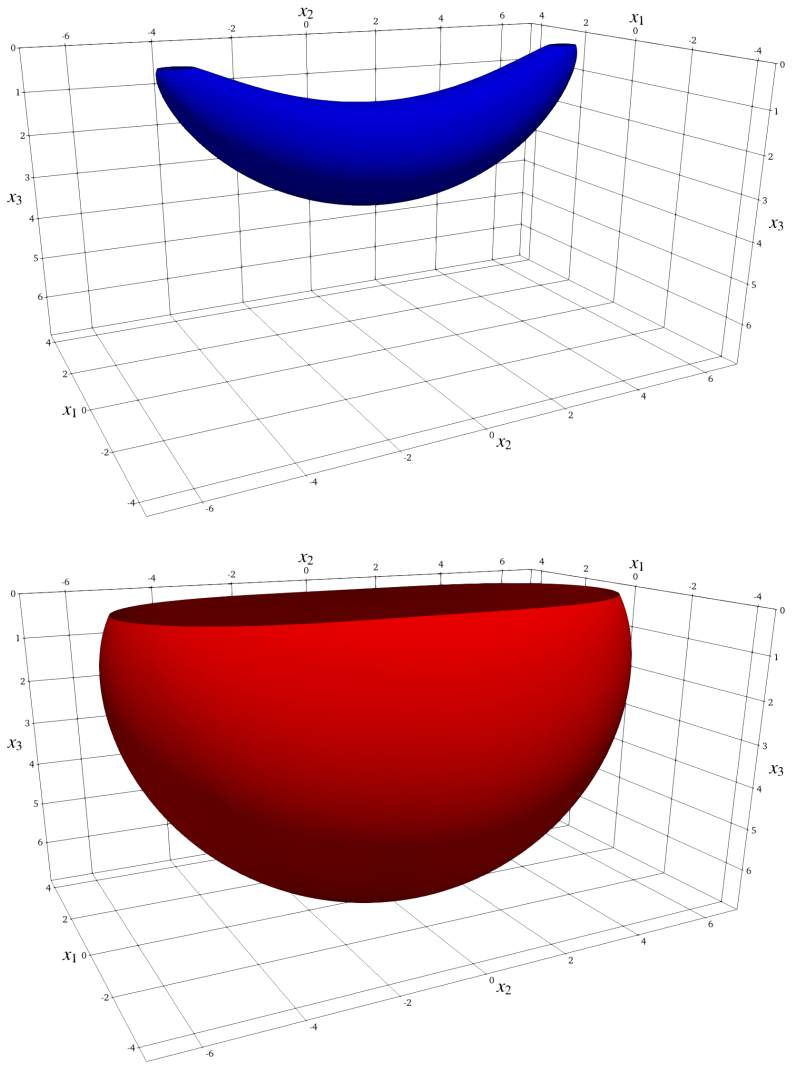


Fig. 7.1: Three-dimensional reflection isochrones $\mathcal{L}_{(\mathbf{0},18)}$ (top) and $\mathcal{L}_{(\mathbf{0},21)}$ (bottom) for the background velocity $c(\boldsymbol{x}) = 0.5 + 0.1x_3$.

which is the explicit expression needed for $\Lambda^{(1)}_{\mathrm{FN}}$. Accordingly, for $K^* = \Delta\partial_{x_3}$ in $\Lambda^{(2)}_{\mathrm{FN}}$, we obtain

$$\Delta\partial_{x_3} e^k_\gamma(\boldsymbol{x}) = C_{k,\gamma} \begin{cases} 4k(k+1)x_3 \left(\gamma^2 - \|x\|^2\right)^{k-3} \left(5\gamma^2 - (2k+1)\|\boldsymbol{x}\|^2\right), & \|\boldsymbol{x}\| < \gamma, \\ 0, & \|\boldsymbol{x}\| \geq \gamma. \end{cases}$$

The three-dimensional Kirchhoff operators are given by

$$\Lambda^{(1)}_{\mathrm{K}} = \frac{1}{4\pi^2} F^\dagger_{|B|/A} (\mathrm{Id} \otimes \partial_t^2) \psi F \qquad \text{and} \qquad \Lambda^{(2)}_{\mathrm{K}} = \frac{1}{4\pi^2} F^\dagger_{|B|/(c^2 A \|\nabla_{\boldsymbol{x}}\varphi\|)} (\mathrm{Id} \otimes \partial_t^3) \psi F,$$

with the Beylkin determinant

$$B(\boldsymbol{s},\boldsymbol{x}) = \det \begin{pmatrix} \nabla_{\boldsymbol{x}}\varphi(\boldsymbol{s},\boldsymbol{x})^\top \\ \partial_{s_1}\nabla_{\boldsymbol{x}}\varphi(\boldsymbol{s},\boldsymbol{x})^\top \\ \partial_{s_2}\nabla_{\boldsymbol{x}}\varphi(\boldsymbol{s},\boldsymbol{x})^\top \end{pmatrix}.$$

Their respective orders are also 0 and 1.

APPENDIX A

Calculations

A.1 Various calculations

A.1.1 Ψ convolution

For $t > 0$ we have for Ψ from equation (2.2.6)

$$\Psi \star \Psi(t) = \int_{\mathbb{R}} \Psi(t-\eta)\Psi(\eta)\,\mathrm{d}\eta = \int_0^t \frac{1}{4\pi^2\sqrt{(t-\eta)\eta}}\,\mathrm{d}\eta = \frac{1}{4\pi^2}\left[\arcsin\left(\frac{2\eta - t}{t}\right)\right]_0^t = \frac{1}{4\pi}.$$

Therefore $\Psi \star \Psi = \frac{1}{4\pi}H$, where $H(t) = \begin{cases} 1, & t > 0 \\ 0, & t \leq 0 \end{cases}$ is the Heaviside function.

For easier notation, we write $\tau_s = \tau(\boldsymbol{x}, \boldsymbol{x_s})$ and $\tau_r = \tau(\boldsymbol{x}, \boldsymbol{x_r}) = \tau(\boldsymbol{x_r}, \boldsymbol{x})$. In our case, we then have

$$\begin{aligned}
\frac{1}{4\pi}H(t-\tau_s-\tau_r) = \Psi \star \Psi(t-\tau_s-\tau_r) &= \int_0^{t-\tau_s-\tau_r} \Psi(t-\tau_s-\tau_r-r)\Psi(r)\,\mathrm{d}r \\
&= \int_{\tau_r}^{t-\tau_s} \Psi(t-\eta-\tau_s)\Psi(\eta-\tau_r)\,\mathrm{d}\eta \qquad (\eta := r+\tau_r) \\
&= \int_0^t \Psi(t-\eta-\tau_s)\Psi(\eta-\tau_r)\,\mathrm{d}\eta,
\end{aligned}$$

where the last equality holds since $\Psi(\eta - \tau_r) = 0$ for $\eta < \tau_r$ and $\Psi(t-\eta-\tau_s) = 0$ for $\eta > t - \tau_s$.

A.1.2 Factored transport equation

Using $\tau = \tau_0\tau_1$, $a = a_0a_1$ we have

$$\nabla_{\boldsymbol{x}}\tau = \tau_0\nabla_{\boldsymbol{x}}\tau_1 + \tau_1\nabla_{\boldsymbol{x}}\tau_0, \qquad \nabla_{\boldsymbol{x}}a = a_0\nabla_{\boldsymbol{x}}a_1 + a_1\nabla_{\boldsymbol{x}}a_0,$$
$$\Delta_{\boldsymbol{x}}\tau = \tau_0\Delta_{\boldsymbol{x}}\tau_1 + 2\nabla_{\boldsymbol{x}}\tau_0 \cdot \nabla_{\boldsymbol{x}}\tau_1 + \tau_1\Delta_{\boldsymbol{x}}\tau_0.$$

Therefore we get (here $\nabla = \nabla_{\boldsymbol{x}}$ and $\Delta = \Delta_{\boldsymbol{x}}$)

$$
\begin{aligned}
&2\nabla a \cdot \nabla\tau + a\Delta\tau \\
&\quad = 2\left(a_0\nabla a_1 + a_1\nabla a_0\right)\cdot\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right) \\
&\qquad + a_0 a_1\left(\tau_0\Delta\tau_1 + 2\nabla\tau_0\cdot\nabla\tau_1 + \tau_1\Delta\tau_0\right) \\
&\quad = \nabla a_1 \cdot \left[2a_0\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right)\right] \\
&\qquad + a_1\left[2\nabla a_0\cdot\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right) + a_0\left(\tau_0\Delta\tau_1 + 2\nabla\tau_0\cdot\nabla\tau_1 + \tau_1\Delta\tau_0\right)\right] \\
&\quad = \nabla a_1 \cdot \left[2a_0\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right)\right] \\
&\qquad + a_1\left[2\tau_0\nabla a_0\cdot\nabla\tau_1 + 2\tau_1\nabla a_0\cdot\nabla\tau_0 + a_0\tau_0\Delta\tau_1 + 2a_0\nabla\tau_0\cdot\nabla\tau_1 + a_0\tau_1\Delta\tau_0\right] \\
&\quad = \nabla a_1 \cdot \left[2a_0\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right)\right] \\
&\qquad + a_1\left[2\tau_0\nabla a_0\cdot\nabla\tau_1 + a_0\tau_0\Delta\tau_1 + 2a_0\nabla\tau_0\cdot\nabla\tau_1 + \tau_1\underbrace{\left(2\nabla a_0\cdot\nabla\tau_0 + a_0\Delta\tau_0\right)}_{=0}\right] \\
&\quad = \nabla a_1 \cdot \left[2a_0\left(\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0\right)\right] + a_1\left(2\tau_0\nabla a_0\cdot\nabla\tau_1 + a_0\tau_0\Delta\tau_1 + 2a_0\nabla\tau_0\cdot\nabla\tau_1\right)
\end{aligned}
$$

Since, in two dimensions, $\tau_0(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x_s}\|$ and $a_0(\boldsymbol{x}) = \frac{1}{2\sqrt{2\pi}\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}$, we further have

$$
\nabla\tau_0(\boldsymbol{x}) = \frac{\boldsymbol{x}-\boldsymbol{x_s}}{\|\boldsymbol{x}-\boldsymbol{x_s}\|}, \qquad \Delta\tau_0(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}-\boldsymbol{x_s}\|}, \qquad \nabla a_0(\boldsymbol{x}) = -\frac{\boldsymbol{x}-\boldsymbol{x_s}}{4\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{5/2}}.
$$

This leaves us with (dropping the $\boldsymbol{x}$ dependence for readability reasons)

$$
\begin{aligned}
&2\nabla a \cdot \nabla\tau + a\Delta\tau \\
&\quad = \nabla a_1 \cdot \left[\frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}}\nabla\tau_1 + \frac{\left(\boldsymbol{x}-\boldsymbol{x_s}\right)\tau_1}{\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}}\right] \\
&\qquad + a_1\left[-\frac{\boldsymbol{x}-\boldsymbol{x_s}}{2\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}}\cdot\nabla\tau_1 + \frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}}\Delta\tau_1 + \frac{\boldsymbol{x}-\boldsymbol{x_s}}{\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}}\cdot\nabla\tau_1\right] \\
&\quad = \nabla a_1 \cdot \left[\frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}}\nabla\tau_1 + \frac{\left(\boldsymbol{x}-\boldsymbol{x_s}\right)\tau_1}{\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}}\right] \\
&\qquad + a_1\left[\frac{\boldsymbol{x}-\boldsymbol{x_s}}{2\sqrt{2\pi}\,\|\boldsymbol{x}-\boldsymbol{x_s}\|^{3/2}}\cdot\nabla\tau_1 + \frac{\sqrt{\|\boldsymbol{x}-\boldsymbol{x_s}\|}}{\sqrt{2\pi}}\Delta\tau_1\right].
\end{aligned}
$$

A.2 Adjoint operators

A.2.1 Weighted F

For the weighted operator $F_W n(s,t) = \int_X n(\boldsymbol{x}) W(s,t)\delta(t-\varphi(s,\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}$ we have

$$\begin{aligned}\langle F_W n\,,\, g\rangle &= \int_S \int_T \left(\int_X n(\boldsymbol{x}) W(s,\boldsymbol{x})\delta(t-\varphi(s,\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}\right) g(s,t)\,\mathrm{d}t\,\mathrm{d}s \\ &= \int_X n(\boldsymbol{x}) \int_S \int_T W(s,\boldsymbol{x}) g(s,t)\delta(t-\varphi(s,\boldsymbol{x}))\,\mathrm{d}t\,\mathrm{d}s\,\mathrm{d}\boldsymbol{x} \\ &= \int_X n(\boldsymbol{x}) \int_S W(s,\boldsymbol{x}) g(s,\varphi(s,\boldsymbol{x}))\,\mathrm{d}s\,\mathrm{d}\boldsymbol{x} \\ &= \int_X n(\boldsymbol{x}) F_W^\dagger g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \left\langle n\,,\, F_W^\dagger g\right\rangle.\end{aligned}$$

Here, T is the corresponding time interval.

A.2.2 Laplace operator

Since we only apply K^* with $K = -\Delta$ to a compactly supported mollifier e_γ^k, we consider $f, g \in \mathcal{C}_c^2(\Omega)$ with $\Omega \subset \mathbb{R}^2$ open. Using the second Green's identity, we have

$$\langle \Delta f\,,\, g\rangle = \int_\Omega \Delta f \cdot g = \int_\Omega f \cdot \Delta g - \underbrace{\int_{\partial\Omega} (f\nabla g - g\nabla f)\cdot dS}_{=0} = \langle f\,,\, \Delta g\rangle.$$

Therefore, $-\Delta$ is a symmetric operator in our setting.

Further, we have

$$\nabla_{\boldsymbol{x}} \left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^k = -2k \left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-1} \boldsymbol{x},$$

and for $i = 1, 2$,

$$\begin{aligned}\partial_{x_i}^2 \left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^k &= k(k-1)\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-2} 4x_1^2 - 2k\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-1} \\ &= 2k\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-2} \left(2(k-1)x_1^2 - (\gamma^2 - \|\boldsymbol{x}\|^2)\right).\end{aligned}$$

Then

$$\begin{aligned}-\Delta \left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^k &= -2k\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-2} \left((2k-2)\|\boldsymbol{x}\|^2 - 2(\gamma^2 - \|\boldsymbol{x}\|^2)\right) \\ &= 4k\left(\gamma^2 - \|\boldsymbol{x}\|^2\right)^{k-2} \left(\gamma^2 - k\|\boldsymbol{x}\|^2\right).\end{aligned} \tag{A.2.1}$$

A.2.3 Square root of the Laplace operator

For integrable $h\colon \mathbb{R}^2 \to \mathbb{C}$ we have the identity $\overline{\int_{\mathbb{R}^2} h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}} = \int_{\mathbb{R}^2} \overline{h(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}$. Therefore

$$\begin{aligned}
\left\langle (-\Delta)^{1/2} f\,,\, g \right\rangle &= \int_{\mathbb{R}^2} \frac{1}{2\pi} \int_{\mathbb{R}^2} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}}\, \|\boldsymbol{\xi}\|\, \widehat{f}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}\, \overline{g(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^2} \|\boldsymbol{\xi}\|\, \widehat{f}(\boldsymbol{\xi}) \int_{\mathbb{R}^2} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}} \overline{g(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}\;\mathrm{d}\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^2} \|\boldsymbol{\xi}\|\, \widehat{f}(\boldsymbol{\xi}) \overline{\frac{1}{2\pi} \int_{\mathbb{R}^2} \mathrm{e}^{-\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}}\, g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}\;\mathrm{d}\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^2} \|\boldsymbol{\xi}\|\, \widehat{f}(\boldsymbol{\xi}) \overline{\widehat{g}(\boldsymbol{\xi})}\,\mathrm{d}\boldsymbol{\xi} = \int_{\mathbb{R}^2} \widehat{f}(\boldsymbol{\xi}) \overline{\|\boldsymbol{\xi}\|\, \widehat{g}(\boldsymbol{\xi})}\,\mathrm{d}\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^2} f(\boldsymbol{x}) \overline{\frac{1}{2\pi} \int_{\mathbb{R}^2} \mathrm{e}^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{\xi}}\, \|\boldsymbol{\xi}\|\, \widehat{g}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{x} \qquad \textit{(Plancherel)} \\
&= \left\langle f\,,\, (-\Delta)^{1/2} g \right\rangle .
\end{aligned}$$

A.2.4 Partial (time-) derivative

For $f, g \in \mathcal{C}_c^1(T)$, $T = (0, t_{\max})$ we have

$$\begin{aligned}
\langle \partial_t f\,,\, g \rangle &= \int_T f'(t) g(t) dt \\
&= \underbrace{f(t)g(t)\Big|_0^{t_{\max}}}_{=0} - \int_0^{t_{\max}} f(t) g'(t) dt \qquad \textit{(p.i.)} \\
&= -\langle f\,,\, \partial_t g \rangle .
\end{aligned}$$

Therefore, $(\partial_t)^\star = -\partial_t$.

A.3 Intermediate mollifier-calculations

The Fourier transform of e_1^k is given by

$$\widehat{e_1^k}(\rho\boldsymbol{w}) = \frac{2^k (k+1)!}{\pi} \rho^{-(k+1)} J_{k+1}(\rho), \qquad \boldsymbol{w} \in \partial B_1(\boldsymbol{0}),\ \rho > 0, \tag{A.3.1}$$

see [Rie01, Section 3].

We further need to evaluate the following integral. Using polar coordinates, we write $\boldsymbol{x} = \|\boldsymbol{x}\| \, (\cos(\nu), \sin(\nu))^\top$ with some $\nu \in [-\pi, \pi]$, and have

$$\begin{aligned}\int_{\partial B_1(\mathbf{0})} \mathrm{e}^{\mathrm{i}r\boldsymbol{x}\cdot\boldsymbol{w}} \, \mathrm{d}\boldsymbol{w} &= \int_0^{2\pi} \mathrm{e}^{\mathrm{i}r\|\boldsymbol{x}\|(\cos(\theta)\cos(\nu)+\sin(\theta)\sin(\nu))} \, \mathrm{d}\theta \\ &= \int_0^{2\pi} \mathrm{e}^{\mathrm{i}r\|\boldsymbol{x}\|\cos(\theta-\nu)} \, \mathrm{d}\theta = \int_0^{2\pi} \mathrm{e}^{\mathrm{i}r\|\boldsymbol{x}\|\cos(\theta)} \, \mathrm{d}\theta \\ &= \int_{-\pi}^{\pi} \mathrm{e}^{\mathrm{i}r\|\boldsymbol{x}\|\cos(\theta)} \, \mathrm{d}\theta = 2\pi J_0(\|\boldsymbol{x}\| \, r).\end{aligned} \tag{A.3.2}$$

Here we also used the substitution $\boldsymbol{w} = (\cos(\theta), \sin(\theta))^\top$ for the first equality.

A.4 An explicit mollifier expression

Now, we show the step-by-step calculation of the term $(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x})$, starting with with the expression from equation (4.1.2). We begin with some supplementary calculations for Euler's Gamma function.

An important property of the Gamma function is

$$\Gamma(z+1) = z \, \Gamma(z), \quad z > 0, \qquad \Gamma(k+1) = k!, \quad k \in \mathbb{N}.$$

The Legendre duplication formula states

$$\Gamma(z)\Gamma(z+1/2) = 2^{1-2z}\sqrt{\pi} \, \Gamma(2z),$$

and Euler's reflection formula is

$$\Gamma(z)\Gamma(1-z) = \frac{\pi}{\sin(\pi z)}.$$

Using the Legendre duplication formula with $z = 1/2$ leads to

$$\Gamma(1/2) = \sqrt{\pi}, \qquad \Gamma(3/2) = \frac{1}{2}\Gamma(1/2) = \frac{\sqrt{\pi}}{2}, \tag{A.4.1}$$

and inserting $z = \frac{3}{2}$ into Euler's reflection formula results in

$$\Gamma\left(\frac{3}{2}\right)\Gamma\left(-\frac{1}{2}\right) = \frac{\pi}{\sin(\pi 3/2)} \qquad \Rightarrow \qquad \Gamma\left(-\frac{1}{2}\right) = \frac{2\pi}{\sqrt{\pi}\sin(3\pi/2)} = -2\sqrt{\pi}. \tag{A.4.2}$$

Again, by using the Legendre duplication formula with $z = k$ we get

$$\frac{1}{\Gamma(k+1/2)} = \frac{2^{2k}\Gamma(k)}{2\sqrt{\pi} \, \Gamma(2k)} = \frac{4^k(k-1)!}{2\sqrt{\pi} \, (2k-1)!} \frac{k}{k} = \frac{4^k k!}{\sqrt{\pi}(2k)!} \tag{A.4.3}$$

Now we apply Therorem 4.2 to equation (4.1.2). In this case, we have $\mu = 2 - k$, $M = 3$, and get the following two cases:

1. $\|\boldsymbol{x}\| > \gamma$: $v_1 = k+1, v_2 = 0$ and $x_1 = \gamma, x_2 = \|\boldsymbol{x}\|$

$$\begin{aligned}(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) &= \frac{2^k(k+1)!}{\pi\gamma^{k+1}} \frac{2^{1-k}\gamma^{k+1}\|\boldsymbol{x}\|^{-3}\Gamma(3/2)}{\Gamma(k+2)\Gamma(-3/2+1)} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; \frac{\gamma^2}{\|\boldsymbol{x}\|^2}\right] \\ &= \frac{2(k+1)!\,\Gamma(\frac{3}{2})}{\pi\|\boldsymbol{x}\|^3\Gamma(k+2)\Gamma(-\frac{1}{2})} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; \frac{\gamma^2}{\|\boldsymbol{x}\|^2}\right] \\ &\overset{\text{(A.4.1)}}{\underset{\text{(A.4.2)}}{=}} \frac{2\frac{\sqrt{\pi}}{2}}{\pi\|\boldsymbol{x}\|^3(-2)\sqrt{\pi}} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; \frac{\gamma^2}{\|\boldsymbol{x}\|^2}\right] \\ &= -\frac{1}{2\pi\|\boldsymbol{x}\|^3} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; \frac{\gamma^2}{\|\boldsymbol{x}\|^2}\right].\end{aligned}$$

2. $\|\boldsymbol{x}\| < \gamma$: $v_1 = 0, v_2 = k+1$ and $x_1 = \|\boldsymbol{x}\|, x_2 = \gamma$

$$\begin{aligned}(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) &= \frac{2^k(k+1)!}{\pi\gamma^{k+1}} \frac{2^{1-k}\|\boldsymbol{x}\|^0\gamma^{k+1-3}\Gamma(\frac{3}{2})}{\Gamma(1)\Gamma(k+1-\frac{3}{2}+1)} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2} - (k+1); 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right] \\ &= \frac{2(k+1)!\,\Gamma(\frac{3}{2})}{\pi\gamma^3\Gamma(k+\frac{1}{2})} \, {}_2F_1\left[\frac{3}{2}, \frac{3}{2} - (k+1); 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right] \\ &\overset{\text{(A.4.1)}}{=} \frac{2(k+1)!\,\frac{\sqrt{\pi}}{2}}{\pi\gamma^3\Gamma(k+\frac{1}{2})} \, {}_2F_1\left[\frac{3}{2}, \frac{1}{2} - k; 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right] \\ &\overset{\text{(A.4.3)}}{=} \frac{(k+1)!\,\sqrt{\pi}}{\pi\gamma^3} \frac{4^k k!}{\sqrt{\pi}(2k)!} \, {}_2F_1\left[\frac{3}{2}, \frac{1}{2} - k; 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right] \\ &= \frac{4^k(k+1)!\,k!}{\pi\gamma^3(2k)!} \, {}_2F_1\left[\frac{3}{2}, \frac{1}{2} - k; 1; \frac{\|\boldsymbol{x}\|^2}{\gamma^2}\right]\end{aligned}$$

Let us now look into the case $\|\boldsymbol{x}\| \to \gamma$. We will use Gauss's summation theorem (see [LCC01])

$$ {}_2F_1(a, b, c; 1) = \frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)}, \quad c > a + b.$$

1. $\lim_{\|\boldsymbol{x}\| \searrow \gamma}$: $a = \frac{3}{2}$, $b = \frac{3}{2}$, $c = k+2$

$$ {}_2F_1\left[\frac{3}{2}, \frac{3}{2}; k+2; 1\right] = \frac{\Gamma(k+2)\Gamma(k-1)}{\Gamma(k+\frac{1}{2})\Gamma(k+\frac{1}{2})},$$

for $\|\boldsymbol{x}\| \searrow \gamma$ it holds

$$(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) = \frac{2(k+1)!\,\Gamma(\frac{3}{2})}{\pi\|\boldsymbol{x}\|^3\Gamma(k+2)\Gamma(-\frac{1}{2})} \frac{\Gamma(k+2)\Gamma(k-1)}{\Gamma(k+\frac{1}{2})^2} = \frac{2(k+1)!\,\Gamma(\frac{3}{2})\Gamma(k-1)}{\pi\gamma^3\Gamma(-\frac{1}{2})\Gamma(k+\frac{1}{2})^2}.$$

2. $\lim_{\|\boldsymbol{x}\|\nearrow\gamma}$: $a = \frac{3}{2}$, $b = \frac{1}{2} - k$, $c = 1$

$$ {}_2F_1\left[\frac{3}{2}, \frac{1}{2} - k; 1; 1\right] = \frac{\Gamma(1)\Gamma(k-1)}{\Gamma(-\frac{1}{2})\Gamma(k+\frac{1}{2})}, $$

for $\|\boldsymbol{x}\| \nearrow \gamma$ we then have

$$ (-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) = \frac{2(k+1)!\,\Gamma(\frac{3}{2})}{\pi\gamma^3\Gamma(k+\frac{1}{2})} \frac{\Gamma(k-1)}{\Gamma(-\frac{1}{2})\Gamma(k+\frac{1}{2})} = \frac{2(k+1)!\,\Gamma(\frac{3}{2})\Gamma(k-1)}{\pi\gamma^3\Gamma(-\frac{1}{2})\Gamma(k+\frac{1}{2})^2}. $$

Consequently, $\lim_{\|\boldsymbol{x}\|\searrow\gamma}(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x}) = \lim_{\|\boldsymbol{x}\|\nearrow\gamma}(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x})$, and $(-\Delta)^{1/2} e_\gamma^k(\boldsymbol{x})$ is continuous for $\|\boldsymbol{x}\| = \gamma$.

A.5 Translation invariances

For all the following parts we assume that the background velocity c is layered, meaning it depends only on the depth coordinate ($c(\boldsymbol{x}) = c(x_2)$). Let $\boldsymbol{y} = (y_1, 0)^\top \in X$ be an arbitrary vector with 0 as second component. Then

$$ c(\boldsymbol{x} + \boldsymbol{y}) = c(\boldsymbol{x}). $$

A.5.1 Traveltime τ

Let $\tau(\cdot,\cdot)$ be the general solution of the eikonal equation with the second argument corresponding to the source position. Also, let $\dot{\tau}$ be a solution of the eikonal equation with source at $\boldsymbol{x_s} \coloneqq (x_s, 0)^\top$ for $x_s \in \mathbb{R}$:

$$ \|\nabla_{\boldsymbol{x}} \dot{\tau}(\boldsymbol{x})\|^2 = \frac{1}{c^2(\boldsymbol{x})}, \qquad \boldsymbol{x} \in \mathbb{R} \times \mathbb{R}_+ $$
$$ \dot{\tau}(\boldsymbol{x_s}) = 0. $$

Therefore $\dot{\tau}(\boldsymbol{x}) = \tau(\boldsymbol{x}, \boldsymbol{x_s})$.

For $y_1 \in \mathbb{R}$ let $\boldsymbol{y} \coloneqq (y_1, 0)^\top$ and define $\tilde{\boldsymbol{x}}_{\boldsymbol{s}} \coloneqq \boldsymbol{x_s} - \boldsymbol{y}$, $\widetilde{\tau}(\boldsymbol{x}) \coloneqq \dot{\tau}(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}})$. Then we have $\widetilde{\tau}\,(\boldsymbol{y}) = 0$ and since the gradient is shift-invariant, we also have $\nabla_{\boldsymbol{x}} \widetilde{\tau}(\boldsymbol{x}) = (\nabla_{\boldsymbol{x}} \tau)\,(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}})$. Therefore

$$ \|\nabla_{\boldsymbol{x}} \widetilde{\tau}(\boldsymbol{x})\|^2 = \|(\nabla_{\boldsymbol{x}} \tau)\,(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}})\|^2 = \frac{1}{c(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}})} = \frac{1}{c(\boldsymbol{x})}, $$

meaning $\widetilde{\tau}$ solves the eikonal equation

$$\|\nabla_{\boldsymbol{x}}\widetilde{\tau}(\boldsymbol{x})\|^2 = \frac{1}{c^2(\boldsymbol{x})}, \qquad \boldsymbol{x} \in \mathbb{R} \times \mathbb{R}_+$$
$$\widetilde{\tau}\,(\boldsymbol{y}) = 0.$$

This especially means that $\widetilde{\tau}(\boldsymbol{x}) = \tau(\boldsymbol{x}, \boldsymbol{y})$ holds, but by definition of $\widetilde{\tau}$ we also have $\widetilde{\tau}(\boldsymbol{x}) = \dot{\tau}(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}}) = \tau(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}}, \boldsymbol{x}_{\boldsymbol{s}})$ and we therefore get

$$\tau(\boldsymbol{x}, \boldsymbol{y}) = \widetilde{\tau}(\boldsymbol{x}) = \tau\left(\boldsymbol{x} + (x_s - y_1, 0)^\top, (x_s, 0)^\top\right).$$

By setting $x_s = 0$ we conclude

$$\tau(\boldsymbol{x}, \boldsymbol{y}) = \tau\left(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{0}\right).$$

Note that this also provides us with

$$\tau(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{z}) = \tau(\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{z}, \boldsymbol{0}) = \tau(\boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z}),$$

which means that we can arbitrarily shift parts of the source to the first argument, as long as the second component of that coordinate is 0.

A.5.2 Amplitude a

The proof for the amplitude is basically identical to the traveltime. Let $a(\cdot, \cdot)$ be the general solution of the transport equation (2.2.8) with the second argument corresponding to the source position. Let $\dot{a}$ be a solution of the transport equation with source at $\boldsymbol{x}_{\boldsymbol{s}} \coloneqq (x_s, 0)^\top$ for $x_s \in \mathbb{R}$:

$$2\nabla\dot{a}(\boldsymbol{x}) \cdot \nabla_{\boldsymbol{x}}\tau\left(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{s}}\right) + \dot{a}(\boldsymbol{x})\Delta_{\boldsymbol{x}}\tau\left(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{s}}\right) = 0,$$
$$\lim_{\boldsymbol{x} \to \boldsymbol{x}_{\boldsymbol{s}}} \dot{a}(\boldsymbol{x})\sqrt{\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{s}}\|} = \frac{\sqrt{c(\boldsymbol{x}_{\boldsymbol{s}})}}{2\sqrt{2\pi}}.$$

Therefore $\dot{a}(\boldsymbol{x}) = a(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{s}})$.

Let $\boldsymbol{y}$, $\tilde{\boldsymbol{x}}_{\boldsymbol{s}}$, $\dot{\tau}$ and $\widetilde{\tau}$ as in A.5.1 and define $\widetilde{a}(\boldsymbol{x}) \coloneqq \dot{a}(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}})$. Since the gradient and Laplace are shift-invariant, we have

$$\nabla\widetilde{a}(\boldsymbol{x}) = \left(\nabla\dot{a}\right)\left(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}}\right),$$
$$\Delta\widetilde{\tau}(\boldsymbol{x}) = \left(\Delta\dot{\tau}\right)\left(\boldsymbol{x} + \tilde{\boldsymbol{x}}_{\boldsymbol{s}}\right).$$

Since $\widetilde{a}(\boldsymbol{y}) = \dot{a}(\boldsymbol{x_s})$, $\widetilde{a}$ solves the transport equation

$$\begin{aligned}
&\nabla\widetilde{a}(\boldsymbol{x}) \cdot \nabla_{\boldsymbol{x}}\widetilde{\tau}(\boldsymbol{x}) + 2\widetilde{a}(\boldsymbol{x})\Delta\widetilde{\tau}(\boldsymbol{x}) \\
&\qquad = (\nabla\dot{a})\,(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}}) \cdot (\nabla\dot{\tau})\,(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}}) + 2\dot{a}\,(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}})\,(\Delta\dot{\tau})\,(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}}) \\
&\qquad = 0. \\
&\lim_{\boldsymbol{x}\to\boldsymbol{y}} \widetilde{a}(\boldsymbol{x})\sqrt{\|\boldsymbol{x}-\boldsymbol{y}\|} = \frac{\sqrt{c(\boldsymbol{y})}}{2\sqrt{2\pi}}.
\end{aligned}$$

Now, the amplitude $\widetilde{a}$ fulfills $\widetilde{a}(\boldsymbol{x}) = a(\boldsymbol{x},\boldsymbol{y})$, but according to its definition we also have $\widetilde{a}(\boldsymbol{x}) = \dot{a}(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}}) = a(\boldsymbol{x}+\tilde{\boldsymbol{x}}_{\boldsymbol{s}}, \boldsymbol{x_s})$ and therefore

$$a(\boldsymbol{x},\boldsymbol{y}) = \widetilde{a}(\boldsymbol{x}) = a\left(\boldsymbol{x} + (x_s - y_1, 0)^\top, (x_s, 0)^\top\right).$$

By setting $\boldsymbol{x}_s = 0$ we conclude

$$a(\boldsymbol{x},\boldsymbol{y}) = a\left(\boldsymbol{x}-\boldsymbol{y}, \boldsymbol{0}\right).$$

A.5.3 Reflection isochrones $\mathcal{L}_{(s,t)}$ and reconstruction kernel $v_{\boldsymbol{p},\gamma,W}$

Again, let $\boldsymbol{y} = (y_1, 0)^\top \in X$. We assume $\boldsymbol{x_s}(s+y_1) = \boldsymbol{x_s}(s)+\boldsymbol{y}$ and $\boldsymbol{x_r}(s+y_1) = \boldsymbol{x_r}(s)+\boldsymbol{y}$, which is the case for the common-offset geometry. For τ and a we have (see A.5.1, A.5.2)

$$\begin{aligned}
\tau(\boldsymbol{x}, \boldsymbol{x_s}(s)) &= \tau\left(\boldsymbol{x}, \boldsymbol{x_s}(s+y_1) - \boldsymbol{y}\right) = \tau\left(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_s}(s+y_1)\right), \\
a(\boldsymbol{x}, \boldsymbol{x_s}(s)) &= a\left(\boldsymbol{x}, \boldsymbol{x_s}(s+y_1) - \boldsymbol{y}\right) = a\left(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_s}(s+y_1)\right),
\end{aligned}$$

and the same holds true for $\boldsymbol{x_r}$. Therefore we have

$$\begin{aligned}
\varphi(s+y_1, \boldsymbol{x}+\boldsymbol{y}) &= \tau(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_s}(s+y_1)) + \tau(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_r}(s+y_1)) \\
&= \tau(\boldsymbol{x}, \boldsymbol{x_s}(s)) + \tau(\boldsymbol{x}, \boldsymbol{x_r}(s)) \;= \varphi(s, \boldsymbol{x}),
\end{aligned}$$

and

$$\begin{aligned}
A(s+y_1, \boldsymbol{x}+\boldsymbol{y}) &= a(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_s}(s+y_1))\; a(\boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x_r}(s+y_1)) \;/\; c^2(\boldsymbol{x}+\boldsymbol{y}) \\
&= a(\boldsymbol{x}, \boldsymbol{x_s}(s))\; a(\boldsymbol{x}, \boldsymbol{x_r}(s)) \;/\; c^2(\boldsymbol{x}) \;= A(s, \boldsymbol{x}).
\end{aligned}$$

For the isochrones we have

$$\begin{aligned}
\mathcal{L}_{(s,t)} &= \{\boldsymbol{x} \in X : \varphi(s, \boldsymbol{x}) = t\} \\
&= \{\boldsymbol{x} \in X : \varphi(s+y_1, \boldsymbol{x}+\boldsymbol{y}) = t\} \\
&= \{\tilde{\boldsymbol{x}} - \boldsymbol{y} \in X : \varphi(s+y_1, \tilde{\boldsymbol{x}}) = t\} && (\tilde{\boldsymbol{x}} := \boldsymbol{x}+\boldsymbol{y}) \\
&= \{\tilde{\boldsymbol{x}} \in X : \varphi(s+y_1, \tilde{\boldsymbol{x}}) = t\} + \boldsymbol{y} = \mathcal{L}_{(s+y_1,t)} + \boldsymbol{y}.
\end{aligned}$$

Let $X \ni \boldsymbol{p} = \widehat{\boldsymbol{p_1}} + \widehat{\boldsymbol{p_2}}$ with $\widehat{\boldsymbol{p_1}} \coloneqq \begin{pmatrix} \widetilde{p_1} \\ 0 \end{pmatrix}$ and $\widehat{\boldsymbol{p_2}} \coloneqq \begin{pmatrix} r \\ p_2 \end{pmatrix}$ for $r \in \mathbb{R}$, $\widetilde{p_1} \coloneqq p_1 - r$. Assume the weight W fulfills the same shift invariance as A, which is the case for all weights we use. Then we conclude $v_{\widehat{\boldsymbol{p_2}},\gamma,W}(s - \widetilde{p_1}, t) = M^\star\, \widetilde{v}_{\widehat{\boldsymbol{p_2}},\gamma,W}(s - \widetilde{p_1}, t)$ using

$$
\begin{aligned}
&\widetilde{v}_{\widehat{\boldsymbol{p_2}},\gamma,W}(s - \widetilde{p_1}, t) \\
&\quad = \frac{1}{\sqrt{2}} \int_{\mathcal{L}_{\left(s-\widetilde{p_1},t\right)}} \frac{c(\boldsymbol{x}) W(s - \widetilde{p_1}, \boldsymbol{x})\; K^\star e_\gamma(\boldsymbol{x} - \widehat{\boldsymbol{p_2}})}{\sqrt{1 + c^2(\boldsymbol{x}) \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_s}(s - \widetilde{p_1})) \cdot \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_r}(s - \widetilde{p_1}))}} \,\mathrm{ds}(\boldsymbol{x}) \\
&\quad = \frac{1}{\sqrt{2}} \int_{\mathcal{L}_{(s,t)}} \frac{c(\boldsymbol{x} - \widehat{\boldsymbol{p_1}}) W(s - \widetilde{p_1}, \boldsymbol{x} - \widehat{\boldsymbol{p_1}})\; K^\star e_\gamma(\boldsymbol{x} - \widehat{\boldsymbol{p_1}} - \widehat{\boldsymbol{p_2}})}{\sqrt{1 + c^2(\boldsymbol{x} - \widehat{\boldsymbol{p_1}}) \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x} - \widehat{\boldsymbol{p_1}}, \boldsymbol{x_s}(s - \widetilde{p_1})) \cdot \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x} - \widetilde{p_1}, \boldsymbol{x_r}(s - \widetilde{p_1}))}} \,\mathrm{ds}(\boldsymbol{x}) \\
&\quad = \frac{1}{\sqrt{2}} \int_{\mathcal{L}_{(s,t)}} \frac{c(\boldsymbol{x}) W(s, \boldsymbol{x})\; K^\star e_\gamma(\boldsymbol{x} - \boldsymbol{p})}{\sqrt{1 + c^2(\boldsymbol{x}) \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_s}(s)) \cdot \nabla_{\boldsymbol{x}} \tau(\boldsymbol{x}, \boldsymbol{x_r}(s))}} \,\mathrm{ds}(\boldsymbol{x}) \\
&\quad = \widetilde{v}_{\boldsymbol{p},\gamma,W}(s, t).
\end{aligned}
$$

And therefore $v_{\widehat{\boldsymbol{p_2}},\gamma,W}(s - \widetilde{p_1}, t) = v_{\boldsymbol{p},\gamma,W}(s, t)$.

This means we can shift an arbitrary part of the first component of $\boldsymbol{p}$ to the argument. For example, using $r = 0$ gives $v_{\boldsymbol{p},\gamma,W}(s, t) = v_{(0,p_2)^\top,\gamma,W}(s - p_1, t)$.

Bibliography

[Ber94] J.-P. Berenger. "A perfectly matched layer for the absorption of electromagnetic waves". In: *Journal of Computational Physics* 114.2 (1994), pp. 185–200. ISSN: 0021-9991. DOI: 10.1006/jcph.1994.1159.

[Bey85] G. Beylkin. "Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized Radon transform". In: *Journal of mathematical physics* 26.1 (1985), pp. 99–108. DOI: 10.1063/1.526755.

[BC79] N. Bleistein and J. K. Cohen. "A Velocity inversion procedure for acoustic waves". In: *Geophysics* 44.6 (1979), pp. 1077–1087. DOI: 10.1190/1.1440996.

[BCS01] N. Bleistein, J. K. Cohen, and J. W. Stockwell Jr. *Mathematics of Multidimensional Seismic Imaging, Migration, and Inversion*. Vol. 13. Interdisciplinary Applied Mathematics. Geophysics and Planetary Sciences. Springer-Verlag, New York, 2001. ISBN: 978-1-4613-0001-4. DOI: 10.1007/978-1-4613-0001-4.

[Bro99] R. L. Brown. "Birth of the Seismic-Reflection Method". In: *Oklahoma Geological Survey Special Publication 99-1* (1999), pp. 6–9.

[CDY09] E. Candès, L. Demanet, and L. Ying. "A Fast Butterfly Algorithm for the Computation of Fourier Integral Operators". In: *Multiscale Modeling & Simulation* 7.4 (2009), pp. 1727–1750. DOI: 10.1137/080734339.

[CCK13] W. Chen, C.-S. Chou, and C.-Y. Kao. "Lax–Friedrichs fast sweeping methods for steady state problems for hyperbolic conservation laws". In: *Journal of Computational Physics* 234 (2013), pp. 452–471. DOI: 10.1016/j.jcp.2012.10.008.

[CH89] R. Courant and D. Hilbert. *Methods of mathematical physics: partial differential equations*. Vol. 2. John Wiley & Sons, 1989. DOI: 10.1002/9783527617234.

[CL84] M. G. Crandall and P.-L. Lions. "Two approximations of solutions of Hamilton-Jacobi equations". In: *Math. Comp.* 43.167 (1984), pp. 1–19. ISSN: 0025-5718. DOI: 10.2307/2007396.

[De +09] M. V. De Hoop, H. Smith, G. Uhlmann, and R. D. Van Der Hilst. "Seismic imaging with the generalized Radon transform: a curvelet transform perspective". In: *Inverse Problems* 25.2 (2009), p. 025005. DOI: 10.1088/0266-5611/25/2/025005.

[Dem21] L. Demanet. *Waves and Imaging*. Class Notes. Department of Mathematics, MIT, 2021. URL: https://math.mit.edu/icg/resources/teaching/18.367/notes367.pdf.

[DH] L. Demanet and R. J. Hewett. *PySIT - Seismic Imaging Toolbox for Python*. URL: http://pysit.org/ (visited on 10/17/2024).

[DH07] H. Douma and M. V. de Hoop. "Leading-order seismic imaging using curvelets". In: *Geophysics* 72.6 (2007), S231–S248. DOI: 10.1190/1.2785047.

[ES79] H. Exton and H. Srivastava. "A generalization of the Weber-Schafheitlin integral". In: *Journal für die reine und angewandte Mathematik* 309 (1979). DOI: doi:10.1515/crll.1979.309.1.

[Fel+16] R. Felea, V. P. Krishnan, C. J. Nolan, and E. T. Quinto. "Common midpoint versus common offset acquisition geometry in seismic imaging". In: *Inverse Problems and Imaging* 10.1 (2016), pp. 87–102. DOI: 10.3934/ipi.2016.10.87.

[FJ98] F. G. Friedlander and M. S. Joshi. *Introduction to the Theory of Distributions*. 2nd. Cambridge University Press, 1998. ISBN: 9780521649711.

[FQ15] J. Frikel and E. T. Quinto. "Artifacts in Incomplete Data Tomography with Applications to Photoacoustic Tomography and Sonar". In: *SIAM Journal on Applied Mathematics* 75.2 (2015), pp. 703–725. DOI: 10.1137/140977709.

[Gan25] K. Ganster. *The Numerical Scheme of Approximate Inverse for 2D Linear Seismic Imaging - Software Package*. 2025. DOI: 10.35097/zv2af52st4qgqcek.

[GQR24] K. Ganster, E. T. Quinto, and A. Rieder. "A microlocal and visual comparison of 2D Kirchhoff migration formulas in seismic imaging". In: *Inverse Problems* (2024). DOI: 10.1088/1361-6420/ad797b.

[GR23] K. Ganster and A. Rieder. "Approximate inversion of a class of generalized Radon transforms". In: *SIAM Journal on Imaging Sciences* 16.2 (2023), pp. 842–866. DOI: 10.1137/22M1512417.

[Gau12] C. F. Gauss. "Disquisitiones generales circa seriem infinitam ..." In: *Pars prior* (1812), pp. 125–162.

[Gau88] C. F. Gauss. *Allgemeine Untersuchungen über die unendliche Reihe ...* Springer, 1888.

[Gra20] C. Grathwohl. "Seismic imaging with the elliptic Radon transform in 3D: analytical and numerical aspects". PhD thesis. KIT-Bibliothek, 2020. DOI: 10.5445/IR/1000105093.

[Gra+17] C. Grathwohl, P. Kunstmann, E. T. Quinto, and A. Rieder. "Approximate inverse for the common offset acquisition geometry in 2D seismic imaging". In: *Inverse Problems* 34.1 (2017), p. 014002. DOI: 10.1088/1361-6420/aa9900.

[Gra+18] C. Grathwohl, P. Kunstmann, E. T. Quinto, and A. Rieder. "Microlocal analysis of imaging operators for effective common offset seismic reconstruction". In: *Inverse Problems* 34.11 (2018), p. 114001. DOI: 10.1088/1361-6420/aadc2a.

[Gra+20] C. Grathwohl, P. C. Kunstmann, E. T. Quinto, and A. Rieder. "Imaging with the Elliptic Radon Transform in Three Dimensions from an Analytical and Numerical Perspective". In: *SIAM Journal on Imaging Sciences* 13.4 (2020), pp. 2250–2280. DOI: 10.1137/20M1332657.

[GO07] V. Grubišić and M. Orlić. "Early observations of rotor clouds by Andrija Mohorovičić". In: *Bulletin of the American Meteorological Society* 88.5 (2007), pp. 693–700. DOI: 10.1175/BAMS-88-5-693.

[Har02] P. Hartman. *Ordinary Differential Equations*. 2nd ed. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898719222.

[Hel11] S. Helgason. *Integral geometry and Radon transforms*. Vol. 1. Springer New York, 2011. DOI: 10.1007/978-1-4419-6055-9.

[Hör65] L. Hörmander. "Pseudo-differential operators". In: *Communications on Pure and Applied Mathematics* 18.3 (1965), pp. 501–517. DOI: 10.1002/cpa.3160180307.

[Hör71] L. Hörmander. "Fourier integral operators. I". In: *Acta Mathematica* 127 (1971), pp. 79–183. DOI: 10.1007/BF02392052.

[Hör03] L. Hörmander. *The analysis of linear partial differential operators I. Distribution Theory and Fourier Analysis*. 2nd ed. Classics in Mathematics. Springer Berlin, Heidelberg, 2003. DOI: 10.1007/978-3-642-61497-2.

[KQ15] V. P. Krishnan and E. T. Quinto. "Microlocal Analysis in Tomography." In: *Handbook of mathematical methods in imaging* 1 (2015), p. 3. URL: https://math.tifrbng.res.in/~vkrishnan/Papers/Microlocal_Analysis_Tomography.pdf.

[KQR23] P. C. Kunstmann, E. T. Quinto, and A. Rieder. "Seismic imaging with generalized Radon transforms: stability of the Bolker condition". In: *Pure and Applied Mathematics Quarterly* 19.4 (2023), pp. 1985–2036. DOI: 10.4310/PAMQ.2023.v19.n4.a11.

[LCC01] H.-J. Lee, Y. Cho, and J. Choi. "Gauss summation theorem and its applications". In: *East Asian mathematical journal* 17 (Jan. 2001).

[LC87] W. E. Lorensen and H. E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm". In: Association for Computing Machinery, 1987. DOI: 10.1145/37401.37422.

[Lou96] A. Louis. "Approximate inverse for linear and some nonlinear problems". In: *Inverse problems* 12.2 (1996), p. 175. DOI: 10.1088/0266-5611/12/2/005.

[LQZ12] S. Luo, J. Qian, and H. Zhao. "Higher-order schemes for 3D first-arrival traveltimes and amplitude". In: *Geophysics* 77.2 (2012), T47–T56. DOI: 10.1190/geo2010-0363.1.

[LQB14] S. Luo, J. Qian, and R. Burridge. "Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media in the high frequency regime". In: *J. Comput. Phys.* 270 (2014), pp. 378–401. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2014.03.066.

[Mar] Marine Geoscience Data System. *SE Caribbean Seismic Project, Cruise EW0404*. URL: https://www.marine-geo.org/tools/search/entry.php?id=EW0404 (visited on 06/14/2024).

[NS97] C. J. Nolan and W. W. Symes. "Global solution of a linearized inverse problem for the wave equation". In: *Communications in Partial Differential Equations* 22.5-6 (1997), pp. 127–149. DOI: 10.1080/03605309708821289.

[OS91] S. Osher and C.-W. Shu. "High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations". In: *SIAM J. Numer. Anal.* 28.4 (1991), pp. 907–922. DOI: 10.1137/0728049.

[Pan95] J. N. Pandey. *The Hilbert transform of Schwartz distributions and applications*. John Wiley & Sons, 1995. DOI: 10.1002/9781118032510.

[Pet83] B. E. Petersen. *Introduction to the Fourier transform & Pseudo-differential Operators*. Pitman Advanced Pub. Program, 1983.

[Rie01] A. Rieder. "Principles of reconstruction filter design in 2D-computerized tomography". In: *Radon Transforms and Tomography*. Vol. 278. American Mathematical Society, 2001, pp. 207–226.

[RT92] E. Rouy and A. Tourin. "A viscosity solutions approach to shape-from-shading". In: *SIAM J. Numer. Anal.* 29.3 (1992), pp. 867–884. DOI: 10.1137/0729053.

[Rud73] W. Rudin. *Functional Analysis*. Mcgraw-Hill Book Company, 1973.

[SQ10] S. Serna and J. Qian. "A stopping criterion for higher-order sweeping schemes for static Hamilton-Jacobi equations". In: *Journal of Computational Mathematics* 4 (2010), pp. 552–568. ISSN: 02549409, 19917139. URL: http://www.jstor.org/stable/43693921.

[Set99] J. A. Sethian. "Fast marching methods". In: *SIAM Rev.* 41.2 (1999), pp. 199–235. ISSN: 0036-1445. DOI: 10.1137/S0036144598347059.

[Sjö00] J. Sjöstrand. "Microlocal Analysis". In: *Development of mathematics 1950-2000*. Ed. by J.-P. Pier. Birkhäuser Verlag, 2000, pp. 967–991.

[Sto00] C. C. Stolk. "Microlocal analysis of a seismic linearized inverse problem". In: *Wave Motion* 32.3 (2000), pp. 267–290. ISSN: 0165-2125. DOI: 10.1016/S0165-2125(00)00043-3.

[Sym98] W. W. Symes. *Mathematics of reflection seismology*. Tech. rep. The Rice Inversion Project, Rice University, Houston, TX, USA, 1998. URL: http://www.trip.caam.rice.edu/downloads/preamble.pdf.

[Tit37] E. C. Titchmarsh. *Introduction to the theory of Fourier integral*. Oxford At The Clarendon Press, 1937.

[TH16] E. Treister and E. Haber. "A fast marching algorithm for the factored eikonal equation". In: *J. Computat. Phys.* 324 (2016), pp. 210–225. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.08.012.

[Trè80] J.-F. Trèves. *Introduction to Pseudodifferential and Fourier Integral Operators. Volume 1: Pseudodifferential operators*. Vol. 1. University Series in Mathematics. Springer Science & Business Media, 1980. DOI: 10.1007/978-1-4684-8780-0.

[Trè81] J.-F. Trèves. *Introduction to Pseudodifferential and Fourier Integral Operators. Volume 2: Fourier integral operators*. Vol. 2. University Series in Mathematics. Springer Science & Business Media, 1981.

[Yil01] Ö. Yilmaz. *Seismic data analysis*. Vol. 1. Society of exploration geophysicists Tulsa, 2001. DOI: 10.1190/1.9781560801580.

[ZZQ06] Y.-T. Zhang, H.-K. Zhao, and J. Qian. "High order fast sweeping methods for static Hamilton-Jacobi equations". In: *J. Sci. Comput.* 29.1 (2006), pp. 25–56. ISSN: 0885-7474. DOI: 10.1007/s10915-005-9014-3.

Danksagung

In meiner Zeit am KIT wurde ich von vielen Menschen unterstützt. Einigen davon möchte ich an dieser Stelle danken.

Prof. Dr. Andreas Rieder danke ich für die Betreuung und das gesetzte Vertrauen während meiner Doktorandenzeit. Auch die eine Lehrveranstaltung, dessen Übung ich betreuen durfte, hat letztenendes mehr Spaß gemacht als ursprünglich erwartet.

Apl. Prof. Dr. Peer Christian Kunstmann danke ich für die Übernahme der Zweitkorrektur und für ein kurzes, aber dennoch hilfreiches Treffen.

Prof. Dr. Eric Todd Quinto möchte ich auch mein herzliches Dank aussprechen. Obwohl er immer viel zu tun hat, hat er trotzdem immer seine Hilfe angeboten, insbesondere wenn es um Fragen zur mikrolokalen Analysis ging. Thank you Todd!

Ich danke Prof. Dr. Thomas Bohlen, Dr. Thomas Hertweck und Dr. Lars Houpt vom Geophysischen Institut des KIT für das Bereitstellen und Aufbereiten der Felddaten und den hilfreichen Kommentaren zu meinem Felddatenexperiment aus Geophysiker-Perspektive.

In meiner Arbeitsgruppe am Institut für Angewandte und Numerische Mathematik 3 habe ich mich nicht erst zu Beginn der Promotion, sondern schon ein paar Jahre vorher wohlgefühlt. Hier durfte ich als Tutor für die Veranstaltung Einführung in Python arbeiten und etliche Stunden in dem ehemaligen Bürozimmer von Johannes Ernesti verbringen und mit ihm diskutieren. Die verschiedenen arbeitgruppeninternen Feiern wie die jährliche Weihnachtfeier, Semesterabschlussfeier oder das eine oder andere Fest im Garten von Prof. Dr. Wieners oder Dr. Daniel Weiß, haben auch immer sehr viel Spaß gemacht und waren extrem wertvoll. Insbesondere hat Daniel für das Organisieren der ganzen Feiern und das generelle Bemühen, das Zusammenleben in der Arbeitsgruppe positiv zu gestalten, meinen Dank verdient. Ich möchte ein paar weitere Persönlichkeiten besonders hervorheben, aber generell besteht unsere Arbeitsgruppe ausschließlich aus tollen Leuten, mit denen man sehr gut und gerne Zusammenarbeiten kann.

Johanna, danke für die Kekse 🍪 und sonstige Süßigkeitsangebote. Für die Wartung meiner Fahhräder 🚲 möchte ich mich bei Michael bedanken. Ebenso bei Laura für die Tipps zur Pflanzenpflege und bei David für die Bespaßung mit dummen Dingen wie z.B. selbsgemachten Elephantengeräuschen.

Hai Dang, danke für die Begleitung und (teilweise) hilfreichen Tipps beim Bouldern, auch wenn “die leichteste X, die du je gesehen hast”, $X \in \{6, 7\}$, meistens doch nicht

free war. Ich hoffe du schaffst es auch irgendwann die inneren Strukturen von M++ zu verstehen.

Auch Christian danke ich als weiteren Boulder-Buddy und dafür, meine etlichen Fragen zu den Übungsblättern in Numerische Optimierungsmethoden mit bestem Wissen und Gewissen zu beantworten.

Lukas Pieronek danke ich für die interessanten Einsichten rund um maschinelles Lernen und für die verschenkten Schokobons. Wir müssen bald Mal wieder zu Kebabi gehen.

~~Bettina~~ David – was geht? ✌ – vielen Dank für die Hilfe bei jeglichen Fragen theoretischer Natur und Korrekturlesen der Arbeit. Viel Erfolg beim Zukleben von bestimmten Parteiplakaten mit Stickern anderer Parteien in den kommenden Wahlen.

Daniele, ich weiß gar nicht wo ich anfangen soll. Eine Zeit am KIT ohne dich kann ich mir rückwirkend gar nicht vorstellen. Seit Beginn des Mathestudiums sind jetzt schon fast 12 Jahre vergangen, und trotzdem erinnere ich mich noch oft an das "gemeinsame"[1] Bearbeiten der Übungsblätter, oder die Mittagspausen, in denen wir immer verzweifelt überlegt haben wo wir essen gehen. Die Diskussionen im Discord (oder im letzten Jahrzehnt noch im TeamSpeak) zusammen mit Lukas sorgen immer noch für das ein oder andere Schmunzeln. Damals, als wir noch zusammen Minecraft oder Path of Exile I gespielt haben, hatte ich auch immer sehr viel Spaß (auch wenn ich den Trade-Spielstil immer noch nicht nachvollziehen kann). Ich hoffe du findest in Zukunft Mal wieder mehr Zeit hin und wieder mit Einzusteigen. Auch dir danke ich natürlich noch für das Korrekturlesen der Arbeit. Deine generelle Hilfbereitschaft ist bewundernswert, ich hoffe die bleibt die noch lange erhalten. Allerdings hoffe ich auch du findest die Überwindung, deine eigene Dissertation fertig zu schreiben. Abschließend wünsche ich dir und Sophie viel Erfolg und wenig Probleme beim Aufziehen von Levio.

Zu guter Letzt möchte ich mich bei den Personen, die nicht zu meinem Arbeitsumfeld gehören, bedanken.

Meinem langjährigen Freund Roman danke ich für die sehr gute Freundschaft, die regen aber trotzdem sehr wertvollen gemeinsamen Treffen, wie zum Beispiel die jährliche Geburtstagsfeier in Müllheim, und sonstige Unterstützung abseits der Arbeit. Weiterhin viel Erfolg beim Abschließen deiner eigenen Doktorarbeit (auch wenn das in der Medizin nicht ganz das Gleiche ist).

Abschließend darf natürlich auch der Dank an meine Familie nicht fehlen. Ihr habt mich nicht nur in meiner Zeit am KIT, sondern auch in allen anderen Entscheidungen unterstützt und ermutigt.

[1] hier bedeutete das meistens, dass Daniele erklärt wie man die Aufgabe löst

Acknowledgement

I gratefully acknowledge the support of the German Research Foundation (DFG) through the Collaborative Research Centre (CRC) 1173 “Wave phenomena: Analysis and Numerics”.