Karlsruhe Institute of Technology

wave phenomena
analysis and numerics

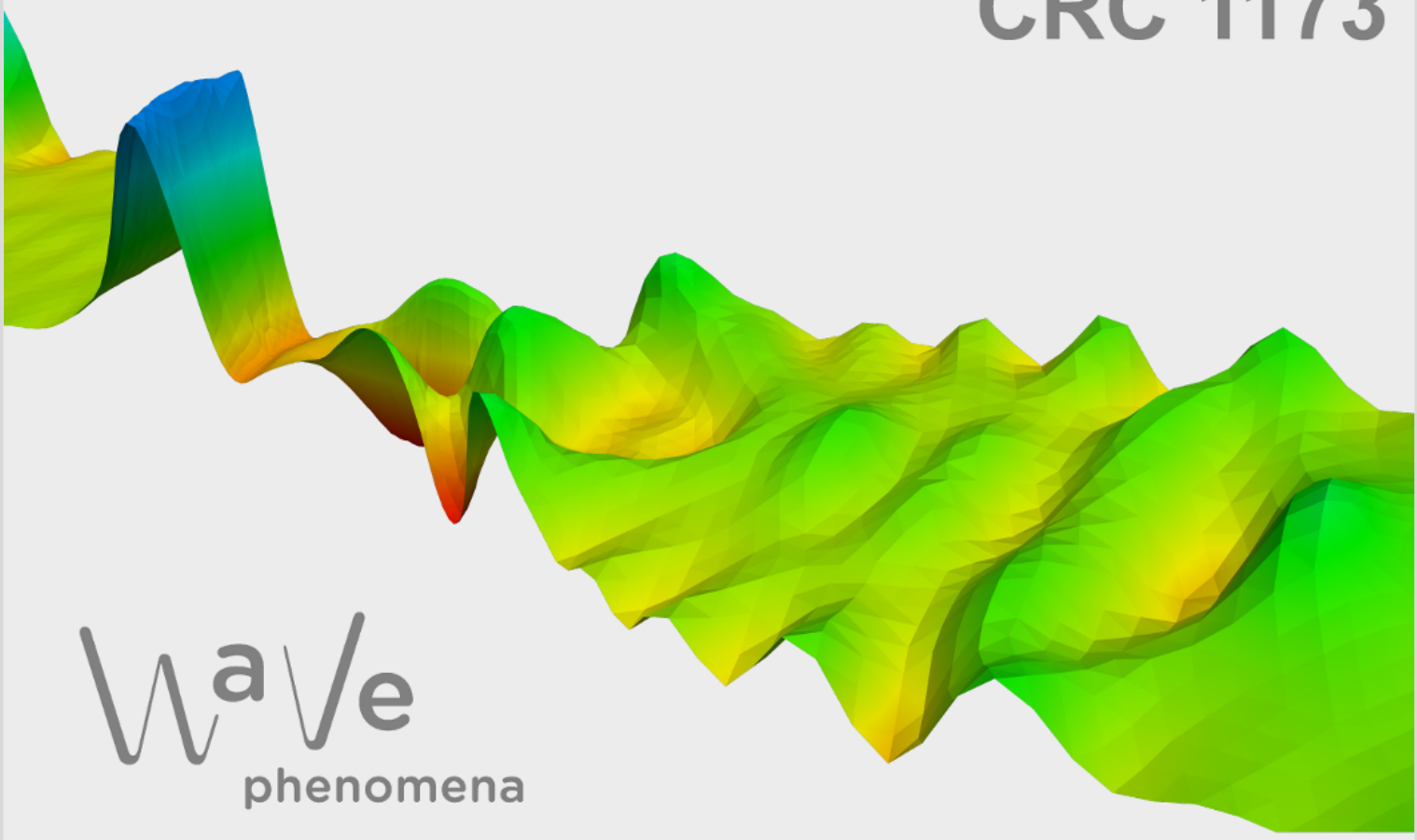
# Approximate inversion of generalized Radon transforms

Kevin Ganster, Andreas Rieder

CRC Preprint 2022/33, July 2022

KARLSRUHE INSTITUTE OF TECHNOLOGY

CRC 1173



**Participating universities**

Universität Stuttgart

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN


**Funded by**


ISSN 2365-662X

# APPROXIMATE INVERSION OF GENERALIZED RADON TRANSFORMS

KEVIN GANSTER AND ANDREAS RIEDER

Abstract. Generalized Radon transforms (GRT) serve, for instance, as linear models for seismic imaging in the acoustic regime. They occur when the corresponding inverse problem is linearized about a known background compression wave speed (Born approximation). The resulting GRT is completely determined by this background velocity. In this work, we present an implementation of approximate inversion formulas for this class of GRTs proposed and analyzed in [*Inverse Problems*, 34 (2018), 014002, 114001], where we restrict ourselves to layered background velocities in 2D. In a series of numerical experiments, we intensively test our implementation, reproducing theoretical predictions. Further, we drive the validity of the linearization to its limits.

## 1. Introduction

Generalized Radon transforms (GRT) emerge, for instance, when linearizing the seismic inverse problem of recovering physical properties of the medium from reflected wave fields. These transforms integrate the sought quantity $n$ over reflection isochrones connecting points of equal travel time to the source that triggers the wave and to the receiver that records the reflected parts of the wave. More specifically, in the acoustic regime with constant density, let $F$ denote the GRT and $g$ the processed data, then $Fn = g$. Here, $n$ represents the high frequency component of the true speed of sound $\nu_{\mathrm{pr}}$ by means of the ansatz

$$\frac{1}{\nu_{\mathrm{pr}}^2(\boldsymbol{x})} = \frac{1+n(\boldsymbol{x})}{c^2(\boldsymbol{x})} \tag{1.1}$$

with the known background velocity $c$. Hence, the concrete expression of $F$ depends on $c$.

For the solution of $Fn = g$ one relies on approximate inversion formulas of filtered backprojection type like Kirchhoff migration. This traditional inversion scheme in geophysics can be expressed as $F^\dagger K y$ where $K$ is a convolution filter and $F^\dagger$ denotes a dual transform (generalized backprojection). Thus, we reconstruct $F^\dagger K F n$ which is a low-pass filtered version of $n$ superimposed with a smooth artifact, see [1].

Recently, an alternative approach was propagated and analyzed in [10, 11, 12]. The authors propose the *imaging operator* $\Lambda = KF^*\psi F$ where $F^*$ is a formal (weighted) $L^2$-adjoint, $\psi$ is a smooth cutoff function, and the properly supported pseudodifferential operator $K$ of positive order emphasizes singularities since these carry most of the information content. Using microlocal analysis, the authors have constructed $K$ with useful imaging properties. They have also developed a corresponding regularization scheme

*Date*: July 28, 2022.

2010 *Mathematics Subject Classification.* 65R32, 45Q05, 35S30, 86A22.

*Key words and phrases.* generalized Radon transform, approximate inverse.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 258734477 - SFB 1173.

based on the approximate inverse [13]: Let $e_{\boldsymbol{p}}$ denote a smooth approximation to the Dirac distribution located at $\boldsymbol{p}$ and define the reconstruction kernel $v_{\boldsymbol{p}} := FK^* e_{\boldsymbol{p}}$. Then,

$$\langle \psi g, v_{\boldsymbol{p}} \rangle = \langle \psi F n, v_{\boldsymbol{p}} \rangle = \langle \Lambda n, e_{\boldsymbol{p}} \rangle \approx \Lambda n(\boldsymbol{p}),$$

that is, the $L^2$-inner product of the kernel with the data approximates the desired quantity. In [9] this inversion scheme was realized and demonstrated for the special case when linearization is done about a constant background wave speed, say, $c(\cdot) = 1$. Then, the GRT reduces to the elliptical Radon transform which integrates over ellipses (ellipsoids in 3D) with source and receiver positions as foci.

In the present work we extend the approximate inverse to GRTs arising from linearization about layered background wave speeds and the common offset data acquisition geometry. Considering exclusively this scanning geometry is not a principle restriction. For instance, data from the common midpoint or common source geometries can be processed as demonstrated in [9, Section 4.1].

We have organized the presentation of our material as follows. In the next section we shortly recall how the GRT arises from a linearization of the nonlinear inverse problem of recovering the compression sound speed in the acoustic regime from reflected wave fields. This linearization is done with respect to a smooth background velocity which satisfies the geometric optics assumption. We will see that the concrete application of the GRT to a function demands the solution of two partial differential equations, the eikonal and a transport equation, both of which depend on the background velocity. Section 3 is then devoted in more detail to the concept of approximate inverse. In particular, we explain all the steps that must be executed to obtain a reconstruction kernel, which essentially requires the application of the GRT to a compactly supported function. So we need to solve the eikonal and transport equations numerically, using the Fast Marching scheme for the former (Section 4.1) and a Fast Sweeping method for the latter (Section 4.2). In the third and last part of Section 4 we explain the evaluation of the kernel and the implementation of the approximate inverse for the GRT in case of a layered background velocity. Finally, we use our implementation to investigate numerically in Section 5 how different background velocities affect the reconstructions. Here we consider velocities which satisfy and which violate the geometric optics assumption. In order to reproduce theoretical predictions about the imaging operators we first rely on consistent data, which are in the image of the GRT and which are therefore free of any model error. Then, we generate the data by solving the underlying acoustic wave equation, which allows us to study the influence of the linearization error.

## 2. The setting

If the medium does not support shear stress, the underlying model for wave propagation is the acoustic wave equation

$$\frac{1}{\nu_{\mathrm{pr}}^2}\partial_t^2 u - \Delta_{\boldsymbol{x}} u = \delta(\boldsymbol{x} - \boldsymbol{x_s})\delta(t), \quad u|_{t=0} = \partial_t u|_{t=0} = 0, \tag{2.1}$$

for the pressure wave $u = u(t; \boldsymbol{x}, \boldsymbol{x_s})$, $\boldsymbol{x} \in \mathbb{R}^d$, $d \in \{2, 3\}$, at time $t \geq 0$. Here, the wave is initiated by a source fired at time $t = 0$ and location $\boldsymbol{x_s}$. The corresponding nonlinear inverse problem is to recover the speed of sound $\nu_{\mathrm{pr}} = \nu_{\mathrm{pr}}(\boldsymbol{x})$ from measurements of $u(\cdot; \boldsymbol{x_r}, \boldsymbol{x_s})$ at pairs $(\boldsymbol{x_s}, \boldsymbol{x_r})$ where $\boldsymbol{x_r}$ indicates receiver positions. In the sequel we assume that pairs of source and receiver points can be smoothly parametrized by a variable $s$ and we restrict ourselves to $d = 2$.

We linearize the inverse problem about a known smooth background velocity $c = c(\boldsymbol{x})$ by the ansatz (1.1), where $n$ is compactly supported in

$$X = \mathbb{R}^2_+, \tag{2.2}$$

which is the lower half space, that is, $x_2 > 0$ (the positive direction of the $x_2$-axis is the depth-coordinate and points downwards to the interior of the earth). Sources and receivers are positioned on the line $x_2 = 0$.

We require $c$ to satisfy the geometric optics assumption, that is, any two points in $X$ are connected by a unique ray (see Appendix B for the definition of a ray). If $\widetilde{u}$ denotes the reference solution, which solves (2.1) with $c$ in place of $\nu_{\mathrm{pr}}$, then we can derive the following linear integral equation

$$Fn(t,s) = 4\pi \int_0^t (\widetilde{u} - u)(r; \boldsymbol{x_r}(s), \boldsymbol{x_s}(s))\mathrm{d}r \tag{2.3}$$

with operator

$$Fw(t,s) = \int_X w(\boldsymbol{x})\, A(s,\boldsymbol{x})\, \delta\left(t - \varphi(s,\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}, \quad t > t_{\mathrm{first}}, \tag{2.4}$$

where

$$\varphi(s,\boldsymbol{x}) \coloneqq \tau(\boldsymbol{x}, \boldsymbol{x_s}(s)) + \tau(\boldsymbol{x}, \boldsymbol{x_r}(s)) \quad \text{and} \quad A(s,\boldsymbol{x}) \coloneqq \frac{a(\boldsymbol{x}, \boldsymbol{x_s}(s)) a(\boldsymbol{x}, \boldsymbol{x_r}(s))}{c^2(\boldsymbol{x})}. \tag{2.5}$$

In (2.4), $t_{\mathrm{first}} = t_{\mathrm{first}}(s)$ is the first arrival time, that is, the minimal time a wave needs to travel from source to receiver, see, e.g., [23, Section 6] or [10] and also [2, Appendix E].

The travel time $\tau$ and the amplitude $a$ in (2.4) can be computed from the eikonal equation

$$|\nabla_{\boldsymbol{x}} \tau(\cdot, \boldsymbol{x_s})|^2 = \frac{1}{c^2}, \quad \tau(\boldsymbol{x_s}, \boldsymbol{x_s}) = 0, \tag{2.6}$$

and from the transport equation

$$2\nabla_{\boldsymbol{x}} a(\cdot, \boldsymbol{x_s}) \cdot \nabla_{\boldsymbol{x}} \tau(\cdot, \boldsymbol{x_s}) + a(\cdot, \boldsymbol{x_s}) \Delta_{\boldsymbol{x}} \tau(\cdot, \boldsymbol{x_s}) = 0, \tag{2.7}$$

respectively. The operator $F$ is a GRT as the integral $Fn(t,s)$ extends over reflection isochrones connecting points of equal travel time $t$ to source and to receiver.

We define the data space

$$Y = \{(s,t) : s \in S_0,\ t \geq t_{\mathrm{first}}(s)\}, \tag{2.8}$$

where $S_0 \subset \mathbb{R}$ is the bounded open set containing the parameters for the source/receiver pairs.

The imaging operators we consider are of the form

$$\Lambda \coloneqq K F^* \psi F, \tag{2.9}$$

where $\psi \colon Y \to [0, \infty[$ is a smooth compactly supported cutoff function and $K$ is a properly supported pseudodifferential operator on $X$ of nonnegative order $\kappa$. Further, $F^*$ is the formal $L^2$-adjoint

$$F^* u(\boldsymbol{x}) = \iint_Y A(s,\boldsymbol{x}) u(s,t) \delta(t - \varphi(s,\boldsymbol{x})) \mathrm{d}t\, \mathrm{d}s = \int_{S_0} A(s,\boldsymbol{x}) u(s, \varphi(s,\boldsymbol{x})) \mathrm{d}s.$$

The cutoff $\psi$ in (2.9) is needed in general so that $F^*$ and $\psi F$ can be composed. It further helps to reduce limited data artifacts.

2.1. **Common offset data acquisition geometry.** In this scanning geometry, sources and receivers are parameterized by

$$\boldsymbol{x_s}(s) = (s-\alpha, 0)^\top \quad \text{and} \quad \boldsymbol{x_r}(s) = (s+\alpha, 0)^\top \tag{2.10}$$

with a common offset $\alpha \geq 0$.

In case $c(\cdot) = 1$, we have $t_{\text{first}} = 2\alpha$ and $t_{\text{first}} = 2\,\text{asinh}(\alpha\,\text{m}/\text{b})/\text{m}$ for $c(\boldsymbol{x}) = \text{m}\,x_2 + \text{b}$, $\text{m}, \text{b} > 0$. In both cases, $F\colon \mathcal{E}'(X) \to \mathcal{D}'(Y)$ is a Fourier integral operator and $\Lambda\colon \mathcal{E}'(X_0) \to \mathcal{D}'(X)$ is a pseudodifferential operator of order $\kappa - 1$ for a suitable open subset $X_0$ of $X$ (for $c(\cdot) = 1$: $X_0 = X$). A microlocal study of $\Lambda$ reveals which singularities (wave fronts) of $n$ are visible in $\Lambda n$ as well. Further, the choice

$$K = \Delta(M^q + \beta\,\text{Id}) \tag{2.11}$$

with the Laplacian $\Delta$, suitable constants $\beta, q \geq 0$, and $M$ being the multiplication operator with the depth-coordinate $x_2$, yields a useful imaging operator of order 1, see [10, 12]. For some results under a different acquisition geometry we refer to [5].

## 3. Approximate inverse

The concept of approximate inverse fits the structure of $\Lambda$ perfectly. In fact, let $\{e_\gamma\}_{\gamma>0}$ be a family of mollifiers, that is, $\int e_\gamma(x)\mathrm{d}x = 1$, $e_\gamma \to \delta$ as $\gamma \to 0$ where $\delta$ denotes the Dirac distribution. Further, let $e_\gamma$ be compactly supported in the ball about the origin with radius $\gamma$. Then, for $\boldsymbol{p} \in X$ and $0 < \gamma < p_2$, we have that

$$\Lambda_\gamma n(\boldsymbol{p}) \coloneqq \Lambda n * e_\gamma(\boldsymbol{p}) = \langle \psi g, v_{\boldsymbol{p},\gamma}\rangle_{L^2(Y)} \quad \text{for } g = Fn, \tag{3.1}$$

where the reconstruction kernel

$$v_{\boldsymbol{p},\gamma} = FK^* e_\gamma(\cdot - \boldsymbol{p}) \tag{3.2}$$

has to be calculated in advance or during inversion. Observe that the smoothed version $\Lambda n * e_\gamma$ of $\Lambda n$ can now be computed from the data $g$ via inner products with the kernel.

We will use admissible and proven mollifiers

$$e_\gamma(\boldsymbol{x}) = C_{k,\gamma} \begin{cases} (\gamma^2 - |\boldsymbol{x}|^2)^k & : \ |\boldsymbol{x}| < \gamma, \\ 0 & : \ |\boldsymbol{x}| \geq \gamma, \end{cases} \tag{3.3}$$

with a design parameter $k > 0$, which determines the smoothness of $e_\gamma$, and

$$C_{k,\gamma} = \frac{k+1}{\pi\,\gamma^{2(k+1)}}.$$

To evaluate the kernel we have to solve numerically the eikonal (2.6) and transport equations (2.7). The former equation, for instance, can be tackled by fast marching methods [21, 22], fast sweeping methods [6] or a hybrid of both methods [24]. For the latter equation we may use fast and highly accurate finite difference schemes [14, 18]. In either case we consider $\tau$ and $a$ to be known at grid points in what follows.

The following tasks have to be performed, where we assume that $c$ is layered, that is, it does only depend on the depth-coordinate. Assume that $\widetilde{e}_{\boldsymbol{p},\gamma} \coloneqq K^* e_\gamma(\cdot - \boldsymbol{p})$ can be calculated analytically and has the same compact support as $e_\gamma(\cdot - \boldsymbol{p})$; an assumption which holds true for $K$ as in (2.11). So, it remains to evaluate the GRT (2.4) applied to $\widetilde{e}_{\boldsymbol{p},\gamma}$ numerically.

Next we refine the arguments from [9, Appendix A] and, thus, clarify how the integral

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

has to be evaluated. Here $\delta$ is the one-dimensional Dirac distribution and $X$ is as in (2.2). Note that

$$v_{\boldsymbol{p},\gamma}(s,t) = \widetilde{F}\left(A(s,\cdot)\,\widetilde{e}_{\boldsymbol{p},\gamma}(\cdot)\right)(s,t), \tag{3.4}$$

see (2.4). We will validate below that $\widetilde{F}\rho(s,t)$ has to be understood as

$$\begin{aligned}\widetilde{F}\rho(s,t) &= \int_{\mathcal{L}_{(s,t)}} \frac{\rho(s,\boldsymbol{x})\,\mathrm{ds}(\boldsymbol{x})}{|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})|}\\ &= \frac{1}{\sqrt{2}}\int_{\mathcal{L}_{(s,t)}} \frac{c(\boldsymbol{x})\,\rho(s,\boldsymbol{x})\,\mathrm{ds}(\boldsymbol{x})}{\sqrt{1+c^2(\boldsymbol{x})\,\nabla_{\boldsymbol{x}}\tau\left(\boldsymbol{x},\boldsymbol{x_s}(s)\right)\cdot\nabla_{\boldsymbol{x}}\tau\left(\boldsymbol{x},\boldsymbol{x_r}(s)\right)}},\end{aligned} \tag{3.5}$$

where

$$\mathcal{L}_{(s,t)} = \{\boldsymbol{x}\in X\colon \varphi(s,\boldsymbol{x}) = t\} \tag{3.6}$$

is the reflection isochrone relative to $(s,t)\in Y$ (2.8). The denominator of the integrand is positive if no forward scattering occurs, that is, if $\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_s}) \neq -\nabla_{\boldsymbol{x}}\tau(\boldsymbol{x},\boldsymbol{x_r})$ for $\boldsymbol{x}\in\mathcal{L}_{(s,t)}$, an assumption which we already made tacitly to derive (2.3) from the acoustic wave equation. A more detailed discussion of this topic can be found in Remark 4.5 of [12].

Finally, $\widetilde{F}\rho(s,t)$ can be approximated, for instance, by the trapezoidal sum based on knots in $\mathcal{L}_{(s,t)}$.

The proof of (3.5) starts with the limit representation of the one-dimensional Dirac distribution, applies then the coarea formula and finally determines the limit by the Lebesgue differentiation theorem:

$$\widetilde{F}\rho(s,t) = \lim_{h\searrow 0}\frac{1}{h}\int_{\bigcup\{\mathcal{L}_{(s,r)}\,:\,t\le r\le t+h\}}\rho(s,\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \lim_{h\searrow 0}\frac{1}{h}\int_t^{t+h}\int_{\mathcal{L}_{(s,r)}}\frac{\rho(s,\boldsymbol{x})}{|\nabla_{\boldsymbol{x}}\varphi(s,\boldsymbol{x})|}\,\mathrm{ds}(\boldsymbol{x})\mathrm{d}r.$$

In the following we summarize and explain in some details the steps that need to be done to compute the kernel. They will be explained in full detail in the following section.

(1) Solve the eikonal (2.6) and transport equations (2.7) by one of the methods mentioned above. Under the assumption of a layered $c$ we enjoy a simplification: $\tau\left(\boldsymbol{x},(y_1,0)^\top\right) = \tau\left(\boldsymbol{x}-(y_1,0)^\top,\boldsymbol{0}\right)$ and the same holds true for $a$. Thus, $\tau$ and $a$ have to be determined only with respect to the origin.
(2) Using well-established algorithms (marching squares/cubes) we find efficiently the reflection isochrones $\mathcal{L}_{(s,t)}$ for the recorded travel times $t > t_{\mathrm{first}}(s)$.
(3) Finally evaluate $\widetilde{F}\left(A(s,\cdot)\,\widetilde{e}_{\boldsymbol{p},\gamma}(\cdot)\right)(s,t)$ by a quadrature rule. Here we will benefit to some extent from simplifications we used before in [9, Section 3] and which come from the fact that the kernel resembles the seismogram of a point reflector. In fact, the kernel is sparse in $Y$.

Under the common offset scanning geometry we have further simplifications which greatly reduce the numerical effort: the reconstruction kernels and the reflection isochrones exhibit translation invariances:

$$v_{\boldsymbol{p},\gamma}(s,t) = v_{(r,p_2),\gamma}(s-p_1+r,t),\ r\in\mathbb{R},\quad\text{and}\quad \mathcal{L}_{(s,t)} = \mathcal{L}_{(0,t)} + (s,0)^\top. \tag{3.7}$$

## 4. Implementation

We now state how we solve the eikonal (2.6) and transport equations (2.7) on a discrete square grid $X_h := \{\boldsymbol{x}_{i,j} := \boldsymbol{z} + (ih_1, jh_2) : i, j \in \mathbb{N}_0,\ i < n_1,\ j < n_2\}$ with grid-spacing $h = (h_1, h_2) \in \mathbb{R}^2_{>0}$, suitable $\boldsymbol{z} \in \mathbb{R}^2$ and $n_1, n_2 \in \mathbb{N}$. The neighborhood of a grid-point $\boldsymbol{x}_{i,j}$ will then be $\mathcal{N}_{i,j} = \mathcal{N}(\boldsymbol{x}_{i,j}) := \{\boldsymbol{x}_{i-1,j}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i,j-1}, \boldsymbol{x}_{i,j+1}\} \cap X_h$.

4.1. **Solving the eikonal equation.** We solve the eikonal equation (2.6) by the Fast Marching scheme which we outline briefly and refer to [22] for a comprehensive presentation. The general idea is to construct $\tau$ off the source $\boldsymbol{x}_{\boldsymbol{s}}$ by using an *upwind discretization* of the gradient. This is feasible since the information propagates in one direction, namely from small to large values of $\tau$.

With $\tau_{i,j} = \tau(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{\boldsymbol{s}})$ we define the (one-sided) backward and forward finite differences for each coordinate direction as

$$D_{i,j}^{-x_1}\tau := \frac{\tau_{i,j} - \tau_{i-1,j}}{h_1}, \qquad D_{i,j}^{+x_1}\tau := \frac{\tau_{i+1,j} - \tau_{i,j}}{h_1},$$
$$D_{i,j}^{-x_2}\tau := \frac{\tau_{i,j} - \tau_{i,j-1}}{h_2}, \qquad D_{i,j}^{+x_2}\tau := \frac{\tau_{i,j+1} - \tau_{i,j}}{h_2}.$$

Now the Godunov upwind discretization of the partial derivatives, see, e.g., [20], yields

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

and

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

Thus, we need to solve the quadratic equation

$$\sum_{k=1}^{2}\max\left\{D_{i,j}^{-x_k}\tau, -D_{i,j}^{+x_k}\tau, 0\right\}^2 = \frac{1}{c_{i,j}^2} \tag{4.1}$$

for $\tau_{i,j}$ where $c_{i,j} = c(\boldsymbol{x}_{i,j})$. Of course, the use of higher order (one-sided) finite differences is also permitted.

The Fast Marching Algorithm operates as follows. The grid $X_h$ is split into three disjoint sets: points $\boldsymbol{x}_{i,j}$ are labeled *known* when the value of $\tau_{i,j}$ has been accepted and will not be recalculated, *unknown* when $\tau_{i,j}$ has not been updated yet, and *front* when $\tau_{i,j}$ received at least one update, but has not been labeled known. Then, the algorithm iterates over the grid points by picking a point in *front* with smallest travel time, which becomes then *known*, that is, the points in *front* 'march' outwards. In this way the correct flow of information is guaranteed. Further details are given in Algorithm 1, where the function solve_quadratic solves (4.1). For the implementation of solve_quadratic the Godunov terms are rewritten as

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

with $\beta_1 = \min\{\tau_{i-1,j}, \tau_{i+1,j}\}$ and $\beta_2 = \min\{\tau_{i,j-1}, \tau_{i,j+1}\}$. Assuming $\tau_{i,j} - \beta_k > 0$, $k = 1, 2$, we get

$$\eta\,\tau_{i,j}^2 + \widetilde{\beta}\,\tau_{i,j} + \gamma = 0,$$

where

$$\eta = \sum_{k=1}^{2} h_k^{-2}, \qquad \widetilde{\beta} = -2\sum_{k=1}^{2} h_k^{-2}\beta_k, \qquad \gamma = -\frac{1}{c_{i,j}^2} + \sum_{k=1}^{2} h_k^{-2}\beta_k^2.$$

If our assumption is violated, i.e., either $\tau_{i,j} - \beta_1 \leq 0$ or $\tau_{i,j} - \beta_2 \leq 0$ (one of the differences must be positive) then we drop the corresponding terms in the above sums defining $\eta$, $\widetilde{\beta}$, and $\gamma$.

**Algorithm 1:** fast_marching

$\tau_{i,j} \leftarrow \begin{cases} 0, & \boldsymbol{x}_{i,j} = \boldsymbol{x}_{\boldsymbol{s}}, \\ \infty, & \text{otherwise.} \end{cases}$ *# initialize*
known $\leftarrow \emptyset$
front $\leftarrow \{\boldsymbol{x}_{\boldsymbol{s}}\}$
**while** front $\neq \emptyset$ **do**
  $\boldsymbol{x}_{l,m} \leftarrow \arg\min\{\tau_{i,j} : \boldsymbol{x}_{i,j} \in \text{front}\}$ *# find the minimal entry in front*
  known $\leftarrow$ known $\cup \{\boldsymbol{x}_{l,m}\}$ *# add $\boldsymbol{x}_{l,m}$ to known and take it out of front*
  front $\leftarrow$ front $\setminus \{\boldsymbol{x}_{l,m}\}$
  $\mathcal{N} \leftarrow X_h \cap \mathcal{N}_{l,m} \setminus$known *# determine unknown neighborhood of $\boldsymbol{x}_{l,m}$*
  front $\leftarrow$ front $\cup\, \mathcal{N}$ *# update front*
  **foreach** $\boldsymbol{x}_{i,j} \in \mathcal{N}$ **do**
    $\tau_{i,j} \leftarrow$ solve_quadratic($\boldsymbol{x}_{i,j}$) *# update $\tau_{i,j}$ by solving the quadratic equation*
  **end**
**end**
**return** $\tau$

Actually, we implemented the Factored Fast Marching scheme [24] to solve a factored version of the eikonal equation

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

so that $\tau = \tau_0\tau_1$. The factor $\tau_0$ is introduced to mitigate the singularity of $\tau$ at the source. Further, $\tau_1$ can be calculated by Algorithm 1 with a few adjustments to the finite difference operators.[1]

4.2. **Solving the transport equation.** Continuing with the transport equation (2.7), we first introduce a broader class of so-called (static) *Hamilton-Jacobi equations*. We consider static ($2D$) Hamilton-Jacobi 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 $H$ is a continuous scalar function on $X \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ called Hamiltonian. For a compact notation, we will write $H(u, p, q)$ instead of $H(\boldsymbol{x}, u(\boldsymbol{x}), \partial_{x_1} u(\boldsymbol{x}), \partial_{x_2} u(\boldsymbol{x}))$.

We set $u_{i,j} = u(\boldsymbol{x}_{i,j})$ and use the *Lax-Friedrichs numerical Hamiltonian* [17, eq. (2.3)]

$$\begin{aligned} H^{LF}(u_{i,j}, u_{\mathcal{N}_{i,j}}) = H\left(u_{i,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2h_1}, \frac{u_{i,j+1} - u_{i,j-1}}{2h_2}\right) \\ - \alpha_1 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{2h_1} - \alpha_2 \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{2h_2}, \end{aligned} \tag{4.2}$$

[1] Our implementation is available at https://github.com/kevinganster/eikonalfm.

where the artificial viscosities $\alpha_k$ are chosen such that the flux $H^{LF}$ is monotone, consistent and has differenced form to ensure convergence of the scheme, see [3] and [17].

As recommended in [16], we set in our implementation

$$\alpha_1 = \max_{A\le u\le B,\, C\le p\le D,\, E\le q\le F} \{|\partial_p H(u,p,q)| + |\partial_u H(u,p,q)|\},$$
$$\alpha_2 = \max_{A\le u\le B,\, c\le p\le D,\, E\le q\le F} \{|\partial_q H(u,p,q)| + |\partial_u H(u,p,q)|\},$$

where $[A,B]$, $[C,D]$ and $[E,F]$ are the ranges of the expressions $u_{i,j}$, $\frac{u_{i+1,j}-u_{i-1,j}}{2h_1}$ and $\frac{u_{i,j+1}-u_{i,j-1}}{2h_2}$, respectively.

Reformulating $H^{LF} = f$ as a fixed point equation for $u_{i,j}$ and employing a nonlinear Gauss-Seidel iteration gives the update rule for the Lax-Friedrichs sweeping scheme depending on the sweeping cycle. For instance, if we sweep from lower left $(i,j=0)$ to upper right $(i=n_1, j=n_2)$, we have

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

To obtain a higher order scheme we replace $u_{\mathcal{N}_{i,j}}$ in (4.2) by the WENO approximations $(u_k)_{i,j}^{\pm}$, see [25]:

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

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

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

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 $\epsilon$ is a small positive number to prevent division by zero. Thus, the update formula we actually implemented is

$$u_{i,j}^{\text{new}} = \frac{1}{\frac{\alpha_1}{h_1}+\frac{\alpha_2}{h_2}} \left[ f_{i,j} - H\left( u_{i,j}^{\text{old}}, \frac{(u_1)_{i,j}^{+} + (u_1)_{ij}^{-}}{2}, \frac{(u_2)_{i,j}^{+} + (u_2)_{i,j}^{-}}{2} \right) + \alpha_1 \frac{(u_1)_{i,j}^{+} - (u_1)_{i,j}^{-}}{2} + \alpha_2 \frac{(u_2)_{i,j}^{+} - (u_2)_{i,j}^{-}}{2} \right] + u_{i,j}^{\text{old}}. \tag{4.3}$$

See Agorithm 2 for a basic outline of the Lax-Friedrich sweeping method according to [16] with 4 sweeping cycles/directions.

**Algorithm 2:** Lax-Friedrich Sweeping

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

**while** $\left\|u^{new} - u^{old}\right\|_\infty > \delta$ **do** *#stop criterion for given tolerance* $\delta > 0$
  **for** $i, j$ **in** directions$[i_{\text{dir}}]$ **do**
    **if** $\boldsymbol{x}_{i,j} \notin$ fixed **then**
      $u_{i,j} \leftarrow$ update by (4.3), extrapolating values at points off $X_h$ (ghost points)
    **end**
  **end**
  $i_{\text{dir}} = (i_{\text{dir}} + 1) \mod 4$ *# cycle to the next sweeping direction*
**end**

In the concrete situation of the transport equation (2.7) the Hamiltonian reads

$$H(u, p, q) = 2 \begin{pmatrix} p \\ q \end{pmatrix} \cdot \nabla\tau + u\Delta\tau.$$

We factorize the amplitude $a = a_0\, a_1$ with $a_0(\boldsymbol{x}) = \frac{1}{2\sqrt{2\pi}\sqrt{|\boldsymbol{x}-\boldsymbol{x_s}|}}$ to handle the source singularity [15, eq. (8)] and solve for $a_1$. Now,

$$\begin{aligned} H(u, p, q) &= 2 \left( a_0 \begin{pmatrix} p \\ q \end{pmatrix} + u\nabla a_0 \right) \cdot \nabla\tau + a_0 u\Delta\tau \\ &= 2a_0\, p\, \partial_{x_1}\tau + 2a_0\, q\, \partial_{x_2}\tau + (2\nabla a_0 \cdot \nabla\tau + a_0\Delta\tau)\, u \end{aligned}$$

and the artificial viscosities are

$$\alpha_k = \max_{\boldsymbol{x} \in X_h} \left\{ |2\nabla a_0 \cdot \nabla\tau + a_0\Delta\tau| + |2a_0\, \partial_{x_k}\tau| \right\}, \quad k = 1, 2.$$

To obtain reasonable initial values (first instruction in Agorithm 2), we consider the constant velocity case $c(\cdot) = c_0$, in which

$$a(\boldsymbol{x}) = \frac{\sqrt{c_0}}{2\sqrt{2\pi}\sqrt{|\boldsymbol{x} - \boldsymbol{x_s}|}} = a_0(\boldsymbol{x})\sqrt{c_0}.$$

In the general situation we accordingly initialize by $a_1(\cdot) = \sqrt{c(\cdot)}$.

4.3. **Computing the kernels and implementing the approximate inverse.** In this section we explain first how to evaluate the reconstruction kernels (3.2) for a layered $c$ via (3.4) and (3.5). Then, we implement the approximate inverse, that is, we compute $\Lambda_\gamma n(\boldsymbol{p})$ efficiently for $\boldsymbol{p} = (p_1, p_2)$ in a certain mesh $\mathcal{M}_p \subset X$.

We restrict ourselves to the common offset scanning geometry (2.10). Our assumptions result in a translation invariance for $\tau$ and $a$:

$$\tau\left(\boldsymbol{x},(y_1,0)^\top\right)=\tau\left(\boldsymbol{x}-(y_1,0)^\top,\mathbf{0}\right),\qquad a\left(\boldsymbol{x},(y_1,0)^\top\right)=a\left(\boldsymbol{x}-(y_1,0)^\top,\mathbf{0}\right).$$

Further, we have the invariances (3.7).

By $\tau_\alpha$ we denote the $\alpha$-shifted version of $\tau(\cdot,\mathbf{0})$, that is,

$$\tau_\alpha(\boldsymbol{x})=\tau(\boldsymbol{x},(\alpha,0)^\top)=\tau((x_1-\alpha,x_2)^\top,\mathbf{0})$$

and we use the same definition for $a_\alpha$. So, if we know $\tau$ and $a$ with respect to the origin we know them everywhere, in principle. However, the involved translations call for a respective enlargement of our computational domain. For example, if we want to evaluate $\tau_{+\alpha}$ on $[-5,5]\times[0,5]$, we need to compute $\tau(\cdot,\mathbf{0})$ on $[-5-\alpha,5-\alpha]\times[0,5]$. Analogously for $\tau_{-\alpha}$. Hence, we compute $\tau(\cdot,\mathbf{0})$ on $[-5-\alpha,5+\alpha]\times[0,5]$ to get $\tau_{\pm\alpha}$ on the requested rectangle.

In the discrete case, the described restriction is a slicing operation on the computed arrays to cut out specific parts. For an array arr, the notation arr[*ind*] defines a sub-array, where *ind* denotes a range $i:j$ with $i,j\in\{0,\dots,\mathrm{len}(\mathrm{arr})\}$, $i<j$, or a set of indices. Here, len(arr) is the number of elements in arr. For a range, the starting index is inclusive, while the end index is exclusive. Note that we are using zero-based numbering for array indices. Consider arr $=[1,2,3,4,5,6]$. Then, arr$[1:4]=[2,3,4]$ and arr$[\{2,5\}]=[3,6]$. To keep the whole array, we simply write arr[:]. Slicing applies to higher dimensions by treating each dimension separately:

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

We need further notation: by $\mathcal{I}([a,b],n)$ we denote the discrete interval $[a,b]$ with number of grid-points $n$, including the end-points, and equidistant spacing with step size $h=\frac{b-a}{n-1}$. Let

$$\mathcal{S}=\mathcal{I}([s_{\min},s_{\max}],n_{\mathcal{S}})\quad\text{and}\quad\mathcal{T}=\mathcal{I}([t_{\min},t_{\max}],n_{\mathcal{T}}) \tag{4.4}$$

where $s_{\min}<s_{\max}$ and $t_{\text{first}}\le t_{\min}<t_{\max}$. Hence, $\mathcal{S}\times\mathcal{T}$ is the discretized $s$-$t$-domain for the kernels. We require $n_{\mathcal{S}}$ to be odd. This is not a principal restriction, however, it eases somewhat the presentation because the midpoint $s_{\mathrm{mid}}=(s_{\max}+s_{\min})/2$ is in $\mathcal{S}$.

In the sequel we must carefully distinguish between functions of continuous arguments (such as $\tau$ and $a$ etc.) and arrays containing numerical approximations of these functions evaluated at points of a grid. To this end we write $f_{\mathcal{G}}$ for the array containing the approximations of $f$ restricted to $\mathcal{G}$. Thus, $f_{\mathcal{G}}\approx f|_{\mathcal{G}}$ but $f_{\mathcal{G}}\neq f|_{\mathcal{G}}$ in general.

4.3.1. *Input data for reference kernels.* To obtain $v_{\boldsymbol{p},\gamma}$ on $\mathcal{S}\times\mathcal{T}$ it is sufficient to calculate $v_{(s_{\mathrm{mid}},p_2),\gamma}$ on an extended grid $\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}$ on which we can realize the shift operation (3.7) with $r=s_{\mathrm{mid}}$ by slicing as explained above in the cases of $\tau_{\pm\alpha}$ and $a_{\pm\alpha}$. Here, $\mathcal{S}_{\mathrm{ref}}=\mathcal{I}([s_{\min}-p_{\mathrm{ref}},s_{\max}+p_{\mathrm{ref}}],n_{\mathrm{ref}})$ and in Section 4.3.3 below we will assign values to $p_{\mathrm{ref}}$ and $n_{\mathrm{ref}}$ to enable slicing for all $\boldsymbol{p}\in\mathcal{M}_p$. Note that $\mathcal{S}\subset\mathcal{S}_{\mathrm{ref}}$ will be satisfied.

In this subsection we process the needed input data to compute $v_{(s_{\mathrm{mid}},p_2),\gamma}$ on $\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}$ for $p_2$ in a discrete subset $P_{\mathrm{ref}}$ of the non-negative real numbers. The resulting arrays $(v_{(s_{\mathrm{mid}},p_2),\gamma})_{\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}}$, $p_2\in P_{\mathrm{ref}}$, are called *reference kernels*.

To start we fix an equidistant Cartesian mesh $\mathcal{M}\subset X$ which contains the origin and is symmetric with respect to the axis $x_1=0$. Further, we require $\mathcal{M}$ to be large enough:

$\mathcal{L}_{(0,t)} \subset \operatorname{conv}(\mathcal{M})$ (convex hull) for all $t \in \mathcal{T}$. Let the number of grid-points be $n_1^{\mathcal{M}} \times n_2^{\mathcal{M}}$ with the step sizes $h_1^{\mathcal{M}}$ and $h_2^{\mathcal{M}}$. Define $i_\alpha := \lceil \alpha / h_1^{\mathcal{M}} \rceil$[2] as the number of points on a line segment on the $x_1$-axis with length $\alpha$. As explained above, to obtain the arrays $(\tau_{\pm\alpha})_{\mathcal{M}}$ and $(a_{\pm\alpha})_{\mathcal{M}}$, we expand $\mathcal{M}$ on both sides of the $x_1$-coordinate direction by adding $i_\alpha$ grid-points to the left and to the right. We then solve the eikonal and transport equations on this expanded mesh $\mathcal{M}_\alpha$ with respect to the origin using Algorithms 1 and 2 to obtain the arrays $\tau_{\mathcal{M}_\alpha}$ and $a_{\mathcal{M}_\alpha}$. Then,

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

and this relation holds for $a$ as well. In the actual implementation we only calculate the right halves of $\tau_{\mathcal{M}_\alpha}$ and $a_{\mathcal{M}_\alpha}$ because we can flip the results for symmetry.

Now, we have $\varphi_{\mathcal{M}} = (\tau_{-\alpha})_{\mathcal{M}} + (\tau_{+\alpha})_{\mathcal{M}}$ to determine the reflection isochrones (3.6). Further, we set

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

where the arithmetic operations on arrays are understood element-wise. The gradients of $(\tau_{-\alpha})_{\mathcal{M}}$ and $(\tau_{+\alpha})_{\mathcal{M}}$ are arrays of pairs containing approximations to the respective partial derivatives at the grid points which have been computed from $(\tau_{\pm\alpha})_{\mathcal{M}}$ by second order central differences at interior points and first order one-sided differences at the boundaries.

Note that $I_{\mathcal{M}}$ is an important ingredient for the kernels being independent of the mollifier, especially, it is independent of $\boldsymbol{p} \in \mathcal{M}_p$, see (3.4) and (3.5). We continue in the following section with finally computing the reference kernels relying on the arrays $\varphi_{\mathcal{M}}$ and $I_{\mathcal{M}}$.

4.3.2. *Reference kernels.* For $p_2 \in P_{\text{ref}}$ we here provide the array $(v_{(s_{\text{mid}},p_2),\gamma})_{\mathcal{S}_{\text{ref}} \times \mathcal{T}}$. For any $t \in \mathcal{T}$ we determine the reflection isochrones $\mathcal{L}_{(0,t)} = \{\boldsymbol{x} \in X : \varphi(0, \boldsymbol{x}) = t\}$ using the marching squares algorithm[3] on the array $\varphi_{\mathcal{M}}$. This yields a list of points in $X$ approximating $\mathcal{L}_{(0,t)}$. In the sequel we identify this list with $\mathcal{L}_{(0,t)}$. As the discrete points in $\mathcal{L}_{(0,t)}$ lie on edges connecting points in $\mathcal{M}$ we extend $I_{\mathcal{M}}$ to $\mathcal{L}_{(0,t)}$ by linear interpolation (along the respective edge). In view of (3.7) we have $\mathcal{L}_{(s,t)}$ as discrete set as well.

Now, if $K^* e_\gamma(\cdot - \boldsymbol{p})$ is explicitly available we are ready to compute $(v_{(s_{\text{mid}},p_2),\gamma})_{\mathcal{S}_{\text{ref}} \times \mathcal{T}}$ via (3.4), where the integral over $\mathcal{L}_{(s,t)}$ in (3.5) is approximated by the trapezoidal sum. For instance, let $K = \Delta M^q$ ($\beta = 0$ in (2.11)). Then, with $e_\gamma$ from (3.3) we have that

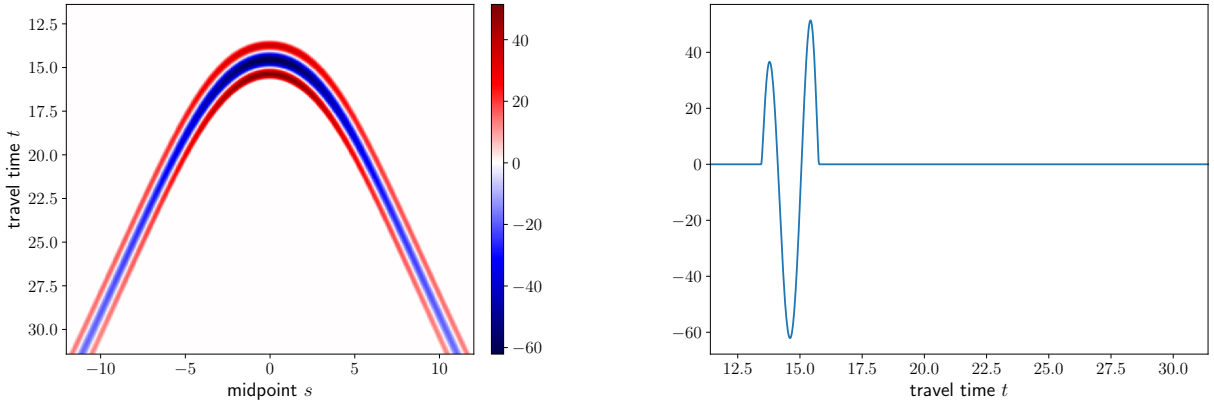
$$K^* e_\gamma(\boldsymbol{x} - \boldsymbol{p}) = C_{k,\gamma} \begin{cases} x_2^q \, 4k \left(\gamma^2 - |\boldsymbol{x} - \boldsymbol{p}|^2\right)^{k-2} \left(k\,|\boldsymbol{x} - \boldsymbol{p}|^2 - \gamma^2\right) & : \ |\boldsymbol{x} - \boldsymbol{p}| < \gamma, \\ 0 & : \ |\boldsymbol{x} - \boldsymbol{p}| \geq \gamma, \end{cases}$$

for $k \geq 2$. Corresponding reference kernels are displayed in Figure 1 for $k = 3$ and $q = 2$.

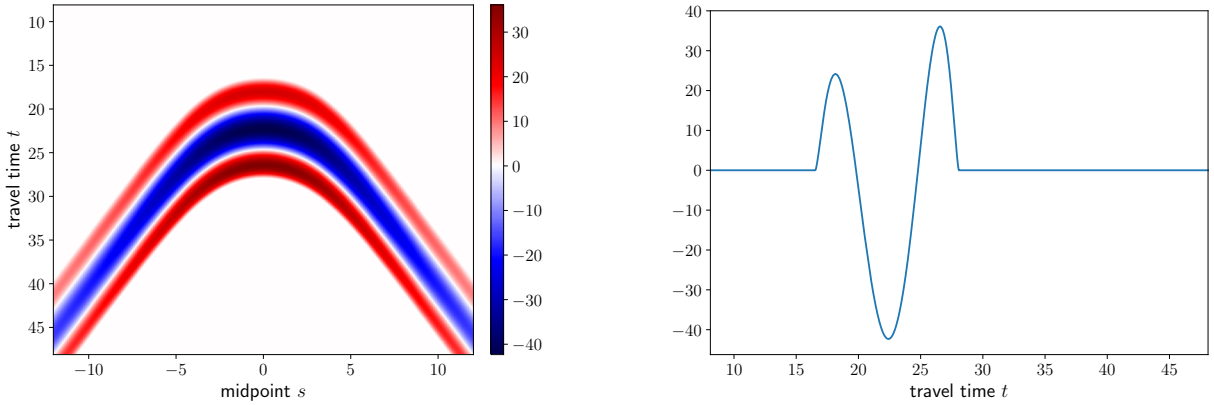
To speed up the kernel calculation, we use parallel computing with shared arrays $\varphi_{\mathcal{M}}$, $I_{\mathcal{M}}$, and $\mathcal{S}_{\text{ref}}$. Here, we avoid repeated computation of the $t$-contour by assigning a single job to the calculation of one row of $(v_{(s_{\text{mid}},p_2),\gamma})_{\mathcal{S}_{\text{ref}} \times \mathcal{T}}$. This is possible because the extension of $I_{\mathcal{M}}$ to $\mathcal{L}_{(s,t)}$ by interpolation is independent of $s \in \mathcal{S}_{\text{ref}}$. Consequently, for each pair of $t \in \mathcal{T}$ and $p_2 \in P_{\text{ref}}$ we start a job that calculates $(v_{(s_{\text{mid}},p_2),\gamma})_{\mathcal{S}_{\text{ref}} \times \mathcal{T}}[:, i_t]$, where $i_t$ is the index of $t$ in $\mathcal{T}$.

[2]For $x \in \mathbb{R}$: $\lceil x \rceil = \min\{m \in \mathbb{Z} : x \leq m\}$.

[3]We used the implementation `skimage.measure.find_contours` from the module `skimage`.




(A) $c(\boldsymbol{x}) = 0.1x_2 + 0.5$.




(B) $c(\boldsymbol{x}) = 0.5 + 0.25\cos\left(\frac{\pi}{4}x_2\right)$.

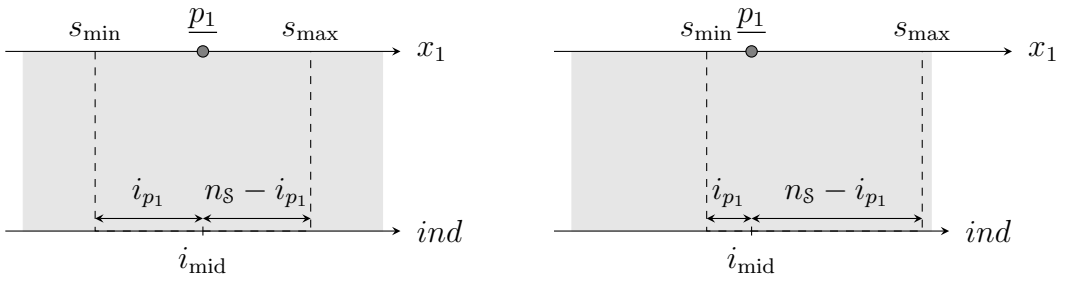
FIGURE 1. Reference kernels approximating $v_{(0,4),0.8}$ for $\alpha = 3$ and $K = \Delta M^2$ with respect to linear and oscillating velocities (both velocity models are used for numerical experiments in Section 5). The cross sections on the right are taken for $s = 0$.

From these reference kernels we finally obtain $(v_{\boldsymbol{p},\gamma})_{\mathcal{S}\times\mathcal{T}}$ at any position $\boldsymbol{p} \in \mathcal{M}_p$ by means of (3.7) and piecewise linear interpolation with respect to $p_2$, see next section for details.

**Remark 4.1.** *Above, we have basically described an implementation of the generalized Radon transform $F$ (2.4) via (3.5) applied to the function $K^*e_\gamma(\cdot - \boldsymbol{p})$. So, using the same concepts, we can compute $Fu(s,t)$ at any $(s,t) \in Y$ and for any function $u$ for which we have an explicit expression. If there is no dependence on additional variables besides $s$ and $t$, each job is assigned to a single $t$.*

4.3.3. *Implementation of the approximate inverse.* Based on the reference kernels we next show how to evaluate the inner products defining the approximate inverse $\Lambda_\gamma$ in (3.1).

Recall that we want to compute $\Lambda_\gamma n$ on $\mathcal{M}_p$ for given data $g$ and a chosen cutoff function $\psi$. We assume $\mathcal{M}_p$ to be a Cartesian mesh whose range for the $x_1$-coordinate is contained in $[s_{\min}, s_{\max}]$ and the range for the depth-coordinate matches $[\min P_{\text{ref}}, \max P_{\text{ref}}]$. To get the (approximated) reconstruction kernel for a point $\boldsymbol{p} = (p_1, p_2) \in \mathcal{M}_p$, we first determine $p_{\text{l}}, p_{\text{r}} \in P_{\text{ref}}$ such that $p_{\text{l}} \le p_2 < p_{\text{r}}$. The kernel $(v_{(s_{\text{mid}},p_2),\gamma})_{\mathcal{S}_{\text{ref}}\times\mathcal{T}}$ will then be




(A) $\underline{p_1} = s_{\mathrm{mid}}$. (B) $\underline{p_1} < s_{\mathrm{mid}}$.

FIGURE 2. Examples for clipping the reference kernels. The light-gray box represents the reference kernel on $\mathcal{S}_{\mathrm{ref}} \times \mathcal{T}$, while the area within the dashed lines is the correct part for $(v_{\boldsymbol{p},\gamma})_{\mathcal{S}\times\mathcal{T}}$ where $\boldsymbol{p} = (p_1, p_2) \in \mathcal{M}_p$.

approximated by linear interpolation of the two corresponding reference kernels:

$$(v_{(s_{\mathrm{mid}},p_2),\gamma})_{\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}} := \frac{p_{\mathrm{r}} - p_2}{p_{\mathrm{r}} - p_{\mathrm{l}}} (v_{(s_{\mathrm{mid}},p_{\mathrm{l}}),\gamma})_{\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}} + \frac{p_2 - p_{\mathrm{l}}}{p_{\mathrm{r}} - p_{\mathrm{l}}} (v_{(s_{\mathrm{mid}},p_{\mathrm{r}}),\gamma})_{\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}}.$$

Now we need to cut out $(v_{\boldsymbol{p},\gamma})_{\mathcal{S}\times\mathcal{T}}$ from this array transferring the shift relation (3.7) to the array concept. Note that the shifted argument $s - p_1$ is not a grid point of $\mathcal{S}_{\mathrm{ref}}$ in general, even for $s \in \mathcal{S}$. In a first step we therefore replace $p_1$ by $\underline{p_1} = \mathcal{S}[i_{p_1}]$, the next smaller entry in $\mathcal{S}$, that is,

$$i_{p_1} = \left\lfloor \frac{p_1 - \mathcal{S}[0]}{h_{\mathcal{S}}} \right\rfloor. \tag{4.5}$$

[4]

For example, if $\mathcal{S} = \mathcal{I}([-1,1],21) = [-1,-0.9,\dots,0.9,1]$, we have $\underline{-0.8} = -0.8$ with $i_{-0.8} = 2$ and $\underline{0.13} = 0.1$ with $i_{0.13} = 11$.

Define $p_{\mathrm{ref}} = \max_{\boldsymbol{p}\in\mathcal{M}_p} |s_{\mathrm{mid}} - \underline{p_1}|$ and set $n_{\mathrm{ref}} = n_{\mathcal{S}} + 2p_{\mathrm{ref}}/h_{\mathcal{S}}$ (note that $p_{\mathrm{ref}}/h_{\mathcal{S}} \in \mathbb{N}$). Then, it is sufficient to calculate the reference kernels on the extended grid $\mathcal{S}_{\mathrm{ref}} := \mathcal{I}([s_{\min} - p_{\mathrm{ref}}, s_{\max} + p_{\mathrm{ref}}], n_{\mathrm{ref}})$. In fact, we have

$$(v_{\boldsymbol{p},\gamma})_{\mathcal{S}\times\mathcal{T}} = (v_{(s_{\mathrm{mid}},p_2),\gamma})_{\mathcal{S}_{\mathrm{ref}}\times\mathcal{T}}[i_{\mathrm{mid}} - i_{p_1} : i_{\mathrm{mid}} - i_{p_1} + n_{\mathcal{S}}, :] \quad \text{where} \quad i_{\mathrm{mid}} = \frac{n_{\mathcal{S}} - 1}{2}, \tag{4.6}$$

which is the array operation accounting for (3.7). Figure 2 illustrates this slicing operation for different $p_1$.

Based on the reconstruction kernel $(v_{\boldsymbol{p},\gamma})_{\mathcal{S}\times\mathcal{T}}$ we now describe the implementation of the approximate inverse $\Lambda_\gamma$ (3.1). Without loss of generality we assume the data $g$ to be measured on a Cartesian grid, say on $\mathcal{S}_{\mathrm{data}} \times \mathcal{T}_{\mathrm{data}}$, where $\mathcal{S}_{\mathrm{data}} = \mathcal{I}([s_{\min}, s_{\max}], n_{\mathcal{S}_{\mathrm{data}}})$ and $\mathcal{T}_{\mathrm{data}} = \mathcal{I}([t_{\min}, t_{\max}], n_{\mathcal{T}_{\mathrm{data}}})$. Thus, we consider $g \in \mathbb{R}^{n_{\mathcal{S}_{\mathrm{data}}} \times n_{\mathcal{T}_{\mathrm{data}}}}$ in what follows. Further, we assume that the reference kernels have at least the resolution of the data: $n_{\mathcal{S}_{\mathrm{data}}} \le n_{\mathcal{S}}$ and $n_{\mathcal{T}_{\mathrm{data}}} \le n_{\mathcal{T}}$. In doing so, we address a situation that can occur in practice when only relatively few measurements can be taken. If we would compute the reference kernels on the same rough data grid $\mathcal{S}_{\mathrm{data}} \times \mathcal{T}_{\mathrm{data}}$, the nearest neighbor interpolation (4.6) via (4.5) could affect the reconstructions, see Figure 9 for a numerical demonstration. Since data and kernels are given on different grids, we need an operator $\Pi\colon \mathbb{R}^{n_{\mathcal{S}}\times n_{\mathcal{T}}} \to \mathcal{C}([s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}])$ which interpolates the kernels, for instance, we use piecewise bilinear interpolation in Section 5.2.

[4] For $x \in \mathbb{R}$: $\lfloor x \rfloor = \max\{m \in \mathbb{Z}\colon m \le x\}$.

Finally, we get $\Lambda_\gamma n \approx \widetilde{\Lambda}_\gamma n$ by

$$\widetilde{\Lambda}_\gamma n(\boldsymbol{p}) \coloneqq h_{\mathcal{S}_{\text{data}}} h_{\mathcal{T}_{\text{data}}} \sum_{\mathcal{S}_{\text{data}}} \sum_{\mathcal{T}_{\text{data}}} \psi|_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}\; g\,(\Pi \upsilon_{\boldsymbol{p},\gamma})|_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}} \quad \text{for any } \boldsymbol{p} \in \mathcal{M}_p. \tag{4.7}$$

Above we first have an element-wise multiplication of three arrays and then we take the sum over rows and columns.

## 5. Numerical experiments

In this section we present numerical experiments to demonstrate the performance of our inversion scheme in several test scenarios using the common offset scanning geometry (2.10) with offset $\alpha \geq 0$. We compute $\widetilde{\Lambda}_\gamma n$ (4.7) from consistent data under the geometric optics assumption and under a violation thereof, and from inconsistent data generated by a wave solver. The needed reference kernels are computed with respect to the mollifier $e_\gamma$ from (3.3) for $k = 3$ on the grid $\mathcal{S} \times \mathcal{T}$ (4.4) in the data space $Y$ (2.8) with parameters $s_{\min}$, $s_{\max}$, $n_{\mathcal{S}}$, and $t_{\min}$, $t_{\max}$, $n_{\mathcal{T}}$.

In both test cases, for truncating the data by a cutoff function $\psi \in \mathcal{C}_0^\infty(Y)$, we modify a function proposed in [19, Section 5], see also [9, Section 4], such that

$$\operatorname{supp}\psi = [s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}], \quad \psi|_{[s_{\min}+0.5, s_{\max}-0.5]\times[t_{\min}, t_{\max}-0.5]} = 1.$$

See Appendix A for an explicit expression of $\psi$.

Our implementation in the `Python` programming language, which we used to execute the numerical experiments of this section, is freely available, see [7].

### 5.1. Consistent data.

The data $g$ here are generated by an application of the GRT (2.4) to the function[5]

$$n = \chi_{B_2((0,5),2)} - \chi_{B_2((0,5),1)} + \chi_{B_\infty((3,6),1.25)} + \chi_{\{\boldsymbol{x}\in X\colon\, x_2 \geq 6.5+\sin(\pi x_1/2)\}}, \tag{5.1}$$

which is a linear combination of indicator functions of circular and rectangular disks and of a half space with a sine-like boundary, see Figure 3 (left). We emphasize strongly that both, data and kernels, are given on the same grid throughout this subsection: $\mathcal{S}_{\text{data}} = \mathcal{S}$ and $\mathcal{T}_{\text{data}} = \mathcal{T}$. As a consequence, the interpolation operator $\Pi$ in (4.7) needs not to be specified and may in fact be omitted. According to Remark 4.1, $g = (Fn)_{\mathcal{S}\times\mathcal{T}}$ can numerically be evaluated just as the reconstruction kernels, see Figure 3 (right) for an illustration.

Further, we use the offset $\alpha = 5$ if not stated otherwise.

#### 5.1.1. *Geometric optics assumption applies.*

The background wave velocity here is linearly increasing: $c(\boldsymbol{x}) = \mathrm{m}x_2 + \mathrm{b}$, $\mathrm{m}, \mathrm{b} > 0$. It satisfies the geometric optics assumption and the corresponding imaging operator $\Lambda$ (2.9) with $K$ from (2.11) is an elliptic pseudodifferential operator of order 1 over the domains

$$X_0 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_2 > x_{\min}\right\} \quad \text{and} \quad Y = S_0 \times ]t_{\text{first}}, \infty[$$

where

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

[5] By $B_q(\boldsymbol{c}, r)$ we denote the (closed) $q$-norm ball with center $\boldsymbol{c}$ and radius $r$.

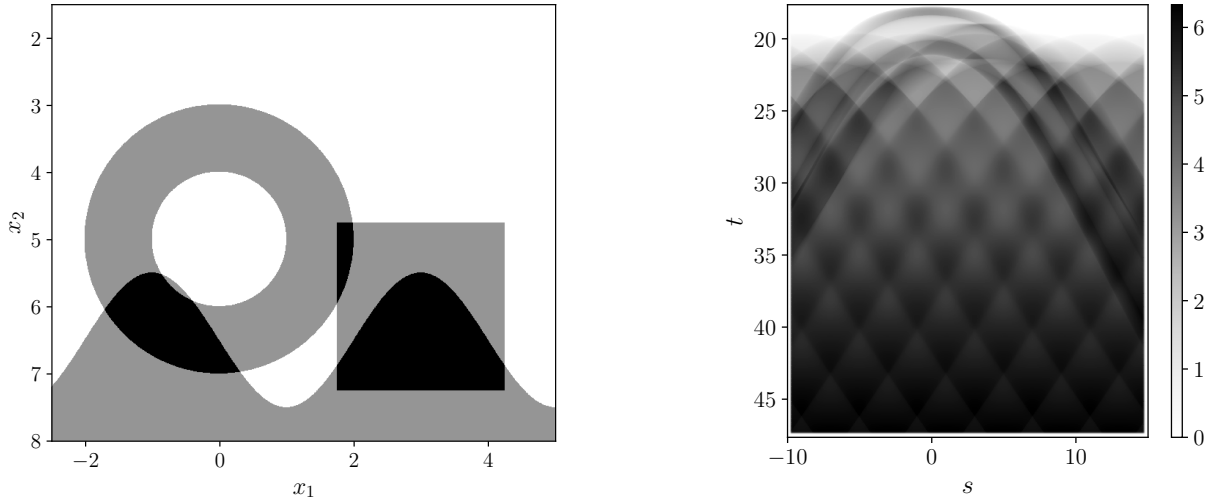



FIGURE 3. Left: illustration of the function $n$ from (5.1) with the following color coding: white, grey, and black represent the numerical values 0, 1, and 2, respectively. Right: corresponding processed data $\psi|_{\mathcal{S}\times\mathcal{T}}\, g = \psi|_{\mathcal{S}\times\mathcal{T}}\,(Fn)_{\mathcal{S}\times\mathcal{T}}$ for the linear velocity model $c(\boldsymbol{x}) = 0.1x_2 + 0.5$ and offset $\alpha = 5$. The reconstruction on the left of Figure 4 has been computed from these data.

see Section 2.1. If $S_0$ is sufficiently large all singularities (wave fronts) of $n$ are in fact visible in $\Lambda n$ [12, Proposition 3.5 and Remark 3.7][6]. The numerical approximation $\widetilde{\Lambda}_{0.2}n$ displayed in Figure 4 (left) clearly confirms the conservation and enhancement of the singularities. In contrast, the image on right of Figure 4 misses all singularities with a horizontal (normal) direction. Again $\widetilde{\Lambda}_{0.2}n$ is shown, however, for the constant background velocity $c(\cdot) = 1$. It is known that singularities with horizontal directions are not visible in $\Lambda n$ for this velocity, regardless of the specific choice of $\psi$ and $K$, see [8].

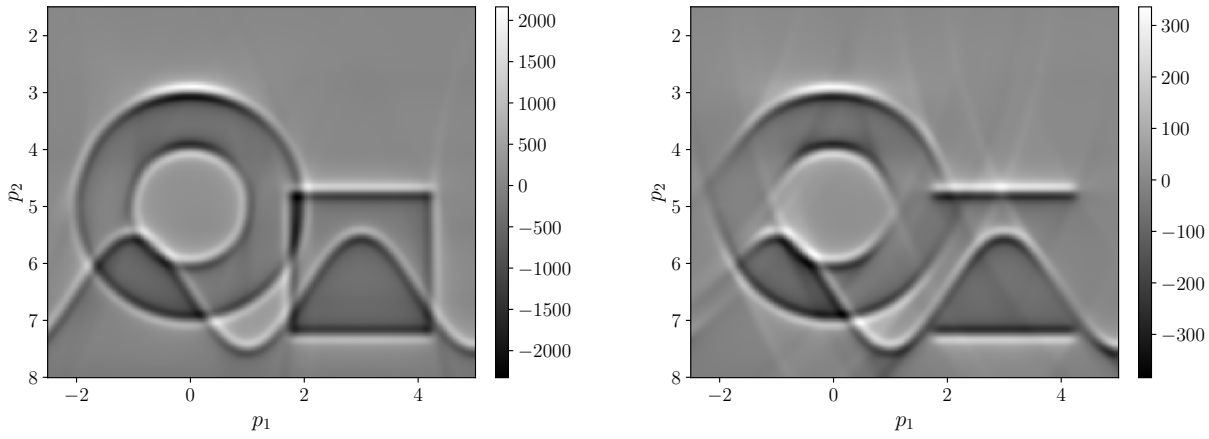



FIGURE 4. Reconstruction $\widetilde{\Lambda}_{0.2}n$ of $\Lambda n$ from data $g = (Fn)_{\mathcal{S}\times\mathcal{T}}$. Left: linear velocity model $c(\boldsymbol{x}) = 0.1x_2 + 0.5$ with $K = \Delta M^2$ and $\mathcal{T} = \mathcal{I}([17.64, 47.64], 3001)$, $\mathcal{S} = \mathcal{I}([-10, 15], 2501)$. Right: constant velocity model $c(\cdot) = 1$ with $K = \Delta M$ and $\mathcal{T} = \mathcal{I}([10.5, 40.5], 3001)$, $\mathcal{S} = \mathcal{I}([-10, 15], 2501)$. For both reconstructions, 130 reference kernels have been computed: $P_{\text{ref}} = \mathcal{I}([1.5, 8], 130)$.

[6]The mathematically correct statement reads $\mathrm{WF}^r(u) = \mathrm{WF}^{r-1}(\Lambda u)$ for any distribution $u$ with compact support in $X$. Here, $\mathrm{WF}^r(u)$, $r \in \mathbb{R}$, denotes the $H^r$-wave front set of $u$.

We have computed two further reconstructions for the linear velocity model $c(\boldsymbol{x}) = 0.1x_2+0.5$. In Figure 5 (left) we chose the same setting as for the reconstruction displayed in Figure 4 (left), however, the data used $g^\varepsilon$ have been artificially disturbed by noise:

$$g^\varepsilon = g + \varepsilon \|g\|_\star \frac{\mathrm{N}}{\|\mathrm{N}\|_\star}, \qquad \varepsilon > 0, \tag{5.2}$$

where N is an $n_{\mathcal{S}} \times n_{\mathcal{T}}$ array containing uniformly distributed random numbers in $[-1, 1]$ and $\|\cdot\|_\star$ denotes the Frobenius norm. So, $\varepsilon$ is the relative noise level with respect to $\|\cdot\|_\star$.

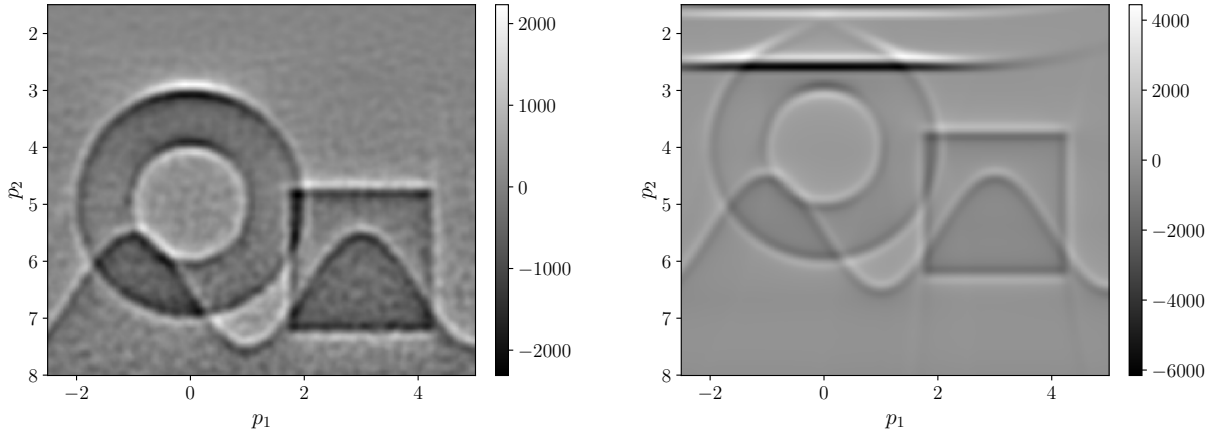



FIGURE 5. Reconstructions for background velocity $c(\boldsymbol{x}) = 0.1x_2 + 0.5$. Left: reconstruction $\widetilde{\Lambda}_{0.2}n$ from noisy data $g^\varepsilon$ (5.2) with $\varepsilon = 10\%$. All other settings are as in Figure 4 (left). Right: reconstruction $\widetilde{\Lambda}_{0.2}n^*$ from data $g = (Fn^*)_{\mathcal{S}\times\mathcal{T}}$ for $n^*(x_1, x_2) = n(x_1, x_2 + 1)$ with $n$ from (5.1). Here, we used a slightly different cutoff function which satisfies $\psi|_{[s_{\min}+0.5, s_{\max}-0.5]\times[t_{\min}+0.1, t_{\max}-0.5]} = 1$. The other settings agree with those of Figure 4 (left).

Finally, in Figure 5 (right) we present a reconstruction where $\Lambda$ fails to be a pseudodifferential operator. To this end we generate the data $(Fn^*)_{\mathcal{S}\times\mathcal{T}}$ where $n^*$ is a version of $n$ (5.1) shifted towards the surface: $n^*(x_1, x_2) = n(x_1, x_2 + 1)$. The support of $n^*$ intersects slightly the strip $\mathbb{R}^2_+ \backslash X_0 = \{\boldsymbol{x} \in \mathbb{R}^2 : 0 < x_2 \leq x_{\min} \approx 2.07\}$, over which $\Lambda$ is "only" a Fourier integral operator. Fourier integral operators might add artifacts (to be precise: the wave front set of $\Lambda n^*$ might be larger than the wave front set of $n^*$). Exactly this phenomenon can be observed in the reconstruction $\widetilde{\Lambda}_{0.2}n^*$ which exhibits a strong horizontally elongated singularity at about $x_2 = 2.5$ that is not present in $n^*$.

5.1.2. *Geometric optics assumption is violated.* Figure 6 contains two reconstructions of $\Lambda n$ for two slightly different background velocities

$$c_1(\boldsymbol{x}) = \frac{1}{2}\left(1 + x_2 + \frac{1}{2}\cos\left(\frac{\pi}{4}x_2\right)\right) \tag{5.3}$$

and

$$c_2(\boldsymbol{x}) = \frac{1}{2}\left(1 + \frac{1}{2}\cos\left(\frac{\pi}{4}x_2\right)\right). \tag{5.4}$$

Velocity $c_1$ violates the geometric optics assumption since different rays intersect (even though this velocity is strictly increasing). In case of $c_2$, the geometric optics assumption is satisfied but $\Lambda$ fails to be a pseudodifferential operator over the support of $n$. We

do not have rigorous proofs, but the numerically computed rays and isochrones strongly support these statements, see Appendix B.

In both cases the Fast Marching algorithm for solving the eikonal equation terminates and our reconstruction algorithm delivers a decent reconstruction for $c_1$ but very strong, added artifacts corrupt the reconstruction for $c_2$.

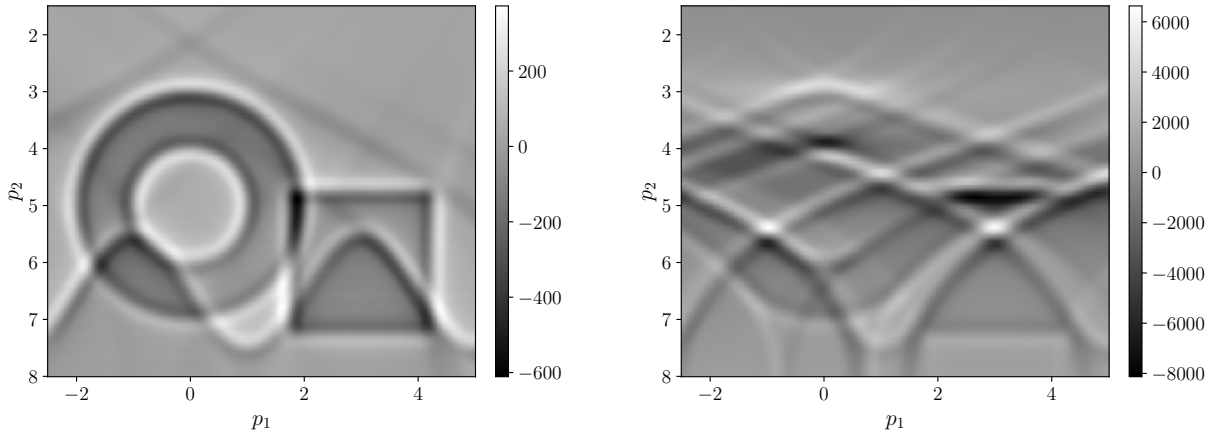



FIGURE 6. Reconstructions $\widetilde{\Lambda}_{0.3}n$ with $K = \Delta M^2$ from data $g = (Fn)_{\mathcal{S}\times\mathcal{T}}$ and $\alpha = 3$. Left: velocity model $c_1$ (5.3), $\mathcal{T} = \mathcal{I}([6.2, 15.2], 901)$, $\mathcal{S} = \mathcal{I}([-12, 12], 2401)$. Right: velocity model $c_2$ (5.4), $\mathcal{T} = \mathcal{I}([8.1, 98.1], 9001)$, $\mathcal{S} = \mathcal{I}([-20, 20], 2001)$. For both reconstructions, 86 reference kernels have been computed: $P_{\text{ref}} = \mathcal{I}([1.5, 8], 86)$

5.2. **Data from the wave equation.** In this subsection, we generate the data $g$ as input to (4.7) by the right-hand side of (2.3), that is, we numerically solve the wave equation (2.1) for $u$ and $\widetilde{u}$. In view of (1.1), for a given background velocity $c$ we set

$$\nu_{\text{pr}}^2(\boldsymbol{x}) = \frac{c^2(\boldsymbol{x})}{1 + \lambda\, n(\boldsymbol{x})}, \tag{5.5}$$

where $n$ is as in (5.1), see Figure 3 left, and the parameter $\lambda > 0$ scales the committed linearization error. By varying $\lambda$ we stress the validity of the linear model (2.3) and (2.4).

In (2.1) we replace the source term by a scaled and time-shifted Gaussian pulse. The resulting equation is then solved on the computational domain $[-15, 20] \times [0, 15]$ using the wave solver of the open source toolbox PySIT [4] with a step size $1/22$ in each direction. To suppress spurious reflections, the computational domain is equipped with an absorbing boundary by a perfectly matched layer (PML). With $\nu_{\text{pr}}$ and $c$ as input for the wave solver, we compute the seismograms

$$u(t; \boldsymbol{x_r}(s), \boldsymbol{x_s}(s)) \quad \text{and} \quad \widetilde{u}(t; \boldsymbol{x_r}(s), \boldsymbol{x_s}(s)) \qquad \text{for} \quad (s,t) \in \mathcal{S}_{\text{data}} \times \mathcal{T}_{\text{data}},$$

where $\mathcal{S}_{\text{data}} = \mathcal{I}([s_{\min}, s_{\max}], n_{\mathcal{S}_{\text{data}}})$, $n_{\mathcal{S}_{\text{data}}}$ odd, and $\mathcal{T}_{\text{data}} = \mathcal{I}([t_{\min}, t_{\max}], n_{\mathcal{T}_{\text{data}}})$.

**Remark 5.1.** *Note that $n_{\mathcal{T}_{\text{data}}}$ will be set by the wave solver to satisfy the CFL condition and will therefore be larger for $u$. For this reason we interpolate $\widetilde{u}$ piecewise linear on the time grid of $u$.*

Some post-processing of the seismograms $u$ and $\widetilde{u}$ is required: we account for the time-shift of the Gaussian pulse by a corresponding time-shift of the seismograms. Further, we fit the seismograms $u$ and $\widetilde{u}$ by scaling $\widetilde{u}$ so that the maximal amplitudes of both match.

Approximating the integral on the right-hand side of (2.3) finally by the trapezoidal sum yields the data $g$ for our reconstruction scheme. In this subsection, if data and kernels exist on different grids, the operator $\Pi$ in (4.7) interpolates piecewise bilinear.

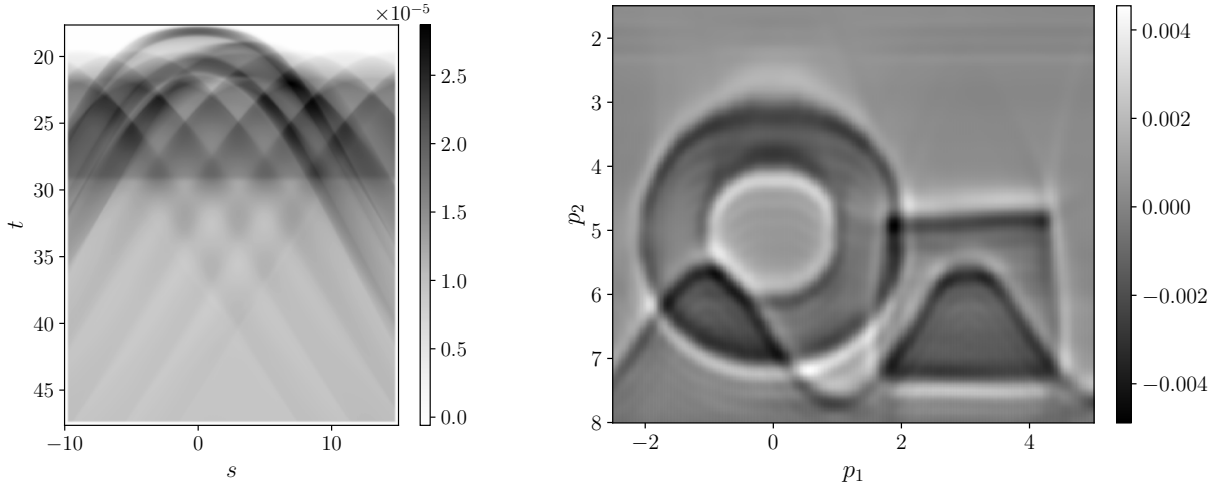



FIGURE 7. Left: processed wave data $\psi|_{\mathcal{S}_{\text{data}}\times\mathcal{T}_{\text{data}}}g$ where $c(\boldsymbol{x}) = 0.1x_2 + 0.5$, $\lambda = 0.05$ in (5.5), $\alpha = 5$, $\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 1001)$, and $\mathcal{T}_{\text{data}} = \mathcal{I}([17.64, 47.64], 7729)$. Right: corresponding reconstruction $\widetilde{\Lambda}_{0.2}n$ with $K = \Delta M^2$. The reference kernels were computed on $P_{\text{ref}} = \mathcal{I}([1.5, 8], 130)$ for $\mathcal{S} = \mathcal{I}([-10, 15], 1501)$ and $\mathcal{T} = \mathcal{T}_{\text{data}}$.

For our first experiments we chose $c(\boldsymbol{x}) = 0.1x_2 + 0.5$ as background velocity and $\alpha = 5$ as offset. On the left of Figure 7 one sees the processed data from the wave solver where $\lambda = 0.05$ in (5.5). Here, the linearization error is rather small, so that these data are qualitatively comparable with those on the right of Figure 3. The resulting reconstruction for $\Lambda n$ is shown on the right.

**Remark 5.2.** *The wave data displayed on the left of Figure 7 exhibit a sudden decrease about along the line $t \approx 29$. This effect originates from reflections at the bottom of the computational domain, even though PML boundary conditions are implemented ($t = 29.012$ is exactly the time it takes for the wavefront reflected at the bottom to travel from the source to the receiver in the medium with wave speed $c$). A remedy is to increase the computational domain at the cost of much more computing time for generating the data. One could also shrink the observation period, say to $[17.64, 29]$, but then not all singularities of $n$ are visible in the selected reconstruction area.*

With Figure 8 we demonstrate how the reconstructions deteriorate with increasing $\lambda$, that is, with an increasing linearization error.

So far in this section, we have presented reconstructions from finely sampled data to study purely the impact of the wave data (data inconsistency) and the nonlinearity. Left in Figure 9 we now show a reconstruction based on a larger sampling rate for the seismograms, namely $h_{\mathcal{S}_{\text{data}}} = 0.25$, which is ten times higher than the sampling rates used for Figures 7 and 8. The obtained quality is remarkably good, however, some high-frequency artifacts appear, even though we increased $\gamma$ to 0.3. Further increasing $\gamma$ would mitigate the artifacts but at the same time blur the image.[7] As before we have computed

[7]See [9, Remark 4.1] for a discussion on how to set the numerical value for $\gamma$.

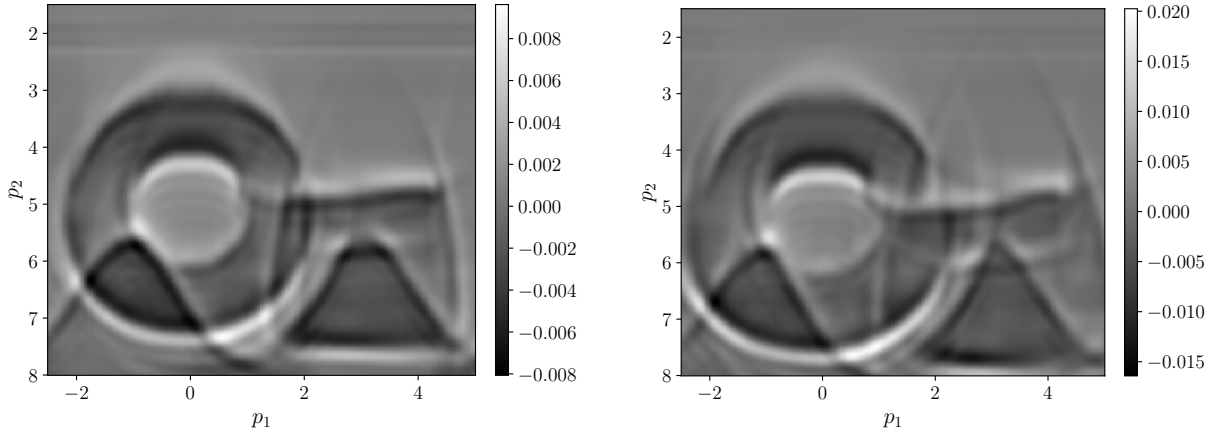



FIGURE 8. Reconstructions for different values of $\lambda$ in (5.5). All other settings are those of Figure 7. Left: $\lambda = 0.1$. Right: $\lambda = 0.2$.

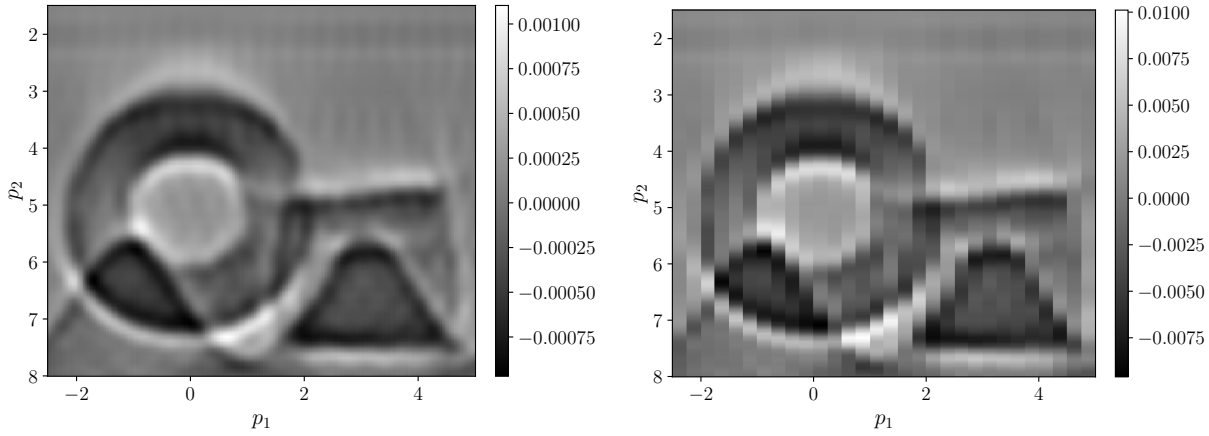



FIGURE 9. Reconstructions $\widetilde{\Lambda}_{0.3}n$ from low resolution wave data: $\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 101)$, $\mathcal{T}_{\text{data}} = \mathcal{I}([17.64, 47.64], 7729)$. Further, $c(\boldsymbol{x}) = 0.1x_2 + 0.5$, $\alpha = 5$, $K = \Delta M^2$, and $\lambda = 0.1$ in (5.5). Left: $\mathcal{S} = \mathcal{I}([-10, 15], 1001)$, $\mathcal{T} = \mathcal{T}_{\text{data}}$, and $P_{\text{ref}} = \mathcal{I}([1.5, 8], 86)$. Right: $\mathcal{S} = \mathcal{S}_{\text{data}}$, $\mathcal{T} = \mathcal{T}_{\text{data}}$, and $P_{\text{ref}} = \mathcal{I}([1.5, 8], 86)$.

the reference kernels with a higher resolution than the data. If we dispense with this and calculate the kernels with the resolution of the data, we obtain the coarse-grained image on the right of Figure 9.

We end this section with numerical examples for the velocity models (5.3) and (5.4), both of which violate the geometric optics assumption. The reconstructions in Figure 10 agree rather well with their counterparts from consistent and higher resolution data in Figure 6.

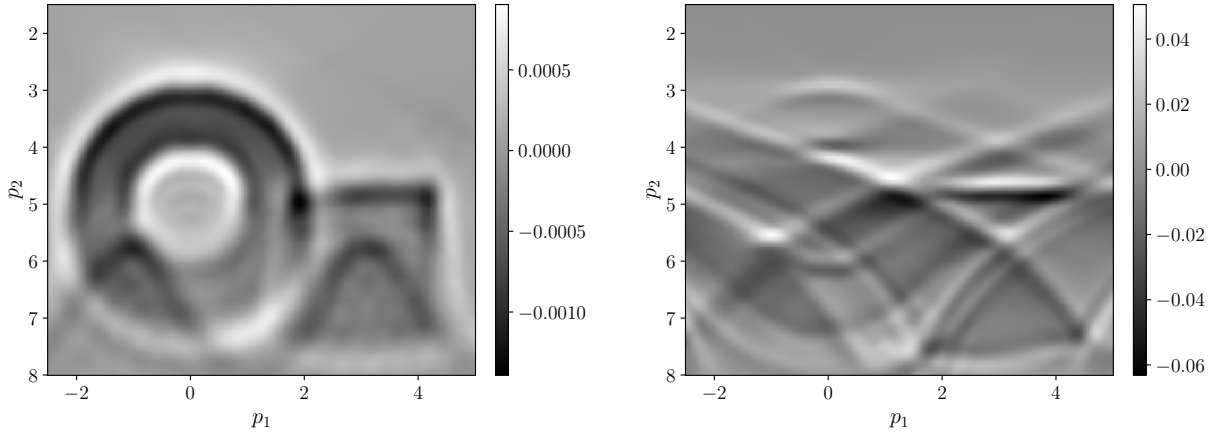



FIGURE 10. Reconstructions $\widetilde{\Lambda}_{0.3}n$ with $K = \Delta M^2$, $\alpha = 3$, and $\lambda = 0.1$ in (5.5). Left: velocity model $c_1$ (5.3), $\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([6.2, 15.2], 9262)$, $\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 1001)$, $\mathcal{S} = \mathcal{I}([-10, 15], 1501)$. Right: velocity model $c_2$ (5.4), $\mathcal{T} = \mathcal{T}_{\text{data}} = \mathcal{I}([8.1, 98.1], 8910)$, $\mathcal{S}_{\text{data}} = \mathcal{I}([-10, 15], 1001)$, $\mathcal{S} = \mathcal{I}([-10, 15], 1501)$. For both reconstructions, 86 reference kernels have been computed: $P_{\text{ref}} = \mathcal{I}([1.5, 8], 86)$

## APPENDIX A. CUTOFF FUNCTION

The cutoff function $\psi$, used in our numerical experiments in Section 5, is defined as follows. It depends on several parameters which steer its shape, that is, the width of the transition zone from 0 to 1. We have that

$$\psi(s,t;\underline{L}_s,\overline{L}_s,\underline{R}_s,\overline{R}_s,\underline{L}_t,\overline{L}_t,\underline{R}_t,\overline{R}_t) = \phi\left(s,\underline{L}_s,\overline{L}_s,\underline{R}_s,\overline{R}_s\right)\phi\left(t,\underline{L}_t,\overline{L}_t,\underline{R}_t,\overline{R}_t\right),$$

where

$$\phi(r,\underline{L},\overline{L},\underline{R},\overline{R}) = \begin{cases} 0 & : \ r \le \underline{L} \text{ or } r \ge \overline{R}, \\ p(r,\underline{L},\overline{L}) & : \ \underline{L} < r < \overline{L}, \\ q(r,\underline{R},\overline{R}) & : \ \underline{R} < r < \overline{R}, \\ 1 & : \ \overline{L} \le r \le \underline{R}, \end{cases}$$

with

$$p\left(r,\underline{R},\overline{R}\right) = \frac{f\left(r-\underline{R}\right)}{f\left(r-\underline{R}\right)+f\left(\overline{R}-r\right)}, \quad q\left(r,\underline{R},\overline{R}\right) = \frac{f\left(\overline{R}-r\right)}{f\left(\overline{R}-r\right)+f\left(r-\underline{R}\right)},$$

and

$$f(r) = \begin{cases} \exp\left(-\frac{1}{r}\right) & : \ r > 0, \\ 0 & : \ r \le 0. \end{cases}$$

The last four parameters of $\phi$ define the lower and upper ranges where the smooth cutoff is performed. If not otherwise stated we used $\underline{L}_s = s_{\min}, \overline{L}_s = s_{\min} + 0.5$, $\underline{R}_s = s_{\max} - 0.5, \overline{R}_s = s_{\max}$ and $\underline{L}_t = t_{\min}, \overline{L}_t = t_{\min}$, $\underline{R}_t = t_{\max} - 0.5, \overline{R}_t = t_{\max}$.

## APPENDIX B. RAY SYSTEMS

In this appendix we provide numerical support for our statements in Section 5.1.2 that velocity $c_1$ (5.3) violates the geometric optics assumption while $c_2$ (5.4) satisfies it, but on the other hand yields an imaging operator $\Lambda$ that is not a pseudodifferential operator.

Seismic or optical rays are the (bi-)characteristic curves of the eikonal equation (2.6), see, e.g., [2, Appendix E.2.3]. We rely on the ray system (the running parameter $t \geq 0$ agrees with the travel time from the source point)

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = c(\boldsymbol{r})\frac{\boldsymbol{p}}{|\boldsymbol{p}|}, \quad \boldsymbol{r}(0) = \boldsymbol{x_s}; \qquad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\nabla c(\boldsymbol{r})}{c(\boldsymbol{r})}, \quad \boldsymbol{p}(0) = \frac{\boldsymbol{\xi}}{c(\boldsymbol{r}(0))},$$

with a unit vector $\boldsymbol{\xi} = (\xi_1, \xi_2)^\top \in S^1$. Here, $\boldsymbol{p}(t) = \nabla\tau(\boldsymbol{r}(t), \boldsymbol{x_s})$.

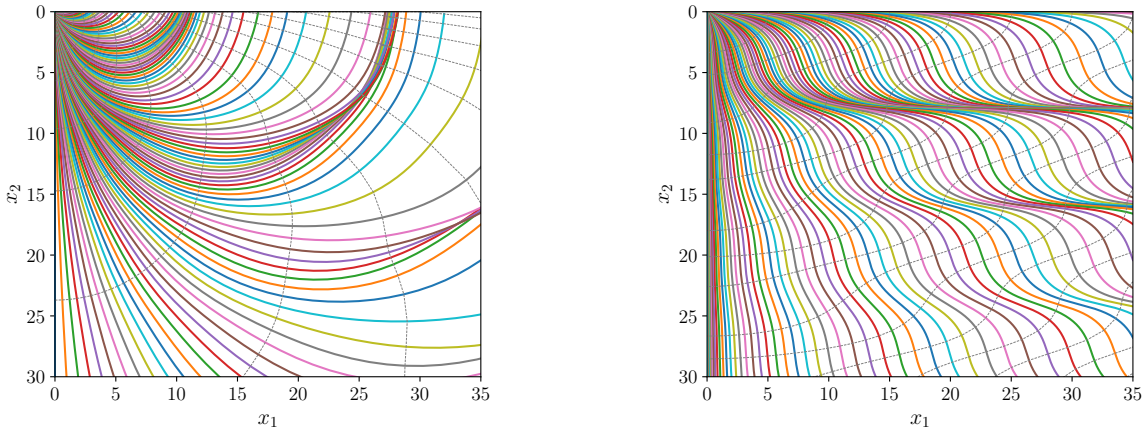



FIGURE 11. Seismic rays with origin as take-off point but different take-off directions (indicated by 10 colors that repeat cyclically). The dotted lines connect points with equal travel times (isochrones). Left: velocity model (5.3). Right: velocity model (5.4).

We solved the above system by a standard ODE solver for $\boldsymbol{x_s} = (0, 0)^\top$ and several $\boldsymbol{\xi}$ pointing downwards to the right of $\boldsymbol{x_s}$, that is, $\xi_1 \geq 0$ and $\xi_2 > 0$. In Figure 11 (left) the rays are plotted with respect to $c_1$. In the upper right corner, rays do intersect. This happens also in other parts of $X$. Contrary, the rays with respect to $c_2$, Figure 11 (right), do not intersect (at least we could not find intersection points numerically) but the rays converge to horizontal lines at depth values being integer multiples of 8 which is the period of $c_2$. They meet virtually as $x_1 \to \infty$. Further, isochrones for large travel times share the same normal direction at different points. For the experts: this observation, together with the relation $\partial_s\varphi = -\partial_{x_1}\varphi$ for the travel time function $\varphi$ defined in (2.5), strongly suggests $\Lambda$ not to be a pseudodifferential operator (the Fourier integral operator $F\colon \mathcal{E}'(X) \to \mathcal{D}'(Y)$ (2.4) presumably does not satisfy the Bolker condition for offset $\alpha = 0$). A rigorous proof would require an analytically explicit representation of $\varphi$.

## REFERENCES

[1] G. BEYLKIN, *Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized Radon transform*, J. Math. Phys., 26 (1985), pp. 99–108, http://dx.doi.org/10.1063/1.526755.

[2] N. BLEISTEIN, J. K. COHEN, AND J. W. STOCKWELL, JR., *Mathematics of multidimensional seismic imaging, migration, and inversion*, vol. 13 of Interdisciplinary Applied Mathematics, Springer-Verlag, New York, 2001, http://dx.doi.org/10.1007/978-1-4613-0001-4. Geophysics and Planetary Sciences.

[3] M. G. CRANDALL AND P.-L. LIONS, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19, https://doi.org/10.2307/2007396.

[4] L. DEMANET AND R. J. HEWETT, PySIT – *Seismic Imaging Toolbox for Python* v1.0. http://pysit.org.

[5] R. Felea, V. P. Krishnan, C. J. Nolan, and E. T. Quinto, *Common midpoint versus common offset acquisition geometry in seismic imaging*, Inverse Probl. Imaging, 10 (2016), pp. 87–102, https://doi.org/10.3934/ipi.2016.10.87.
[6] S. Fomel, S. Luo, and H. Zhao, *Fast sweeping method for the factored eikonal equation*, J. Computat. Phys., 228 (2009), pp. 6440–6455, https://doi.org/10.1016/j.jcp.2009.05.029.
[7] K. Ganster, *Approximate inversion of generalized Radon transforms - software package*, 2022, https://doi.org/10.35097/707.
[8] C. Grathwohl, *Seismic imaging with the elliptic Radon transform in 3D: analytical and numerical aspects*, PhD thesis, Karlsruhe Institute of Technology, 2020, https://doi.org/10.5445/IR/1000105093.
[9] C. Grathwohl, P. Kunstmann, E. T. Quinto, and A. Rieder, *Approximate inverse for the common offset acquisition geometry in 2D seismic imaging*, Inverse Problems, 34 (2018), p. 014002, https://doi.org/10.1088/1361-6420/aa9900.
[10] C. Grathwohl, P. Kunstmann, E. T. Quinto, and A. Rieder, *Microlocal analysis of imaging operators for effective common offset seismic reconstruction*, Inverse Problems, 34 (2018), pp. 114001, 24, https://doi.org/10.1088/1361-6420/aadc2a.
[11] C. Grathwohl, P. Kunstmann, E. T. Quinto, and A. Rieder, *Imaging with the elliptic Radon transform in three dimensions from an analytical and numerical perspective*, SIAM J. Imaging Sci., 13 (2020), pp. 2250–2280, https://doi.org/10.1137/20M1332657.
[12] P. C. Kunstmann, E. T. Quinto, and A. Rieder, *Seismic imaging with generalized Radon transforms: stability of the Bolker condition*, CRC 1173 Preprint 2022/3, Karlsruhe Institute of Technology, 2022, https://doi.org/10.5445/IR/1000141638. Pure and Applied Mathematics Quarterly (to appear).
[13] A. K. Louis, *Corrigendum: "Approximate inverse for linear and some nonlinear problems" [Inverse Problems* **11** *(1995), no. 6, 1211–1223]*, Inverse Problems, 12 (1996), pp. 175–190, http://dx.doi.org/10.1088/0266-5611/12/2/005.
[14] S. Luo and J. Qian, *Factored singularities and high-order Lax–Friedrichs sweeping schemes for point-source traveltimes and amplitudes*, J. Computat. Phys., 230 (2011), pp. 4742–4755, https://doi.org/10.1016/j.jcp.2011.02.043.
[15] S. Luo, J. Qian, and R. Burridge, *Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media in the high frequency regime*, J. Comput. Phys., 270 (2014), pp. 378–401, https://doi.org/10.1016/j.jcp.2014.03.066.
[16] S. Luo, J. Qian, and H. Zhao, *Higher-order schemes for 3D first-arrival traveltimes and amplitude*, Geophysics, 77 (2012), pp. T47–T56, https://doi.org/10.1190/geo2010-0363.1.
[17] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Numer. Anal., 28 (1991), pp. 907–922, https://doi.org/10.1137/0728049.
[18] J. Qian and W. W. Symes, *An adaptive finite-difference method for traveltimes and amplitudes*, Geophysics, 67 (2002), p. 167, https://doi.org/10.1190/1.1451472.
[19] E. T. Quinto, A. Rieder, and T. Schuster, *Local inversion of the sonar transform regularized by the approximate inverse*, Inverse Problems, 27 (2011), pp. 035006, 18, http://dx.doi.org/10.1088/0266-5611/27/3/035006.
[20] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM J. Numer. Anal., 29 (1992), pp. 867–884, https://doi.org/10.1137/0729053.
[21] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595, https://doi.org/10.1073/pnas.93.4.1591.
[22] J. A. Sethian, *Fast marching methods*, SIAM Rev., 41 (1999), pp. 199–235, https://doi.org/10.1137/S0036144598347059.
[23] W. W. Symes, *Mathematics of reflection seismology*, tech. report, The Rice Inversion Project, Rice University, Houston, TX, USA, 1998, http://www.trip.caam.rice.edu/downloads/preamble.pdf.
[24] E. Treister and E. Haber, *A fast marching algorithm for the factored eikonal equation*, J. Computat. Phys., 324 (2016), pp. 210–225, https://doi.org/10.1016/j.jcp.2016.08.012.
[25] Y.-T. Zhang, H.-K. Zhao, and J. Qian, *High order fast sweeping methods for static Hamilton-Jacobi equations*, J. Sci. Comput., 29 (2006), pp. 25–56, https://doi.org/10.1007/s10915-005-9014-3.

Department of Mathematics, Karlsruhe Institute of Technology (KIT), D-76128 Karlsruhe, Germany
*Email address*: kevin.ganster@kit.edu
*Email address*: andreas.rieder@kit.edu