University of Helsinki
Faculty of Science
Master's Programme in Mathematics and Statistics

Master's thesis

Iterative Time-Reversal Control for an Inverse Problem of the 1+1 Dimensional Wave Equation: A Computational Study

Daniel Repo

Supervisors:
Lauri Oksanen
Marvin Knöller

June 2025

Title: Iterative Time-Reversal Control for an Inverse Problem of the 1+1 Dimensional Wave Equation: A Computational Study
Author: Daniel Repo
Month and year: June 2025
Page count: 39 pp.

Abstract:

The problem of measuring acoustic wave speeds inside an object using boundary measurements is a physical example of an inverse boundary value problem. The inverse problem arises from the acoustic wave equation, and consists of recovering the wave speed function when the Neumann-to-Dirichlet map of the initial-boundary value problem is given.

This master's thesis demonstrates a computational implementation of the iterative time-reversal control method used to solve the inverse problem. We consider the case of one spatial dimension. In order to formulate the implementation, the theory underlying main regularization results by Korpela, Lassas and Oksanen is presented. Particular focus is on the Blagovestchenskii identities used to compute travel time volumes. A proof is given for the identities.

After considering the theoretical regularization method, a simulation of the forward problem using the finite difference method is detailed, followed by an implementation of a two-step regularization method. The results of iterative computation of regularized travel time volumes need to be differentiated in order to compute the wave speed function: To make the differentiation less sensitive to noise, total variation regularization is employed.

We observe that our results achieve noise tolerance at a level similar to an implementation by Korpela, Lassas and Oksanen based on more developed regularization theory.

Keywords: inverse problem, wave equation, numerical methods

Contents

Chapter 1

Introduction

Imaging is a physical task where the inner structure of an object is reconstructed based on boundary measurements of a phenomenon. Examples of imaging problems are X-ray imaging and electric impedance tomography (EIT), where the forward problems are modeled from X-ray attenuation and conductivity equations, respectively. Another imaging problem arises when pressure waves are used in measurements: These waves may be modeled using the acoustic wave equation.

In this thesis, we consider an inverse problem arising from a boundary value problem of the wave equation in one spatial dimension. This problem can be seen as a simple model of pressure wave imaging arising in geophysics and medical ultrasound applications. We demonstrate a computational implementation of a solution method, presenting also its theoretical foundations.

Our work uses the iterative time-reversal control method originally introduced by Bingham et al. [2] for three spatial dimensions, then reformulated by Korpela, Lassas and Oksanen [6] for the 1+1 dimensional problem. The latter article serves as the basis for our computational implementation as well as multiple proofs.

We begin by introducing background results in Chapter 2, focusing on topics that are not covered in introductory master's studies in mathematics. Following this, we formulate our problem statement and introduce the mathematical solution scheme in Chapter 3, then describe the computational scheme in Chapter 4. Afterwards, the results of the computation with added noise are examined in Chapter 5. Finally, we discuss the results in Chapter 6.

Chapter 2

Preliminaries

In this chapter, we record notations, theorems and algorithms used throughout the thesis.

2.1 Real Analysis and Partial Differential Equations

In this thesis, we assume that all scalar-valued functions take values in $\mathbb{R}$. We will use basic notions of L^p spaces with $p > 1$. We take it to be known that $L^2(\mathbb{R})$ is a Hilbert space (a complete inner product space) under the Lebesgue measure dx. For $S \subset \mathbb{R}$, we denote the inner product of $f, g \in L^2(S)$ by $\langle f, g\rangle_{L^2(S)}$. If we use a measure $d\eta$ that is not the Lebesgue measure, we specify this using the notation $\langle f, g\rangle_{L^2(S;d\eta)}$. We write $C^\infty(a, b)$ for smooth (infinitely continuously differentiable) functions on the closed interval $[a, b] \subset \mathbb{R}$. In addition, for $a, b \in \mathbb{R}$ we write

$$C_0^\infty(a, b) = \{f \in C^\infty(a, b) \mid \operatorname{supp}(f) \subset (a, b)\}.$$

Furthermore, for a Banach space E we define $\mathcal{L}(E)$ to be the set of bounded linear maps from E to itself. We typically denote the application of a linear operator A on a function f with juxtaposition: as Af. In cases of ambiguity, we use brackets: as $A[f]$.

Let us recall terminology from PDE theory. Suppose we have a cylinder $\Omega \coloneqq (0, +\infty) \times [0, +\infty)$. We denote points in Ω with the notation (t, x). A *Cauchy problem* for an unknown function u consists of a partial differential equation for u along with constraints for the values of u and $\partial_t u$ at a particular time. These constraints will be called *Cauchy data* in this thesis.

The formal equation $(\partial_t^2 - c(x)^2\partial_x^2)u(t, x) = 0$ for a given choice of $c(x)$ is the (homogeneous 1+1–dimensional) *wave equation*. The wave equation is a well-studied partial differential equation arising from the physical model of a vibrating string. Consider a Cauchy problem of the form

$$\begin{cases} (\partial_t^2 - c(x)^2\partial_x^2)u(t,x) = 0 & \text{in } \Omega, \\ u|_{t=0} = \partial_t u|_{t=0} = 0 & x \in [0,+\infty). \end{cases} \tag{2.1}$$

Let $f \in L^2([0,+\infty))$. Then a *Dirichlet boundary condition* on the Cauchy problem is an additional requirement of the form

$$u|_{x=0} = f, \quad t \in (0,+\infty).$$

A *Neumann boundary condition* is a requirement of the form

$$\partial_x u|_{x=0} = f, \quad t \in (0,+\infty).$$

We will correspondingly call f the *Dirichlet* or *Neumann data*. A Cauchy problem with boundary conditions will be termed an initial-boundary value problem.

2.1.1 Theory of Classical Solutions

Two classical solution formulas and a regularity theorem are recorded here. We emphasize that none of these results hold for the forward problem considered in this thesis. Rather, the contents of this section are used in proofs alongside limiting arguments to derive results related to the problem statement of Chapter 3: In particular, the Blagovestchenskii identities.

In one spatial dimension with constant wave speed $c \equiv 1$, a Cauchy problem on the real line with non-constant Cauchy data has an explicit solution known as *d'Alembert's formula*. For a proof, see [4, p.68: Thm.1].

Theorem 2.1.1 (d'Alembert's formula)**.** *Consider the region $D = (0,+\infty) \times \mathbb{R}$. Let $g \in C^2(\mathbb{R})$ and $h \in C^1(\mathbb{R})$. Then the Cauchy problem*

$$\begin{cases} (\partial_t^2 - \partial_x^2)u(t,x) = 0 & \textit{in } D, \\ u|_{t=0} = g & x \in \mathbb{R}, \\ \partial_t u|_{t=0} = h & x \in \mathbb{R}. \end{cases}$$

has the unique solution

$$u(t,x) = \frac{1}{2}\left(g(x-t) + g(x+t)\right) + \frac{1}{2}\int_{x-t}^{x+t} h(y)dy.$$

Moreover, $C^2(D)$. The Cauchy conditions are satisfied in the sense that for all $x^0 \in \mathbb{R}$, we have $u(t,x) \to g(x^0)$ and $\partial_t u(t,x) \to h(x^0)$ as $(t,x) \to (0,x^0)$, where $x \in \mathbb{R}$ and $t > 0$.

We remark that, in this thesis, d'Alembert's principle will only be used in the proof of Theorem 3.2.6 involving a problem with constant coefficients. Using the solutions of a family of these homogeneous Cauchy problems, one can create a solution of an inhomogeneous problem using *Duhamel's principle*. For a proof, see [4, p.81: Thm.4].

Theorem 2.1.2 (Duhamel's principle)**.** *Consider the region $D = (0, +\infty) \times \mathbb{R}$. Let $f \in C^1(D)$. For any $s \in \mathbb{R}$, $s > 0$, define $\tilde{u}(t, x; s)$ to be the solution of*

$$\begin{cases} (\partial_t^2 - \partial_x^2)\tilde{u}(t, x; s) = 0 & \text{in } (s, \infty) \times \mathbb{R}, \\ \tilde{u}(s, x; s) = 0 & x \in \mathbb{R}, \\ \partial_t \tilde{u}(s, x; s) = 0 & x \in \mathbb{R}. \end{cases}$$

Then the function $u \in C^2(D)$ defined by

$$u(t, x) := \int_0^t \tilde{u}(t, x; s) ds$$

solves the problem

$$\begin{cases} (\partial_t^2 - \partial_x^2) u(t, x) = f & \text{in } D, \\ u|_{t=0} = \partial_t u|_{t=0} = 0 & x \in \mathbb{R}. \end{cases}$$

where the Cauchy conditions are satisfied in the sense that for all $x^0 \in \mathbb{R}$, we have $u(t, x) \to 0$ and $\partial_t u(t, x) \to 0$ as $(t, x) \to (0, x^0)$, where $x \in \mathbb{R}$ and $t > 0$.

Finally, we record a regularity result used in several proofs.

Theorem 2.1.3. *Consider the region $\Omega = (0, +\infty) \times [0, +\infty)$. Let $c \in C^1(0, +\infty)$ and $f \in C_0^\infty(0, 2T)$. Then the problem*

$$\begin{cases} (\partial_t^2 - c^2(x)\partial_x^2) u(t, x) = 0 & \text{in } \Omega, \\ \partial_x u|_{x=0} = -f & t \in (0, 2T), \\ u|_{t=0} = \partial_t u|_{t=0} = 0 & x \in [0, +\infty), \end{cases}$$

has a unique solution $u \in C^\infty(\Omega)$.

The theorem can be proven by arguments similar to [4, p.416: Thm.7], which uses zero Dirichlet terms instead of $C_0^\infty(0, 2T)$ Neumann terms. A detailed proof is beyond the scope of this thesis.

2.2 Numerical Analysis and Optimization

In this section, we formulate the computational algorithms used in our implementation of the considered regularization strategy.

2.2.1 Finite Difference Method

The finite difference method is used to compute solutions to a discretized version of a PDE by converting the problem to a system of algebraic equations. The method is simpler than the well-established finite element method, but requires the domain of the PDE to be a hypercube. In this section, we present the main ideas of the method applied to the wave equation, following Langtangen and Linge [7, ch. 2].

Let $N_x, N_t \in \mathbb{N}$. Let $L, T \in \mathbb{R}_+$, and define uniform meshes

$$0 = x_0 < x_1 < \cdots < x_{N_x-1} < x_{N_x} = L$$

and

$$0 = t_0 < t_1 < \cdots < t_{N_t-1} < t_{N_t} = 2T$$

with mesh spacing Δx and Δt, respectively. To stay consistent with the notation of Langtangen and Linge, we denote the index of the x-mesh by i and the t-meshes by n, despite our differing order of parameters. For a function $u_i^n = u(t_n, x_i)$ defined on the two-dimensional mesh, define discrete second derivative operators $D_t D_t$ and $D_x D_x$ by

$$[D_t D_t u]_i^n = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}, \qquad [D_x D_x u]_i^n = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}.$$

Note that these operators can be seen as repeated applications of forward difference operators: If we define $[D_t u]_i^n := \frac{u_i^{n+1} - u_i^n}{\Delta t}$, then applying D_t to the two terms separately gives $\frac{u_i^{n+2} - 2u_i^{n+1} + u_i^n}{\Delta t^2}$, which is equal to our definition of $[D_t D_t u]_i^{n+1}$. For all $0 < i < N_x$ and $0 < n < N_t$, the finite difference method requires the discretized wave equation

$$[D_t D_t u]_i^n = c(x_i)^2 [D_x D_x u]_i^n \tag{2.2}$$

to be satisfied at the interior points of the mesh ($0 < i < N_x$ and $0 < n < N_t$). Writing out the definition of the discretized operators yields the following update rule in the interior:

$$u_i^{n+1} = -u_i^{n-1} + 2u_i^n + c(x_i)^2 \frac{\Delta t^2}{\Delta x^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n). \tag{2.3}$$

The case $i = 0$ must be treated separately. Define the discrete centered derivative $[D_{2x} u]_i^n = \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$. Then a natural requirement for the Neumann boundary value f is $[D_{2x} u]_0^n = f(t_n)$. Note that this is only a formal requirement, as the derivative would require the computation of u_{-1}^n, which is not yet defined. Using the equation as a definition, we have

$$u_{-1}^n := u_1^n - 2f(x_n)\Delta x.$$

Inserting this definition into the update rule (2.3) gives the boundary update rule

$$u_0^{n+1} = -u_0^{n-1} + 2u_0^n + 2c(x_0)^2 \frac{\Delta t^2}{\Delta x^2}(u_1^n - u_0^n - f(x_n)\Delta x).$$

Finally, the values u_i^0 are given by the Cauchy data.

2.2.2 Gradient Descent

Gradient descent is an optimization method for finding minima of differentiable functions. For a function $f \in C^1(\mathbb{R}^n)$, the method starts with an initial guess $x_0 \in \mathbb{R}^n$ and iterates the formula

$$x_{i+1} \coloneqq x_i - \alpha_i \nabla f(x_i),$$

where α_i are step sizes, typically chosen via line search methods. An example α_i is the minimizer

$$\alpha_i \coloneqq \arg\min_{t>0} f(x_i + t\nabla f(x_i)).$$

The choice of minimizing step sizes is known as *exact line search* in optimization.

However, in practice, exact line search can be computationally demanding, as the method needs to find a global minimizer of f on the line

$$\{x_i + t\nabla f(x_i) \mid t \in \mathbb{R}\}$$

on step number $i+1$. Thus *inexact line search* methods using a different choice of step size are of practical interest [9, p. 31]. A particular example of inexact line search is the (long) Barzilai-Borwein method, where the step size is chosen by the formula

$$\alpha_i \coloneqq \frac{\Delta x \cdot \Delta x}{\Delta x \cdot \Delta g},$$

where $\Delta x = x_i - x_{i-1}$ and $\Delta g = \nabla f(x_i) - \nabla f(x_{i-1})$ [1]. The Barzilai-Borwein method is less affected by ill-conditioning than gradient descent with exact line search [8, p. 91]. Note that the first step size α_0 must be specified manually: We simply set α_0 to a small constant.

2.2.3 Conjugate Gradient Method

The conjugate gradient method is an iterative method for finding approximate solutions to systems of linear equations with a symmetric positive definite matrix. Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $b \in \mathbb{R}^n$ be a given vector, and consider the equation $Ax = b$ to be solved for $x \in \mathbb{R}^n$.

Algorithm 1: Gradient Descent with Barzilai-Borwein step size

Data: function $f \in C^1(\mathbb{R}^n)$, vector $x_0 \in \mathbb{R}^n$
Data: integer $N > 0$, float $\epsilon > 0$, float $\alpha_0 > 0$
Output: vector $x \in \mathbb{R}^n$ that approximates a local minimizer of f

$k \leftarrow 0$;
$x_{\text{old}} \leftarrow x_0$;
$g_{\text{old}} \leftarrow \bar{0}$;
$g \leftarrow \nabla f(x_0)$;
$x \leftarrow x_{\text{old}} - \alpha_0 g$ `// initial step`;
$g \leftarrow \nabla f(x)$;
$k \leftarrow k + 1$;
while $k < N$ **do**
  $\Delta x \leftarrow x - x_{\text{old}}$;
  $\Delta g \leftarrow g - g_{\text{old}}$;
  $\alpha \leftarrow \frac{\Delta x \cdot \Delta x}{\Delta x \cdot \Delta g}$;
  $x_{\text{old}} \leftarrow x$;
  $g_{\text{old}} \leftarrow g$;
  $x \leftarrow x_{\text{old}} - \alpha g_{\text{old}}$;
  if $\|x - x_{old}\|_2 < \epsilon$ **then**
    break `// early stop`;
  end
  $g \leftarrow \nabla f(x)$;
  $k \leftarrow k + 1$;
end
return x

Algorithm 2: The Conjugate Gradient Method

Data: symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$, vectors $x_0, b \in \mathbb{R}^n$
Data: integer $N > 0$, float $\epsilon > 0$
Output: vector $x \in \mathbb{R}^n$ for which $Ax \approx b$
$r_{\text{old}} \leftarrow b - Ax_0$;
$p \leftarrow r_{\text{old}}$;
$x \leftarrow x_0$;
$k \leftarrow 0$;
while $k < N$ **do**
    $\alpha \leftarrow \frac{r_{\text{old}}^T r_{\text{old}}}{p^T A p}$;
    $x \leftarrow x + \alpha p$;
    $r \leftarrow r_{\text{old}} - \alpha A p$;
    if $\|r\|_2 < \epsilon$ **then**
        break `// early stop`;
    end
    $\beta \leftarrow \frac{r^T r}{r_{\text{old}}^T r_{\text{old}}}$;
    $p \leftarrow r + \beta p$;
    $r_{\text{old}} \leftarrow r$;
    $k \leftarrow k + 1$;
end
return x

The conjugate gradient method then seeks the solution x iteratively as formulated in Algorithm 2. As an iterative method, the algorithm can be terminated at a desired number of iterations or when step sizes cross under a predetermined threshold.

The conjugate gradient method can also be formulated as an optimization method. However, in this thesis, we will only use it to solve systems of linear equations. A more comprehensive explanation of the conjugate gradient method is given by Shewchuk [10].

Chapter 3

Theoretical Framework

In this chapter, the forward and inverse problems are defined. In addition, we examine the theoretical solution method for the inverse problem underpinning the computational solutions of Chapter 4: In particular, the classical Blagovestchenskii identities are proven, followed by a description of the iterative time-reversal control method formulated by Korpela, Lassas and Oksanen for the 1+1-dimensional inverse problem [6].

3.1 Forward Problem

Following the problem definition of [6], we consider waves on the non–negative real axis $M = [0, +\infty)$. Let $c \in C^2(M)$. For $C_0, C_1, L, m > 0$, the function c is required to satisfy the following:

$$\begin{cases} c(x) & \in [C_0, C_1], \\ \|c\|_{C^2(M)} & \leq m, \\ c - 1 & \in C_0^2(0, L). \end{cases} \tag{3.1}$$

In particular, outside of $(0, L) \subset M$ the function c is identically one. Let $T > \frac{L}{C_0}$, ensuring that waves from a source f can propagate all over $[0, L)$ in time T. Define $\Omega \coloneqq (0, 2T) \times M$, and let $f \in L^2(0, 2T)$. We consider the following formulation of the wave equation. Note the negative Neumann boundary term, which differs from the formulation in [6]:

$$\begin{cases} (\partial_t^2 - c(x)^2 \partial_x^2) u(t, x) = 0 & \text{in } \Omega, \\ \partial_x u(t, 0) = -f(t) & t \in (0, 2T), \\ u|_{t=0} = \partial_t u|_{t=0} = 0 & x \in M. \end{cases} \tag{3.2}$$

Suppose that the boundary term f is given. By [6, Thm.4], the initial-boundary value problem (3.2) has a unique solution $u^f \in H^1(\Omega)$. We may take the trace of this solution at $x = 0$ to obtain the Dirichlet boundary value $u^f|_{x=0} \in L^2(0,2T)$. Hence we can consider the *Neumann-to-Dirichlet operator* $\Lambda\colon L^2(0,2T) \to L^2(0,2T)$, which maps f to $u^f|_{x=0}$. The operator Λ is bounded and therefore continuous [6, Thm.4].

We note that finding Λ is equivalent to solving the Neumann initial-boundary value problem (3.2). Thus, knowledge of the wave speed c is required to compute Λ. Let us denote the operator mapping c to Λ by $\mathcal{A}\colon C^2(M) \to \mathcal{L}(L^2(0,2T))$. Then, given c, our forward problem is to compute

$$\Lambda = \mathcal{A}(c). \tag{3.3}$$

We note that Λ commutes with time delay, as shown in the next theorem.

Theorem 3.1.1. *Let $s > 0$, and define the forward shift operator*

$$F\colon L^2(0,2T) \to L^2(0,2T), \quad Fg(t) = g(t-s),$$

extending g by zero on the negative real axis[1]. *Let $h \in L^2(0,2T)$, and denote $f(t) := Fh(t)$. Then the solution u^f of (3.2) satisfies*

$$u^f(t,x) = u^h(t-s,x),$$

extending u^h by zero outside Ω. In particular, $\Lambda F = F\Lambda$.

Proof. First consider the case with smooth, compactly supported boundary sources. Let $h \in C_0^\infty(0,2T)$ and $f := Fh$. By definition, the solution u^f satisfies

$$\begin{cases} (\partial_t^2 - c(x)^2\partial_x^2)u^f(t,x) = 0 & \text{in } \Omega, \\ \partial_x u^f(t,0) = -f(t) & t \in (0,2T), \\ u^f|_{t=0} = \partial_t u^f|_{t=0} = 0 & x \in M. \end{cases}$$

Then the definition of f gives

$$\begin{cases} (\partial_t^2 - c(x)^2\partial_x^2)u^f(t,x) = 0 & \text{in } \Omega, \\ \partial_x u^f(t,0) = -h(t-s) & t \in (0,2T), \\ u^f|_{t=0} = \partial_t u^f|_{t=0} = 0 & x \in M. \end{cases}$$

However, because $h(r) = 0$ when $r < 0$, we can rewrite the Cauchy data:

$$\begin{cases} (\partial_t^2 - c(x)^2\partial_x^2)u^f(t,x) = 0 & \text{in } \Omega, \\ \partial_x u^f(t,0) = -h(t-s) & t \in (s,2T), \\ u^f|_{t=s} = \partial_t u^f|_{t=s} = 0 & x \in M. \end{cases}$$

[1]Note that F is well defined, as the definition is not affected by choice of representatives in $L^2(0,2T)$.

Let us define $t' = t - s$. Now we may rewrite the problem as

$$\begin{cases} (\partial_t^2 - c(x)^2\partial_x^2)u^f(t'+s,x) = 0 & \text{in } (0,2T-s)\times M, \\ \partial_x u^f(t'+s,0) = -h(t') & t' \in (0,2T-s), \\ u^f|_{t'=0} = \partial_t u^f|_{t'=0} = 0 & x \in M. \end{cases}$$

However, this problem is also solved by $u^h(t',x)$ inside $(0,2T-s)\times M$. Then because problem 3.2 is uniquely solvable, we must have

$$u^f(t'+s,x) = u^h(t',x)$$

inside the subdomain $(t',x) \in (0,2T-s)\times M \subset \Omega$. Now by substituting back $t = t'+s$ and identifying both functions with their zero extensions, we have

$$u^f(t,x) = u^h(t-s,x) \quad \text{in } \Omega. \tag{3.4}$$

Setting $x = 0$ in equation (3.4), we get $\Lambda F h = F\Lambda h$ for $h \in C_0^\infty(0,2T)$.

Now consider the more general case with boundary source $g \in L^2(0,2T)$. Then because $C_0^\infty(0,2T)$ is dense in $L^2(0,2T)$, we can choose a sequence $(g_k)_{k=1}^\infty$ in $C_0^\infty(0,2T)$ converging to g. We remark that F is bounded, as for any $\tilde{h} \in L^2(0,2T)$, we have

$$\|F\tilde{h}\|_{L^2(0,2T)} \leq \|\tilde{h}\|_{L^2(0,2T)}$$

due to the zero extension of $\tilde{h}$. In addition, Λ is bounded by [6, Thm.5]. Now since bounded linear operators on a Hilbert space are continuous, we have

$$\begin{aligned} \Lambda F g &= \lim_{k\to\infty} \Lambda F g_k && \text{(continuity of } \Lambda, F) \\ &= \lim_{k\to\infty} F\Lambda g_k && \text{(commutativity on } C_0^\infty(0,2T)) \\ &= F\Lambda g. \end{aligned}$$

As $g \in L^2(0,2T)$ is arbitrary, we conclude that $\Lambda F = F\Lambda$. □

3.2 Inverse Problem

We now turn to the inverse problem corresponding to (3.3). That is, we know Λ and seek to reconstruct the wave speed c. In [6], Korpela, Lassas and Oksanen propose a 1+1-dimensional formulation of the iterative time-reversal control method (ITRC) originally defined in [2]. Before presenting the method, we define relevant concepts.

Definition 3.2.1. Let $x \in M$. Let the wave speed c satisfy the assumptions (3.1) of the forward problem. The *travel time distance* from 0 to x corresponding to the wave speed c is defined by $d(x, 0) := \int_0^x \frac{1}{c(t)} dt$. The corresponding *travel time volume form* is given by $dV = c^{-2} dx$.

Definition 3.2.2. Let $r > 0$. The *domain of influence* of the problem (3.2) is $M(r) := \{x \in M \mid d(x, 0) \le r\}$. The travel time volume of the domain of influence is denoted by $V(r) := \|1_{M(r)}\|_{L^2(M;dV)}$.

We have now defined two coordinate systems on M, and will need mappings between the systems.

Definition 3.2.3. Define $\tau\colon M \to [0, +\infty)$ by $\tau(x) := d(x, 0)$. Since $c > 0$, the function τ is strictly increasing and hence bijective. Denote its inverse by $\chi\colon [0, +\infty) \to M$.

The considered solution method has two steps. First, the travel time volumes $V(r)$ are approximated using a regularization result requiring knowledge of Λ. Second, the wave speed c is recovered from $V(r)$ using calculus. The second step is the more straightforward of the two, and is presented here.

In order to compute the wave speed c from travel time volumes $V(r)$, note first that

$$\chi'(t) = \frac{1}{\tau'(\chi(t))} = c(\chi(t)).$$

Define $v\colon [0, +\infty) \to [0, +\infty)$ by $v(t) = c(\chi(t))$, so $\chi'(t) = v(t)$. Then as in [6, p. 9], we can change variables in the definition of V to get

$$\begin{aligned} V(r) = \|1_{M(r)}\|^2_{L^2(M;dV)} &= \int_0^{\chi(r)} c(x)^{-2} dx \\ &= \int_0^r v(t)^{-2} \chi'(t) dt = \int_0^r \frac{1}{v(t)} dt. \end{aligned}$$

Thus we have the following identities:

$$v(t) = \frac{1}{V'(t)}, \tag{3.5}$$

$$\chi(t) = \int_0^t v(t') dt', \tag{3.6}$$

$$c(x) = v(\chi^{-1}(x)). \tag{3.7}$$

Suppose that we have found a way to calculate the volumes $V(r)$ for all $r \in [0, T]$. Then using identities (3.5), (3.6) and (3.7) in order, we can compute the wave speed c. Note that although the identities use the basic operations of calculus, a computational implementation will need special care when computing the inverse derivative in identity (3.5). This topic is explored in further detail in Section 4.2.2.

Having described the second step of the theoretical solution scheme, we follow by presenting a method to approximate $V(r)$.

3.2.1 Blagovestchenskii Identities

Computing the travel time volumes required to find c requires two important results that will directly influence our implementation: the Blagovestchenskii identities, and an ITRC–based minimization result by Korpela, Lassas and Oksanen [6]. To formulate the results, we must first define certain integral operators.

Definition 3.2.4. Let Λ be the Neumann-to-Dirichlet operator corresponding to problem (3.2). Let

$$1_{\triangle}(t,s) := \begin{cases} 1, & t+s \leq 2T \text{ and } s > t > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Let $r > 0$. The *Blagovestchenskii operators* $B, J, R, K \in \mathcal{L}(L^2(0,2T)$ are defined by

$$\begin{aligned} Bf(t) &:= 1_{(0,T)}(t) \int_t^T f(s)ds, \\ Jf(t) &:= \frac{1}{2} \int_0^{2T} 1_{\triangle}(t,s) f(s) ds, \\ Rf(t) &:= f(2T-t) \\ K &:= J\Lambda - R\Lambda R J. \end{aligned}$$

The proof of the Blagovestchenskii identities requires computing the adjoint of Λ, which is in fact characterized by the previously presented operator R. In more detail, we have the following lemma:

Lemma 3.2.5. *Denote the $L^2(0,2T)$-adjoint of Λ by Λ^*. Then $\Lambda^* = R\Lambda R$.*

Proof. Denote $\Box := \partial_t^2 - c(x)^2\partial_x^2$ for brevity. Consider first the action of Λ on $C_0^\infty(0,2T)$. Let $f, \phi \in C_0^\infty(0,2T)$. Using the operator R from Definition 3.2.4, let u and w solve the initial-boundary value problems

$$\begin{cases} \Box u(t,x) = 0 & \text{in } \Omega, \\ \partial_x u|_{x=0} = -f & t \in (0,2T), \\ u|_{t=0} = \partial_t u|_{t=0} = 0 & x \in M, \end{cases} \qquad \begin{cases} \Box w(t,x) = 0 & \text{in } \Omega, \\ \partial_x w|_{x=0} = -R\phi & t \in (0,2T), \\ w|_{t=0} = \partial_t w|_{t=0} = 0 & x \in M, \end{cases}$$

respectively. By Theorem 2.1.3, u and w are in $C^\infty(\Omega)$.

Define $v(t,x) \coloneqq w(2T-t,x)$. Then $\partial_x v(t,x) = \partial_x w(2T-t,x)$, implying that $\partial_x v|_{x=0} = R(\partial_x w|_{x=0}) = -\phi$. Thus $v \in C^\infty(\Omega)$ solves the initial-boundary value problem

$$\begin{cases} \Box v(t,x) = 0 & \text{in } \Omega, \\ \partial_x v|_{x=0} = -\phi & t \in (0,2T), \\ v|_{t=2T} = \partial_t v|_{t=2T} = 0 & x \in M, \end{cases}$$

and is the unique solution by uniqueness of w.

Consider the product measure $d\eta = dtdV$ on Ω formed by scaling the x-component of the Lebesgue product measure by $c^{-2}(x)$. Then because $\Box u = \Box v = 0$, we have

$$\begin{aligned} 0 =& \langle \Box u, v\rangle_{L^2(\Omega;d\eta)} - \langle u, \Box v\rangle_{L^2(\Omega;d\eta)} \\ =& \langle \partial_t^2 u, v\rangle_{L^2(\Omega;d\eta)} - \langle c^2\partial_x^2 u, v\rangle_{L^2(\Omega;d\eta)} - \langle u, \partial_t^2 v\rangle_{L^2(\Omega;d\eta)} + \langle u, c^2\partial_x^2 v\rangle_{L^2(\Omega;d\eta)} \\ =& \langle \partial_t^2 u, v\rangle_{L^2(\Omega;d\eta)} - \langle \partial_x^2 u, v\rangle_{L^2(\Omega)} - \langle u, \partial_t^2 v\rangle_{L^2(\Omega;d\eta)} + \langle u, \partial_x^2 v\rangle_{L^2(\Omega)}. \end{aligned}$$

Integrating the first term by parts, we get

$$\begin{aligned} \langle \partial_t^2 u, v\rangle_{L^2(\Omega;d\eta)} &= \int_M \int_0^{2T} \partial_t^2 u(t,x) v(t,x) dtdV \\ &= \int_M \left([\partial_t u(t,x)v(t,x)]_{t=0}^{2T} - \int_0^{2T} \partial_t u(t,x)\partial_t v(t,x)dt \right) dV \\ &= \int_M \left(0 - \int_0^{2T} \partial_t u(t,x)\partial_t v(t,x)dt \right) dV \\ &= -\langle \partial_t u, \partial_t v\rangle_{L^2(\Omega;d\eta)} \end{aligned}$$

Next, we consider the term $\langle \partial_x^2 u, v\rangle_{L^2(\Omega)}$. Recall that $c(x) \in [C_0, C_1]$ by the standing assumptions of the forward problem. Note that the product function $\partial_x^2 u(t,x)v(t,x)$ is in $C^\infty(\Omega)$, and is thus measurable. Using Tonelli's theorem and the fact that $u = v = 0$ when $x > C_1 2T$ due to the finite speed of wave propagation, we have

$$\begin{aligned} \langle \partial_x^2 u, v\rangle_{L^2(\Omega)} &= \int_M \int_0^{2T} \partial_x^2 u(t,x)v(t,x)dtdx \\ &= \int_0^{2T} \int_M \partial_x^2 u(t,x)v(t,x)dxdt \qquad \text{(Tonelli)} \\ &= \int_0^{2T} \lim_{r\to\infty} \int_0^r \partial_x^2 u(t,x)v(t,x)dxdt \\ &= \int_0^{2T} \lim_{r\to\infty} \left([\partial_x u(t,x)v(t,x)]_0^r - \int_0^r \partial_x u(t,x)\partial_x v(t,x)dx \right) dt \\ &= \int_0^{2T} \left(-\partial_x u(t,0)v(t,0) - \int_M \partial_x u(t,x)\partial_x v(t,x)dx \right) dt \\ &= \langle -\partial_x u|_{x=0}, v|_{x=0}\rangle_{L^2(0,2T)} - \langle \partial_x u, \partial_x v\rangle_{L^2(\Omega)}. \end{aligned}$$

Similarly,

$$\begin{aligned}\langle u, \partial_t^2 v\rangle_{L^2(\Omega;d\eta)} &= -\langle \partial_t u, \partial_t v\rangle_{L^2(\Omega;d\eta)},\\ \langle u, \partial_x^2 v\rangle_{L^2(\Omega)} &= \langle u|_{x=0}, -\partial_x v|_{x=0}\rangle_{L^2(0,2T)} - \langle \partial_x u, \partial_x v\rangle_{L^2(\Omega)}.\end{aligned}$$

Thus

$$\begin{aligned}0 =&\langle \Box u, v\rangle_{L^2(\Omega;d\eta)} - \langle u, \Box v\rangle_{L^2(\Omega;d\eta)}\\ =&\langle \partial_t^2 u, v\rangle_{L^2(\Omega;d\eta)} - \langle \partial_x^2 u, v\rangle_{L^2(\Omega)}\\ &- \langle u, \partial_t^2 v\rangle_{L^2(\Omega;d\eta)} + \langle u, \partial_x^2 v\rangle_{L^2(\Omega)}\\ =&\langle u|_{x=0}, -\partial_x v|_{x=0}\rangle_{L^2(0,2T)} - \langle -\partial_x u|_{x=0}, v|_{x=0}\rangle_{L^2(0,2T)}\\ =&\langle \Lambda f, \phi\rangle_{L^2(0,2T)} - \langle f, v|_{x=0}\rangle_{L^2(0,2T)}.\end{aligned}$$

This implies that

$$\langle \Lambda f, \phi\rangle_{L^2(0,2T)} = \langle f, v|_{x=0}\rangle_{L^2(0,2T)}$$

for all $f, \phi \in C_0^\infty(0, 2T)$. Next, we observe that

$$v|_{x=0} = R(w|_{x=0}) = R\Lambda(-\partial_x w|_{x=0}) = R\Lambda R\phi.$$

Combining these results, we have

$$\langle \Lambda f, \phi\rangle_{L^2(0,2T)} = \langle f, R\Lambda R\phi\rangle_{L^2(0,2T)} \tag{3.8}$$

for any $f, \phi \in C_0^\infty(0, 2T)$.

Consider now the case of $g, h \in L^2(0, 2T)$. Then because $C_0^\infty(0, 2T)$ is dense in $L^2(0, 2T)$, we can choose sequences of functions $(g_i)_{i=1}^\infty$ and $(h_j)_{j=1}^\infty$ in $C_0^\infty(0, 2T)$ converging to g and h, respectively. Now as the $L^2(0, 2T)$ -inner product is continuous, and because Λ and R are bounded and hence continuous, we may take limits in equation (3.8) to obtain

$$\begin{aligned}\langle \Lambda g, h\rangle_{L^2(0,2T)} &= \langle \Lambda[\lim_{i\to\infty} g_i], \lim_{j\to\infty} h_j\rangle_{L^2(0,2T)}\\ &= \lim_{i\to\infty}\lim_{j\to\infty}\langle \Lambda g_i, h_j\rangle_{L^2(0,2T)} && \text{(continuity)}\\ &= \lim_{i\to\infty}\lim_{j\to\infty}\langle g_i, R\Lambda R h_j\rangle_{L^2(0,2T)} && \text{(equation (3.8))}\\ &= \langle g, R\Lambda R h\rangle_{L^2(0,2T)}.\end{aligned}$$

For a given $h \in L^2(0, 2T)$, this equality holds with any $g \in L^2(0, 2T)$. Now by the definition of the adjoint, $\Lambda^* h = R\Lambda R h$. Thus $\Lambda^* = R\Lambda R$ on $L^2(0, 2T)$. $\square$

Using the Blagovestchenskii operators, we can formulate the *Blagovestchenskii identities*, which relate boundary terms to travel time volumes. We present a proof based on [2, pp. 71-72].

Theorem 3.2.6 (Blagovestchenskii identities)**.** *Let $f, h \in L^2(0, 2T)$. Denote the solution $u(t, x)$ of* (3.2) *corresponding to a boundary source f at a time $t = T$ by $u^f(T)$. Then we have*

$$\langle u^f(T), 1\rangle_{L^2(M;dV)} = \langle f, B1\rangle_{L^2(0,2T)}, \tag{3.9}$$

$$\langle u^f(T), u^h(T)\rangle_{L^2(M;dV)} = \langle f, Kh\rangle_{L^2(0,2T)}. \tag{3.10}$$

Proof. First consider $f, h \in C_0^\infty(0, 2T)$. By Theorem 2.1.3, the solutions u^f and u^h are in $C^\infty(\Omega)$. By the finite speed of wave propagation, it follows that $u^f(t)$ and $u^h(s)$ are in $L^2(M)$ and $L^2(M; dV)$ for all $t, s \in [0, 2T]$.

We start by proving the second identity, equation (3.10). Let

$$w(t, s) := \int_M u^f(t)u^h(s)dV.$$

Note that the integrand is bounded by smoothness and the finite speed of wave propagation. Thus by the Leibniz integral rule, we have

$$\begin{aligned}
(\partial_t^2 - \partial_s^2)w(t, s) &= (\partial_t^2 - \partial_s^2)\int_M u^f(t)u^h(s)dV \\
&= \int_M \left(\partial_t^2 u^f(t)u^h(s) - u^f(t)\partial_s^2 u^h(s)\right) dV \\
&= \int_M c^2(x)\left(\partial_x^2 u^f(t)u^h(s) - u^f(t)\partial_x^2 u^h(s)\right) dV \\
&= \int_M \left(\partial_x^2 u^f(t)u^h(s) - u^f(t)\partial_x^2 u^h(s)\right) dx \\
&= \langle \partial_x^2 u^f(t), u^h(s)\rangle_{L^2(M)} - \langle u^f(t), \partial_x^2 u^h(s)\rangle_{L^2(M)}.
\end{aligned}$$

Integrating by parts as in the proof of Lemma 3.2.5, this simplifies to

$$\begin{aligned}
(\partial_t^2 - \partial_s^2)w(t, s) =& - \partial_x u^f(t)|_{x=0}u^h(s)|_{x=0} \\
&+ u^f(t)|_{x=0}\partial_x u^h(s)|_{x=0} \\
=& f(t)\Lambda h(s) - \Lambda f(t)h(s).
\end{aligned}$$

Combining this result and the Cauchy data satisfied by u^f and u^h means that w solves the following inhomogeneous wave equation initial-boundary value problem:

$$\begin{cases}
(\partial_t^2 - \partial_s^2)w(t, s) = f(t)\Lambda h(s) - \Lambda f(t)h(s) & \text{in } (0, 2T)^2, \\
w|_{s=0} = \partial_s w|_{s=0} = 0 & t \in (0, 2T), \\
w|_{t=0} = \partial_t w|_{t=0} = 0 & s \in [0, 2T).
\end{cases} \tag{3.11}$$

We will solve problem (3.11) explicitly, following Evans [4, pp. 80-81]. By Duhamel's principle, the problem can be solved by first considering the solution $\tilde{w}(t,s;r)$ of the problem

$$\begin{cases} (\partial_t^2 - \partial_s^2)\tilde{w}(t,s;r) = 0 & \text{in } (r,2T)\times(0,2T), \\ \tilde{w}|_{s=0} = \partial_s \tilde{w}|_{s=0} = 0 & t \in (0,2T), \\ \tilde{w}|_{t=r} = 0 & s \in [0,2T), \\ \partial_t \tilde{w}(r,s;r) = f(r)\Lambda h(s) - \Lambda f(r) h(s) & s \in [0,2T). \end{cases}$$

To calculate $\tilde{w}(t,s;r)$, we use d'Alembert's formula (with odd reflection due to the bound $s \in [0,\infty)$, see [4, p. 69]):

$$\tilde{w}(t,s;r) = \begin{cases} \frac{1}{2}\int_{s-(t-r)}^{s+t-r} (f(r)\Lambda h(y) - \Lambda f(r)h(y))dy, & s \geq t-r \geq 0 \\ \frac{1}{2}\int_{-s+t-r}^{s+t-r} (f(r)\Lambda h(y) - \Lambda f(r)h(y))dy, & 0 \leq s \leq t-r \end{cases} \tag{3.12}$$

Duhamel's principle then states that $w(t,s) \coloneqq \int_0^t \tilde{w}(t,s;r)dr$ is a formal solution of problem (3.11). To check that it truly is a solution, the values of w and $\partial_s w$ at $s=0$ must be computed. Let us write

$$g(r,y) \coloneqq f(r)\Lambda h(y) - \Lambda f(r) h(y).$$

At $s=0$, formula (3.12) gives

$$\tilde{w}(t,s;r) = \frac{1}{2}\int_{-s+t-r}^{s+t-r} g(r,y)dy.$$

Hence

$$w(t,0) = \frac{1}{2}\int_0^t \int_{t-r}^{t-r} g(r,y)dydr = 0,$$

and by the Leibniz integral rule,

$$\begin{aligned} \partial_s w|_{s=0} &= \frac{1}{2}\left[\int_0^t \partial_s \int_{-s+t-r}^{s+t-r} g(r,y)dydr\right]_{s=0} \\ &= \frac{1}{2}\left[\int_0^t \partial_s \left(\int_0^{s+t-r} g(r,y)dy - \int_0^{-s+t-r} g(r,y)dy\right) dr\right]_{s=0} \\ &= \frac{1}{2}\left[\int_0^t \left(g(r,s+t-r) - g(r,-s+t-r)\right) dr\right]_{s=0} \\ &= 0. \end{aligned}$$

Thus w satisfies the boundary conditions imposed on the Cauchy problem, and is a solution of problem (3.11).

Recall the function $1_\triangle$ from Definition 3.2.4. Evaluating w at $t=s=T$, we have

$$
\begin{aligned}
w(T,T) &= \frac{1}{2}\int_0^T \int_r^{2T-r} \left(f(r)\Lambda h(y) - \Lambda f(r) h(y)\right) dy dr \\
&= \frac{1}{2}\int_0^T \int_0^{2T} 1_{\triangle}(r,y) \left(f(r)\Lambda h(y) - \Lambda f(r) h(y)\right) dy dr \\
&= \int_0^T \left(f(r)J\Lambda h(r) - \Lambda f(r) Jh(r)\right) dr \\
&= \int_0^{2T} \left(f(r)J\Lambda h(r) - \Lambda f(r) Jh(r)\right) dr,
\end{aligned}
$$

where the last equality follows from the fact that $Jh(r) = J\Lambda h(r) = 0$ when $r > T$ by the definition of J. Now by the definition of the adjoint of a linear operator, we have

$$
\begin{aligned}
w(T,T) &= \int_0^{2T} \left(f(r)J\Lambda h(r) - \Lambda f(r) Jh(r)\right) dr \\
&= \langle f, J\Lambda h\rangle_{L^2(0,2T)} - \langle \Lambda f, Jh\rangle_{L^2(0,2T)} \\
&= \langle f, J\Lambda h\rangle_{L^2(0,2T)} - \langle f, \Lambda^* Jh\rangle_{L^2(0,2T)} \\
&= \langle f, J\Lambda h\rangle_{L^2(0,2T)} - \langle f, R\Lambda RJh\rangle_{L^2(0,2T)} && \text{(Lemma 3.2.5)} \\
&= \langle f, (J\Lambda - R\Lambda RJ)h\rangle_{L^2(0,2T)} \\
&= \langle f, Kh\rangle_{L^2(0,2T)}. && \text{(Definition of } K\text{)}
\end{aligned}
$$

Thus $\langle u^f(T), u^h(T)\rangle_{L^2(M;dV)} = \langle f, Kh\rangle_{L^2(0,2T)}$.

Next, we prove the first identity, equation (3.9). This time, we define $\varphi(t,s) := \int_M u^f(t) dV$. Now as φ is constant with regards to s, we get

$$
\begin{aligned}
(\partial_t^2 - \partial_s^2)\varphi(t,s) &= \int_M \partial_t^2 u^f(t) dV = \int_M c^2(x)\partial_x^2 u^f(t) dV \\
&= \int_M \partial_x^2 u^f(t) dx = \langle \partial_x^2 u^f, 1\rangle_{L^2(M)}.
\end{aligned}
$$

An integration by parts gives

$$
(\partial_t^2 - \partial_s^2)\varphi(t,s) = -\partial_x u^f(t)|_{x=0} = f(t).
$$

Due to the Cauchy data satisfied by u^f and the fact that φ is constant with respect to s, the function φ must solve the following inhomogeneous wave equation Neumann initial-boundary value problem:

$$
\begin{cases}
(\partial_t^2 - \partial_s^2)\varphi(t,s) = f(t) & \text{in } (0,2T)^2, \\
\partial_s \varphi|_{s=0} = 0 & t \in (0,2T) \\
\varphi|_{t=0} = \partial_t \varphi|_{t=0} = 0 & s \in [0,2T).
\end{cases}
$$

As previously, Duhamel's principle and d'Alembert's formula give the formal solution

$$\begin{aligned}
\tilde{\varphi}(t,s;r) &= \begin{cases} \frac{1}{2}\int_{s-(t-r)}^{s+t-r} f(r)dy, & s \geq t-r \geq 0 \\ \frac{1}{2}\int_{-s+t-r}^{s+t-r} f(r)dy, & 0 \leq s \leq t-r \end{cases} \\
\varphi(t,s) &= \int_0^t \tilde{\varphi}(t,s;r)dr.
\end{aligned}$$

The resulting φ satisfies the boundary condition $\partial_s\varphi|_{s=0} = 0$, which may be shown in a similar manner as for w. Hence

$$\begin{aligned}
\varphi(T,T) &= \frac{1}{2}\int_0^T \int_r^{2T-r} f(r)dydr = \frac{1}{2}\int_0^T f(r)\int_r^{2T-r} dydr \\
&= \frac{1}{2}\int_0^T f(r)\cdot 2\int_r^T 1(y)dydr = \int_0^T f(r)B1(r)dr \\
&= \int_0^{2T} f(r)B1(r)dr = \langle f, B1\rangle_{L^2(0,2T)},
\end{aligned}$$

and so $\langle u^f(T), 1\rangle_{L^2(M;dV)} = \langle f, B1\rangle_{L^2(0,2T)}$.

We have now proven the Blagovestchenskii identities for smooth boundary terms $f, h \in C_0^\infty(0,2T)$. Now because the identities are continuous in f and h and because $C_0^\infty(0,2T)$ is dense in $L^2(0,2T)$, we may take limits as in the proof of Theorem 3.2.5 to get the identities for $f, h \in L^2(0,2T)$. $\square$

The Blagovestchenskii identities directly imply that K is a symmetric, positive semidefinite operator. These properties will be used to justify our choice of computational scheme in Chapter 4, and so we record the result as a corollary.

Corollary 1. The operator K is symmetric and positive semidefinite.

Proof. By the Blagovestchenskii identity (3.10), we have for all $f, h \in L^2(0,2T)$ that

$$\begin{aligned}
\langle f, Kh\rangle_{L^2(0,2T)} &= \langle u^f(T), u^h(T)\rangle_{L^2(M;dV)} \\
&= \langle u^h(T), u^f(T)\rangle_{L^2(M;dV)} \\
&= \langle h, Kf\rangle_{L^2(0,2T)}.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\langle f, Kf\rangle_{L^2(0,2T)} &= \langle u^f(T), u^f(T)\rangle_{L^2(M;dV)} \\
&\geq 0.
\end{aligned}$$

$\square$

3.2.2 Approximating the Travel Time Volumes

We will now construct a boundary source f that allows us to approximate a given volume $V(r)$. First we define a helpful subspace of boundary sources.

Definition 3.2.7. Let $r \in [0, T]$. Define the set S_r and the projection $P_r \colon L^2(0, 2T) \to S_r$ by

$$\begin{aligned} S_r &\coloneqq \{f \in L^2(0, 2T) \mid \operatorname{supp}(f) \subset [T - r, T]\}, \\ P_r f &\coloneqq \chi_{[T-r,T]} f. \end{aligned}$$

Due to the finite speed of wave propagation, for $f \in S_r$, the solution $u^f(T)$ is supported in $M(r)$ in travel time coordinates.

We are now ready to state the main regularization result used in the thesis. For the proofs, we refer the reader to [6, pp. 6-8].

Theorem 3.2.8. *Let $r \in [0, T]$. Then the regularized minimization problem*

$$\min_{f \in S_r} (\langle f, Kf \rangle_{L^2(0,2T)} - 2\langle f, B1 \rangle_{L^2(0,2T)} + \alpha \|f\|^2_{L^2(0,2T)})$$

has the unique minimizer

$$f_{\alpha,r} \coloneqq (P_r K P_r + \alpha)^{-1} P_r B1$$

and the map $r \mapsto f_{\alpha,r}$ is continuous from $[0, T]$ to $L^2(0, 2T)$.

Theorem 3.2.9. *Let $f_{\alpha,r}$ be as in Theorem 3.2.8. Then $u^{f_{\alpha,r}}(T)$ converges to the indicator function of the domain of influence:*

$$\lim_{\alpha \to 0} \|u^{f_{\alpha,r}} - 1_{M(r)}\|_{L^2(M;dV)} = 0.$$

Remarkably, the Blagovestchenskii identities thus allow us to approximate $V(r) = \|1_{M(r)}\|_{L^2(M;dV)}$ by choosing a small enough α. Now by computing approximations of $V(r)$ for different values of r, we may approximate c using the scheme presented in Section 3.2. We will present the computational implementation of the scheme in Chapter 4.

Chapter 4

Computational Implementation

In this chapter, we take a closer look at the numerical formulation of our inverse problem. We first explain a method to simulate the forward problem, and then describe the computational steps that implement the regularization scheme from Section 3.2. As our computational environment, we use Python[1] together with Jupyter notebooks[2] and the Scipy ecosystem[3]. Our code is available at `https://github.com/danielrepodev/wave-1d-ITRC-TV`.

4.1 Simulation of the Forward Problem

To simulate a Neumann-to-Dirichlet operator Λ, we need to solve the forward problem (3.3) for a suitable family of Neumann boundary terms f. For such f, we can compute Λf by evaluating the solution $u^f(t)$ at the origin for all $t \in [0, 2T]$.

As in Chapter 2, we define meshes on the x and t axes. We take Δt and $c_{\max} := \max_{x\in[0,L]} c(x)$ as arguments, and compute N_t as $N_t = 2 \cdot \mathrm{ceil}\left(\frac{T}{2\Delta t}\right)$ to ensure T is contained in the time mesh. The value of Δt is updated to match the new mesh spacing. We then compute N_x as the closest integer to $\frac{L}{c_{\max}\Delta t}$ to avoid numerical dispersion, where traveling waves split into different frequency components due to mismatched mesh sizes: The reader is referred to Langtangen and Linge [7] for a more thorough analysis of the phenomenon. The resulting meshes satisfy $x_0 = t_0 = 0$, $x_{N_x} = L$ and $t_{N_t} = 2T$.

Next, we define $\mathbf{h}$ to be the discrete indicator function on the second

[1] www.python.org

[2] jupyter.org

[3] scipy.org

point of our time mesh, the point Δt. In other words,

$$\mathbf{h}_n = \begin{cases} 1, & n = 1 \\ 0, & n = 0 \text{ or } 2 \le n \le N_t. \end{cases}$$

To enable mesh refinement, $\mathbf{h}$ is interpolated between t-mesh points. This is done by treating $\mathbf{h}$ as a bump function as follows: Let $\epsilon = 10^{-16}$. For a fixed t-mesh index n, let $p(t)$ be the unique ascending cubic spline interpolant on the continuous interval $[0, \Delta t - \epsilon\Delta t]$ with $p(0) = 0$, $p(1 - \epsilon\Delta t) = 1$ and first-order derivatives set to zero at the endpoints. Let $q(t)$ similarly be the unique descending cubic spline interpolant on the continuous interval $[\Delta t + \epsilon\Delta t, 2\Delta t]$ with $q(\Delta t + \epsilon\Delta t) = 1$, $q(2\Delta t) = 0$ and first-order derivatives set to zero at the endpoints. We then interpolate $\mathbf{h}$ to a bump function h on denser time meshes by setting

$$h(t) = \begin{cases} 1, & |t - \Delta t| < \epsilon\Delta t \\ p(t), & t \in [0, \Delta t - \epsilon\Delta t] \\ q(t), & t \in [\Delta t + \epsilon\Delta t, 2\Delta t] \\ 0, & \text{otherwise.} \end{cases}$$

Observe that $\mathbf{h}$ and h agree on mesh points. Having $h(t)$ be equal to 1 in a small region around Δt helps us avoid floating-point comparison errors in the computational implementation.

After defining the bump function, the time mesh is refined to mitigate the use of inverse crime. The x-mesh is computed after the t-mesh refinement to avoid numerical dispersion. After refinement, we compute values of the solution u_i^n (using notation of Section 2.2.1) for all mesh points on the refined meshes using the finite difference method, and save the Dirichlet boundary values. Finally, we subsample the Dirichlet boundary values to obtain N_t equally spaced boundary values, completing the computation of the discrete Neumann-to-Dirichlet operator on the boundary term $\mathbf{h}$. The solver and bump function code we use is based on an implementation by Lauri Oksanen[4], which builds on [7, pp. 105-106].

Denote the computed Dirichlet boundary value corresponding to $\mathbf{h}$ by $\mathbf{u}_n$. Define F to be the forward shift operator on our time mesh so that

$$\begin{cases} (F\mathbf{u})_n = \mathbf{u_{n-1}}, & 1 \le n \le N_t, \\ (F\mathbf{u})_n = 0, & n = 0. \end{cases}$$

Then Theorem 3.1.1 allows us to define an approximation to Λ using F as follows. Define the discrete matrix $\mathbf{L}$ by translating $\mathbf{u}$ through time so that the column number j (zero-indexed) with $j \ge 1$ is equal to $F^{j-1}\mathbf{u}$, where

[4]Github link: https://github.com/l-oksanen/oprecnn/blob/main/wave1d.py

F^{j-1} is the repeated composition of F. The column number 0 is treated specially: it is computed using the interpolated half-bump function

$$\tilde{h}(t) = \begin{cases} 1, & 0 \leq t < \epsilon\Delta t \\ \tilde{q}(t), & t \in [\epsilon\Delta t, \Delta t] \\ 0, & \text{otherwise} \end{cases}$$

corresponding to the discrete incitator function $\tilde{h}$ with

$$\tilde{\mathbf{h}}_n = \begin{cases} 1, & n = 0 \\ 0, & 1 \leq n \leq N_t. \end{cases}$$

Here $\tilde{q}$ is the decreasing cubic interpolant on $[\epsilon\Delta t, \Delta t]$ with $\tilde{q}(\epsilon\Delta t) = 1$, $\tilde{q}(\Delta t) = 0$ and first-order derivatives set to zero at the endpoints.

The resulting matrix $\mathbf{L}$ is a discrete analogue to the operator Λ represented in a basis of bump functions on the time mesh points. Note that computing $\mathbf{L}$ only requires two runs of the finite difference method (for the two boundary terms $\mathbf{h}$ and $\tilde{\mathbf{h}}$) due to time translation.

4.2 Computational Regularization

The regularization scheme presented in Section 3.2 requires numerical differentiation, which is an unstable operation. To reduce the effect of noise on the derivative $\partial_r V(r)$, a second regularization step with total variation (TV) regularization [8, pp. 83-86] is employed. Thus two distinct regularization parameters are required to tune the algorithm: The parameter of volume calculation is referred to as α_{Vol}, while the parameter of differentiation is referred to as α_{TV}.

We begin by illustrating the computational volume calculation, then proceed to TV regularized differentiation.

4.2.1 Computing the Travel Time Volumes

Recall from Section 4.1 that our time mesh contains an odd number of N_t points, with T as the middle point. Denote by N_r the number of points up to T, so that $N_r = \text{floor}(\frac{N_t}{2})$. We compute the approximate travel time volumes $V(r)$ by choosing a global regularization parameter α_{Vol} and using Theorem 3.2.8 to compute values $f_{\alpha_{\text{Vol}},r}$ for N_r equally spaced mesh points $r \in [0, T]$ on the time domain. We discuss the effect of using a single α_{Vol} for all r in Chapter 6.

Recall that the energy-minimizing boundary term $f_{\alpha_{\text{Vol}},r}$ is the unique solution of

$$(P_r K P_r + \alpha_{\text{Vol}}) f_{\alpha_{\text{Vol}},r} = P_r B 1.$$

By Corollary 1, the operator K is symmetric positive semidefinite. As the operator P_r is also positive semidefinite and $\alpha_{\mathrm{Vol}}I$ is positive definite, the left-hand side operator must be positive definite. Thus the conjugate gradient (CG) method is suitable for solving the corresponding discretized equation.

The convergence of the CG algorithm may be improved by preconditioning the left-hand side matrix, using incomplete LU factorization or other methods [10]. However, as CG is run for all mesh values r separately, the computational cost of factoring the left-hand side operators was impractical in our tests. This, combined with the satisfactory convergence of CG, led us to abandon preconditioning altogether.

In our code, the Blagovestchenskii operators are defined by their operation on vectors using `scipy.sparse.linalg.LinearOperator` objects. The systems of equations are not large in our tests; with $T = 2.5$, $L = 2.5$ and $\Delta t = 0.01$, the left-hand side matrix has dimensions $(501, 501)$. For large values of r, the matrix is not sparse. We remark that the conjugate gradient method without matrix calculations is suitable for solving problems with far greater number of dimensions [8, pp. 78-82] [8, Ch.9]: However, generating the forward problem matrix $\mathbf{L}$ for the wave speeds of Chapter 5 proved computationally infeasible with our implementation.

The method is run with a maximum of $N = 20$ iterations, and an early stopping tolerance $\epsilon = 10^{-5}\|P_r B1\|_2$ (the Scipy default).

4.2.2 Computing the Wave Speed

To numerically solve v from equation (3.5), we must differentiate the travel time volumes $V(r)$. We formulate this differentiation as an inverse problem, and solve it using TV regularization.

Let $A \in \mathcal{L}(L^2(0,T))$ be the integral operator defined by $Av(r) = \int_0^r v(t)dt$. By the fundamental theorem of calculus, we have (in the continuous case)

$$A[V'] = V.$$

For the discrete case, we let $\mathbf{A}$ be a matrix determined by the cumulative trapezoidal rule. That is,

$$(\mathbf{A}v)_i = \sum_{j=0}^{i-1} v_j,$$

with $(\mathbf{A}v)_0 = 0$.

Let ε be a noise term. Then we may formulate numerical differentiation as an inverse problem in the following way: given $V + \varepsilon$, find v such that $\mathbf{A}v = V$. One method to solve this inverse problem is TV regularization, which penalizes the total variance (ℓ^1 norm of the derivative) of the solution.

More precisely, TV regularization seeks the minimizer v of the functional

$$\Phi(v) = \|\mathbf{A}v - V\|_2^2 + \alpha_{\mathrm{TV}}\|D_t v\|_1,$$

where $\|\cdot\|_p$ is the ℓ^p–norm and D_t is the forward finite difference operator on the time mesh. We write $\Phi(v) = \mathrm{DF}(v) + \alpha_{\mathrm{TV}}\mathrm{TV}(v)$, where $\mathrm{DF}(v)$ and $\mathrm{TV}(v)$ are the data fidelity and total variation terms, respectively. TV regularization captures sudden jumps in the derivative more accurately than the standard Tikhonov regularization method [3].

In our code, the minimum of the functional Φ is sought using gradient descent with the Barzilai–Borwein choice of step size (Algorithm 1). To make Φ differentiable, we use the absolute value approximation $|v_i| \approx |v_i|_\beta := \sqrt{v_i^2 + \beta}$ with $\beta = 10^{-8}$. This yields the gradient

$$\nabla\Phi = \nabla\mathrm{DF} + \alpha_{\mathrm{TV}}\nabla\mathrm{TV},$$

where

$$\begin{aligned}
\nabla\mathrm{DF}(v) &= 2A^T(Av - V),\\
(\nabla\mathrm{TV}(v))_i &= \partial_i \sum_{i=0}^{N_r-1} |v_{i+1} - v_i|_\beta\\
&= \partial_i\sqrt{(v_{i+1} - v_i)^2 + \beta} + \partial_i\sqrt{(v_i - v_{i-1})^2 + \beta}\\
&= \frac{-2v_{i+1} + 2v_i}{\sqrt{v_{i+1}^2 - 2v_{i+1}v_i + v_i^2 + \beta}} + \frac{2v_i - 2v_{i-1}}{\sqrt{v_i^2 - 2v_iv_{i-1} + v_{i-1}^2 + \beta}}.
\end{aligned}$$

The adjoint A^T is given by

$$A^T v_i = \sum_{j=i}^{n} v_j$$

for $v \in \mathbb{R}^n$.

The method is run with a maximum number of $N = 10^6$ iterations and a tolerance level $\epsilon = 10^{-5}$ for the early stopping criterion $\|x - x_{\mathrm{old}}\| < \epsilon$ of Algorithm 1. With the relatively large number of iterations, our implementation typically terminates early with well-behaved inputs.

The resulting values of v are then inserted into equation (3.6) using `scipy.integrate.trapezoid`, yielding values of $\chi(t)$ on the spatial axis. Recall from Section 4.1 that the discretization of x in the solver is done based on Δt and the maximum value of the particular wave speed considered, so as to avoid numerical dispersion. When computing reconstructions, we instead use a separately defined x-mesh to further mitigate inverse crime. Piecewise linear interpolation methods are used to compute the inverse function $\chi^{-1}(x)$, utilizing code written by Lauri Oksanen[5]. This allows us to compute c on the reconstruction x-mesh using equation (3.7).

[5] Github link: https://github.com/l-oksanen/oprecnn/blob/main/volume_inversion_data.py

Chapter 5

Results

In this chapter, we examine the performance of our implementation based on the methods of Chapter 4 on simulated data. First, the method is tested on noise–free data to determine inherent sources of error. Afterwards, Gaussian noise is added to the data, and we study robustness of the implementation. Throughout this chapter, we use the values $L = 2.5$, $T = 2.5$ and $\Delta t = 0.01$. When computing $\mathbf{L}$, a mesh refinement factor of 5 is used. Larger refinement factors caused computation times to become prohibitive for the test cases we consider in this chapter. We set the reconstruction x-mesh discussed in Section 4.2.2 to have 200 points.

5.1 Computing Volumes without Additional Noise

The condition numbers of the LHS operators for various values of α_{Vol} are shown in Table 5.1. Our results indicate that the condition number decreases proportionally to α_{Vol} as the latter increases. However, the condition numbers we got are an order of magnitude greater with $r \approx 2.49$ compared to $r \approx 0.62$. A similar effect is observed with the norms of $P_r K P_r + \alpha_{\text{Vol}}$ in 5.2.

α_{Vol}	Condition number ($r \approx 0.62$)	Condition number ($r \approx 2.49$)
1e-08	183153219.89	8653094271.04
1e-07	31142667.52	2904352843.85
1e-06	379969.87	10252048.68
1e-05	18533.47	308551.13
0.0001	1665.57	25990.82
0.001	164.77	2559.12
0.01	17.34	255.75

Table 5.1: Condition number of $P_r K P_r + \alpha_{\text{Vol}}$ with $c \equiv 1$ and two choices of r.

α_{Vol}	Norm ($r \approx 0.62$)	Norm ($r \approx 2.49$)
1e-08	0.16	2.56
1e-07	0.16	2.56
1e-06	0.16	2.56
1e-05	0.16	2.56
0.0001	0.16	2.56
0.001	0.17	2.56
0.01	0.28	2.58

Table 5.2: Norms of $P_r K P_r + \alpha_{\text{Vol}}$, $c \equiv 1$.

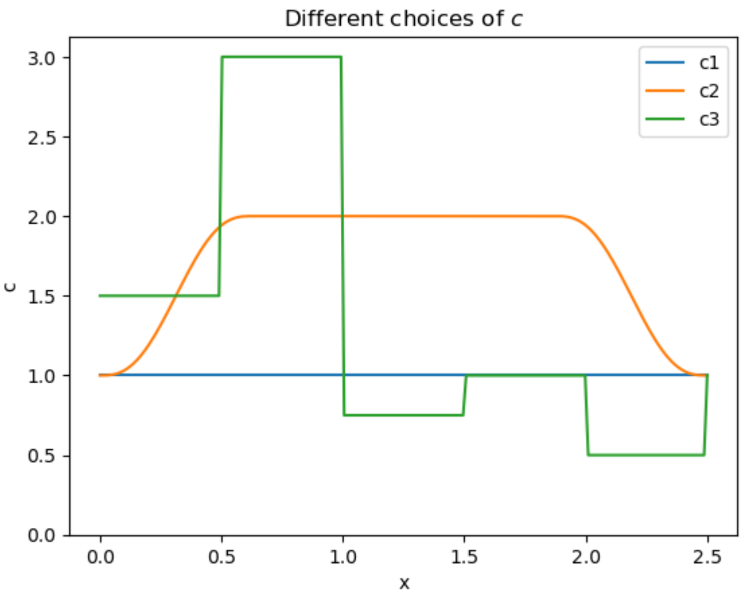

Figure 5.1: Choices of c used to test the implementation.

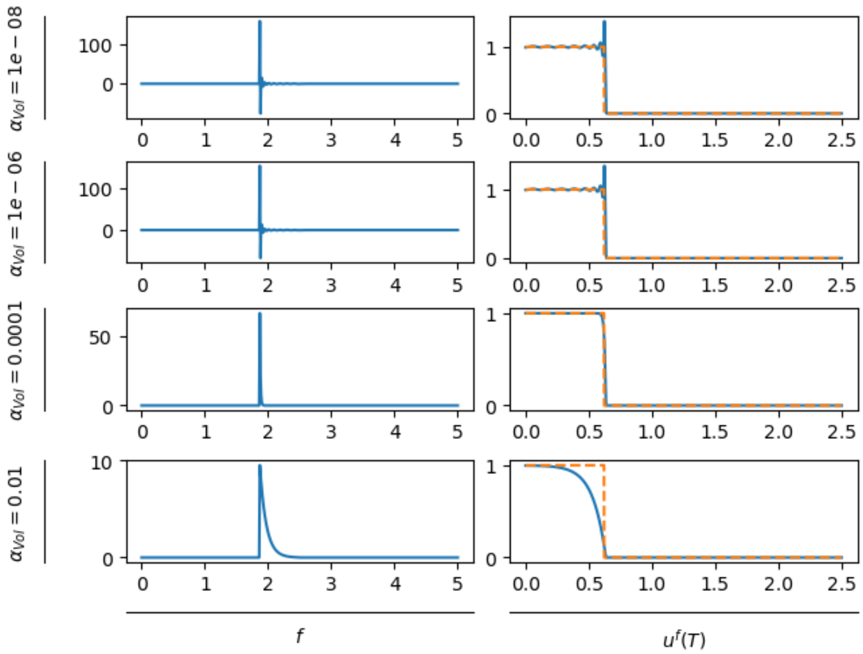

Figure 5.2: Boundary sources $f_{\alpha_{\text{Vol}},r}$ and corresponding solutions $u^{f_{\alpha_{\text{Vol}},r}}(T)$ on the t and x axes, respectively. Here $c = c_1$, $r \approx 0.62$, $T = 2.5$, $L = 2.5$ and $\Delta t = 0.01$, while α_{Vol} varies. On the right hand side, the indicator function $1_{M(r)}$ is in dashed orange.

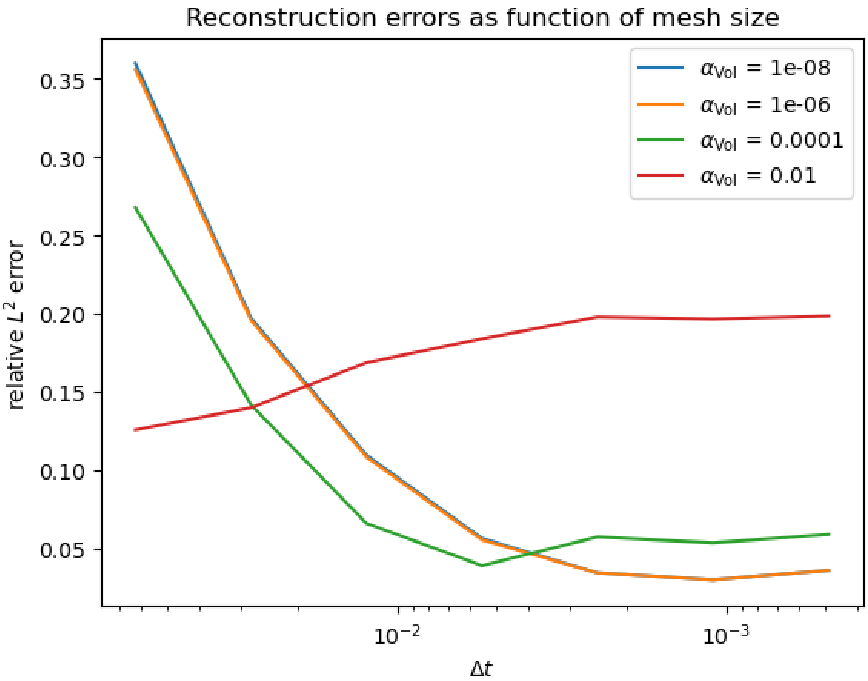

Figure 5.3: Relative L^2 errors of $u^{f_{\alpha_{\mathrm{Vol}},r}}(T)$ compared to $1_{M(r)}$ with varying mesh sizes $\Delta t = 2^{-i}$, $4 \leq i \leq 11$. Here $c = c_1$, $r \approx 0.62$, $T = 2.5$, $L = 2.5$, as in Figure 5.2. Note the decreasing, logarithmic horizontal axis. The graphs of $\alpha_{\mathrm{Vol}} = 10^{-6}$ and $\alpha_{\mathrm{Vol}} = 10^{-8}$ are almost identical.

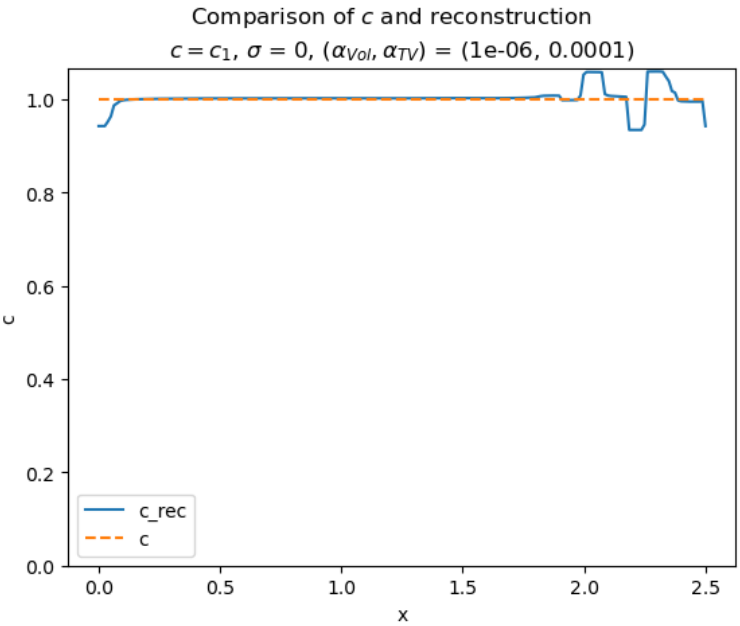

Figure 5.4: Comparison of c_1 and a reconstruction with $\Delta t = 0.01$.

To assess accuracy of the conjugate gradient results, the boundary sources $f_{\alpha_{\mathrm{Vol}},r}$ were plotted for different values of α_{Vol} and a fixed $r \approx 0.62$. Figure 5.2 displays the results. Theorem 3.2.8 states that in the continuous case with exact arithmetic, we should have $u^{f_{\alpha_{\mathrm{Vol}},r}} \to 1_{M(r)}$ in the $L^2(M; dV)$ norm as $\alpha_{\mathrm{Vol}} \to 0$. This differs from what we observe in Figure 5.2, as CG converges to noisier results with smaller α_{Vol}. This is expected, as the method converges more slowly when the left-hand side matrix has a large condition number [10, p. 37]. The noise can be seen as a form of modeling error, which diminishes as α_{Vol} increases. In Figure 5.2, the best approximation of $1_{M(r)}$ is given by the case $\alpha_{\mathrm{Vol}} = 10^{-4}$.

Relative reconstruction errors of solutions $u^{f_{\alpha_{\mathrm{Vol}},r}}$ compared to the theoretical limit of $1_{M(r)}$ are plotted in Figure 5.3. Note that the figure uses the same choices of α_{Vol} as Figure 5.2. Figure 5.3 suggests that the reconstructions with small α_{Vol} have lower bounds for their relative errors, as the error graphs increase slightly after reaching a local minimum at $\Delta t \approx 10^{-3}$. Note that the optimal value for α_{Vol} is dependent on Δt: For coarse grids (large Δt) the large parameter choice $\alpha_{\mathrm{Vol}} = 0.01$ gives better results despite the lack of added noise. In Figure 5.3, the best approximation for $1_{M(r)}$ near $\Delta t = 0.01$ is given by $\alpha_{\mathrm{Vol}} = 10^{-4}$. This agrees with the plotted solutions of Figure 5.2.

Figures 5.4, 5.5 and 5.6 show noiseless reconstructions of c_1, c_2 and c_3 respectively, with a small parameter $\alpha_{\mathrm{Vol}} = 10^{-6}$ to reduce modeling error.

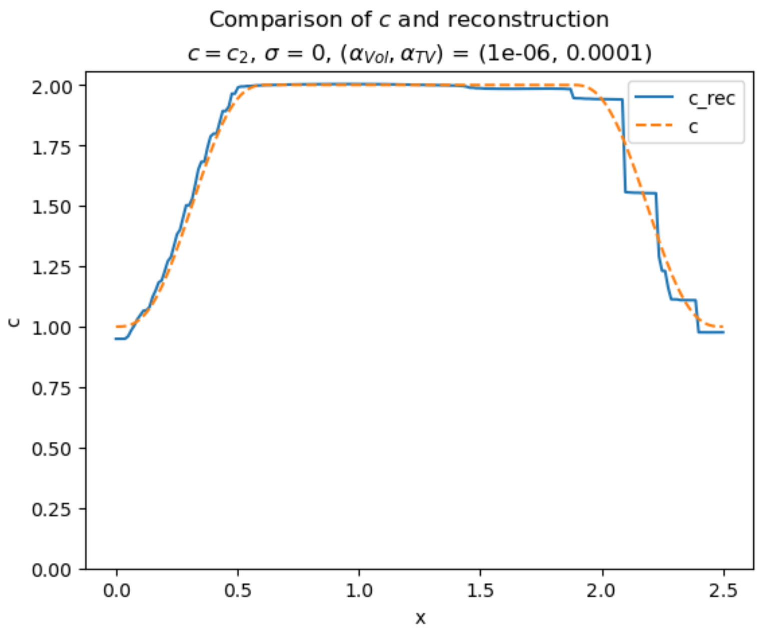

Figure 5.5: Comparison of c_2 and a reconstruction with $\Delta t = 0.01$.

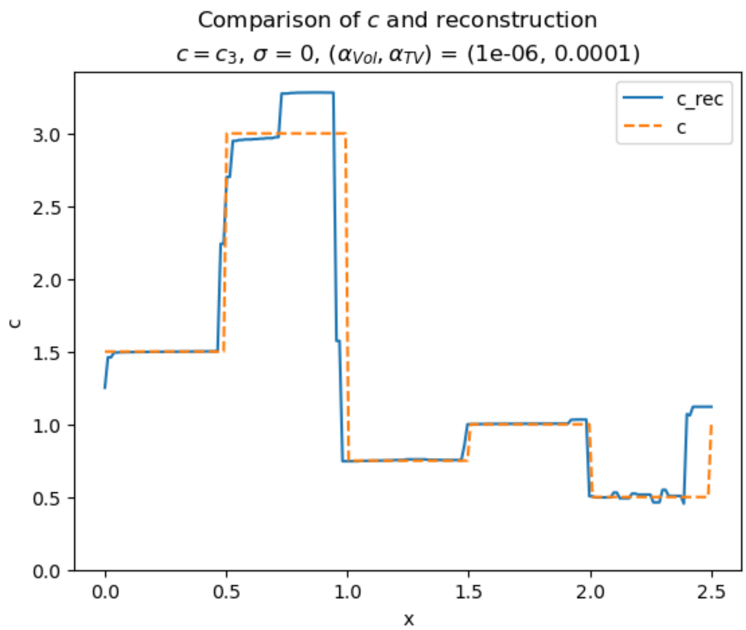

Figure 5.6: Comparison of c_3 and a reconstruction with $\Delta t = 0.01$.

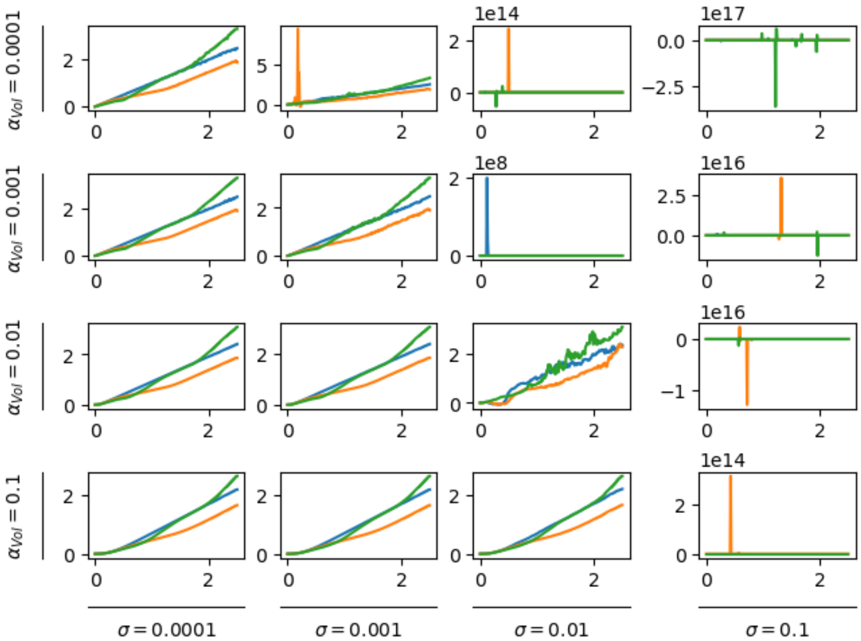

Figure 5.7: Travel time volumes computed from c_1, c_2 and c_3 with a variety of regularization parameters α_{Vol} and noise levels σ, using $\Delta t = 0.01$. Due to the large amount of computations needed, the figure uses a maximum of 10 iterations for CG as opposed to the 20 iterations used throughout the thesis. As in Figure 5.1, c_1 is in blue, c_2 in yellow and c_3 in green. Note how the volumes close to the origin are pulled towards 0 as α_{Vol} increases. The method breaks down if α_{Vol} is smaller than the noise level σ, as well as in the case $\alpha_{\mathrm{Vol}} = \sigma = 0.1$.

We remark that the reconstructions are highly sensitive to the choice of α_{TV}, particularly in the case of c_3. This means that even without noise, prior information about the form of the wave speed is needed to choice α_{TV} suitably. In addition, the reconstructions display step-like artefacts near T. These artefacts indicate that TV could not sufficiently regularize volumes near T: This may be related to the condition numbers of Table 5.1, which suggest that the computed travel time volumes are more irregular at later points of the time mesh.

5.2 Suppressing Noise

We tested our implementation with a noise term to assess its practicality. The matrix $\mathbf{L}$ was simulated using a piecewise constant c and normally distributed noise $\mathcal{E} \sim \mathcal{N}(0, \sigma^2)$ for different noise levels σ. As demonstrated in Figure 5.7, the volumes given by first regularization step diverge when

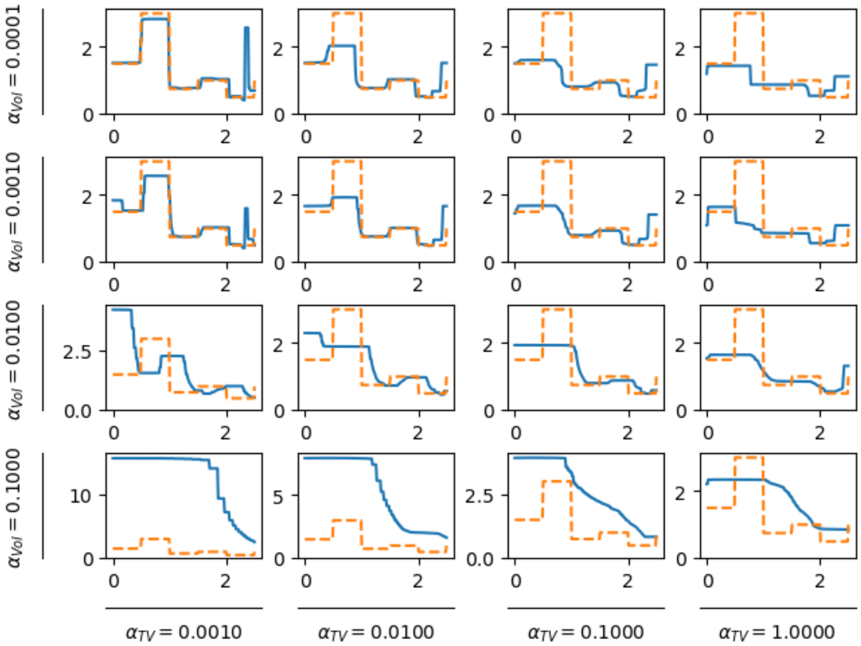

Figure 5.8: Reconstructions of c_3 with varying α, with a low noise level $\sigma = 10^{-4}$. Here $\Delta t = 0.01$. Reconstructions in solid blue, c_3 in dashed orange.

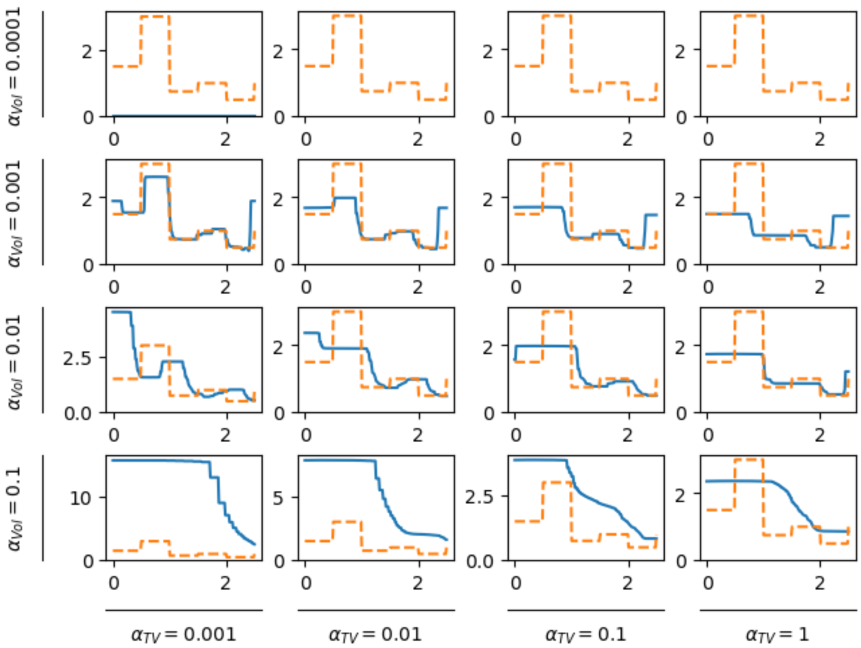

Figure 5.9: Reconstructions of c_3 with varying α, with a higher noise level $\sigma = 10^{-3}$. Here $\Delta t = 0.01$. Reconstructions in solid blue, c_3 in dashed orange. Note that the method breaks down for all α_{Vol} when $\alpha_{\text{Vol}} = 0.001 < \sigma$. In this case, the results diverge for $\alpha_{\text{Vol}} \in \{0.01, 0.1, 1\}$, while for $\alpha_{\text{Vol}} = 0.001$ the results converge to zero.

$\alpha_{\text{Vol}} < \sigma$. Hence, our results suggest using the volume parameter choice $\alpha_{\text{Vol}} = \sigma$. Taking too large α_{Vol} pulls the volumes near the origin closer to zero.

Figure 5.8 showcases the results of the two-step regularization method with noise level $\sigma = 10^{-4}$ and varying choices of α. The figure shows that a good approximation is found with $(\alpha_{\text{Vol}}, \alpha_{\text{TV}}) = (10^{-4}, 10^{-3})$.

Results with a slightly higher noise level $\sigma = 10^{-3}$ are shown in Figure 5.9. In the figure, choices of $\alpha_{\text{Vol}} = 0.001 < \sigma$ cause the method to break down. This aligns with the behavior of volumes in Figure 5.7.

Figures 5.8 and 5.9 demonstrate that increasing α_{Vol} increases the reconstructed values of c_3 near the origin, obscuring details of the true c_3. Increasing the TV parameter decreases the difference between the extremal values of c_3, which de-emphasizes the origin but does not return the obscured details. We observe that the computed values of c_3 are overestimated near L.

Taking $\alpha_{\text{Vol}} = 10^{-2}$ in Figures 5.8 and 5.9 yields reasonable approximations of c_3 past the point $L/2$ with the higher values of α_{TV}, whereas with $\alpha_{\text{Vol}} = 10^{-1}$ the reconstructions do not match the shape of c_3 anywhere. This means that with $L = 2.5$ and $\Delta t = 0.01$, only choices $\alpha_{\text{Vol}} < 10^{-1}$ were found to create sensible reconstructions. Thus using the parameter choice $\alpha_{\text{Vol}} = \sigma$ limits the noise tolerance of the method to approximately $\sigma \leq 10^{-2}$ for our test cases. This is small when compared to the norm of $\mathbf{L}$ (approximately 3.54 when $c \equiv 1$) and the norm of $P_r K P_r + \alpha_{\text{Vol}}$ with $r \approx 2.49$ (shown in Table 5.2).

Chapter 6

Discussion

The method performs reasonably within its error limits. However, the limits are small compared to the norm of $\mathbf{L}$, which may indicate a lack of robustness required for physical applications. Furthermore, prior information of the wave speed is required to choose the total variation parameter in a suitable manner.

The authors of our main theoretical source [6] studied a computational implementation of another, more theoretically sound regularization method for the inverse problem corresponding to (3.3) in [5]. The method is based on the main result of [6, pp. 10-22]. In their study, Korpela, Lassas and Oksanen demonstrate a similar noise tolerance of 0.02 for choices of regularization parameter subject to a heuristic version of Morozov's discrepancy principle [5, pp. 21-24]. The fact that our achieved noise tolerance is comparable to that of Korpela, Lassas and Oksanen suggests that the theoretically sound strategy of [5] is also computationally performant, which may motivate further studies in inverse problems related to the acoustic wave equation.

We note that when not limiting the choice of parameter using the principle, Korpela, Lassas and Oksanen achieve good reconstruction quality even with noise level 0.1. In addition, the results of Korpela, Lassas and Oksanen do not suffer from the TV artefacts or information loss near the origin present in our reconstructions: See [5, Fig.5] for reconstructions in the case of piecewise constant wave speed.

6.1 Further Improvements

One possible source of error when calculating the travel time volumes is that no information from preceding mesh points is used when calculating the next volume. Theorem 3.2.8 only guarantees convergence for each r separately, and does not guarantee that a uniform choice of α would provide a good approximation of $1_{M(r)}$ for all r. If such a result did hold, then a dynamic programming scheme might allow for more robust volume calculation. Furthermore, the addition of noise may make the left-hand side operator $P_r K P_r + \alpha_{\mathrm{Vol}}$ cease to be positive definite, which suggests that alternative methods not relying on positive definitiveness may outperform CG in suppressing noise.

Bibliography

[1] Jonathan Barzilai and Jonathan M. Borwein. "Two-Point Step Size Gradient Methods". In: *IMA Journal of Numerical Analysis* 8.1 (Jan. 1988), pp. 141–148.

[2] Kenrick Bingham et al. "Iterative Time-Reversal Control for Inverse Problems". In: *Inverse Problems and Imaging* 2.1 (2008).

[3] Rick Chartrand. "Numerical Differentiation of Noisy, Nonsmooth Data". In: *International Scholarly Research Notices* 1 (2011), p. 164564.

[4] Lawrence C. Evans. *Partial differential equations*. Second edition. Graduate studies in mathematics ; volume 19. Providence, R.I: American Mathematical Society, 2010.

[5] Jussi Korpela, Matti Lassas, and Lauri Oksanen. "Discrete regularization and convergence of the inverse problem for 1+1 dimensional wave equation". In: *Inverse Problems & Imaging* (2018).

[6] Jussi Korpela, Matti Lassas, and Lauri Oksanen. "Regularization strategy for an inverse problem for a 1 + 1 dimensional wave equation". In: *Inverse Problems* 32.6 (Apr. 2016), p. 065001.

[7] Hans Petter Langtangen and Svein Linge. *Finite Difference Computing with PDEs : A Modern Software Approach*. eng. 1st ed. 2017. Texts in Computational Science and Engineering, 16. Springer Nature, 2017.

[8] Jennifer L. Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*. Computational science and engineering. Philadelphia: SIAM, 2012.

[9] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Second edition. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer New York, NY, 2006.

[10] Jonathan Richard Shewchuk. "An introduction to the conjugate gradient method without the agonizing pain". In: *Technical Report CMU-CS-94-125* (1994). URL: `www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf`.