DISSERTATION

# COEFFICIENT RECOVERY IN PARABOLIC INITIAL BOUNDARY VALUE PROBLEMS

Submitted by

Trent C. Kull

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2006

COLORADO STATE UNIVERSITY

September 15, 2006

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY TRENT C. KULL ENTITLED "COEFFICIENT RECOVERY IN PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Eugene L. Allgower

Dr. Eugene Allgower

Dr. Greg Butters

Jennifer Mueller

Dr. Jennifer Mueller

**Adviser:** Dr. Paul DuChateau

Simon Tavener

**Department Head:** Dr. Simon Tavener

# ABSTRACT OF DISSERTATION

## COEFFICIENT RECOVERY IN PARABOLIC INITIAL BOUNDARY VALUE PROBLEMS

We consider the inverse problem of identifying spatially dependent coefficients in linear, parabolic, partial differential equations with specified initial and boundary conditions. We primarily explore the recovery of the diffusivity coefficient in the heat equation using a very limited amount of data on a portion of the spatial boundary. In the process, we show that the coefficient dependent observation operator is injective. With this knowledge, we use the Backus-Gilbert approach, modified with an adjoint method, to construct an approximate solution to the problem. We show the results of numerical experiments for the single, spatial variable case and then apply the method to related problems.

Trent C. Kull
Mathematics Department
Colorado State University
Fort Collins, CO 80523
Fall 2006

# ACKNOWLEDGEMENTS

I am indebted to my adviser, Dr. Paul DuChateau: A patient teacher, wonderful mentor, and outstanding mathematician. I thank Dr. Gene Allgower, Dr. Greg Butters, and Dr. Jennifer Mueller, who generously offered their time to serve on my committee and provide important suggestions for improvement. I am grateful to my friend and colleague, the soon to be Dr. Josh Ladd, for being a sounding board for ideas and sharing his unique mathematical insights. Finally, my thanks go to Kristen and Makenna for their ever present, unselfish support.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER 1

# INTRODUCTION

The identification of spatially dependent coefficients in partial differential equations is an important type of inverse problem. As examples, we might wish to recover the coefficients $s$ or $a$ in Equation (1.1) or the coefficients $s$ or $q$ in Equation (1.2).

$$s(x)\,\partial_t u(x,t) + \nabla \cdot [a(x)\,\nabla u(x,t)] = f(x,t) \tag{1.1}$$

$$s(x)\,\partial_t u(x,t) + \nabla^2 u(x,t) + q(x)\,u(x,t) = f(x,t) \tag{1.2}$$

To recover such coefficients, we associate specific initial and boundary values to the differential equation, the specific nature of which are modified as needed for theoretical needs or experimental constraints.

In Chapters 3 through 9, our efforts are directed at the recovery of $a$ in (1.1), with $s = -1$ and $f \equiv 0$. This problem has been widely studied and has applications in the context of groundwater flow, as the recovery of the hydraulic conductivity coefficient in a confined, heterogeneous aquifer [**3, 14**]. In Chapter 10, we briefly look at the recovery of $q$ in (1.2), which has been studied outside the context of

groundwater flow in publications such as [**1**]. Again we'll specifiy $s = -1$ and $f \equiv 0$.

A typical approach to inverse problem of these types is via output least squares, in linear (spatially dependent), nonlinear (state dependent), and multidimensional settings [**3, 11, 21, 22**]. Here the unknown coefficient is chosen from an appropriate space and the output computed by solving the direct problem. An error functional is then defined and minimized. Although powerful and widely applicable, the method has its share of disadvantages: Lack of uniqueness, false minima, parameter instability, for example.

We present an alternative to this method, drawing on the algorithm first introduced by geophysicists Backus and Gilbert in 1968 to invert seismic data in obtaining density profiles of the Earth [**2**]. The "Backus-Gilbert method" is commonly used to approximate the solution $f$ of the inverse problem

$$Af = g, \tag{1.3}$$

where (1.3) is an integral equation of the first kind. We'll see in Chapter 4 that this can involve an approximation of the $\delta$-distribution, and can be alternatively described as a projection method in Sobolev spaces [**12, 17, 18**]. We approximate a solution to our inverse problem using a modified Backus-Gilbert approach

[**5, 17**], as we attempt to reconstruct the coefficients at specified points in the spatial domain. We do so by using a very limited amount of information concerning the "solution" $u$ on the boundary. Contrast this with electric impedence tomography, which has better resolution but uses much more data for reconstruction, and is computationally more complex [**24**]. After the initial reconstruction algorithm in our modified method is complete, we are able to refine the recovery quite easily with minimal further effort. This proves advantageous over the method of output least squares, which requires a repeat of its entire algorithm, which can be computationally burdensome.

We provide a brief explanation of the Backus-Gilbert method in general and specify the modifications made here, in detail. We then look at the results of the process, making various adjustments to the algorithmic inputs in an attempt to optimize the recovery. Prior to this, however, we set the stage with necessary theory and problem formulation.

CHAPTER 2

# THE FUNCTION SPACES

## 2.1. The Hilbert Spaces: $H^s$

To ensure the proper setting for this problem, we must construct a set of function spaces in which to work [**6, 7, 15**]. We work with the complete inner product space $L^2(U)$ on a bounded set $U$. In light of our interest in (1.1) and (1.2), it will be convenient to consider the family of normalized eigenfunctions, $\{\phi_n\}$, generated by

$$-\nabla^2\phi_n(x,y) = \lambda_n\phi_n(x,y), \quad (x,y) \in U,$$

under certain boundary conditions to be specified. This family is an orthonormal basis for $L^2(U)$. We adopt the following notation for the inner product on $L^2(U)$,

$$(g,h)_0 = \int_U gh,$$

with induced norm

$$\|g\|_0 = \left(\int_U g^2\right)^{\frac{1}{2}}.$$

Relative to this basis, Fourier coefficients of $g \in L^2(U)$ are defined as

$$g_n := (g, \phi_n)_0 = \int_U g\phi_n.$$

We make use of the fact that a function is in $L^2(U)$ if and only if its Fourier coefficients are in $l^2$. In fact, this mapping of $g$ into $\{g_n\}$ is an isometric isomorphism from $L^2(U)$ onto $l^2$. Recall that for $\{g_n\}, \{h_n\} \in l^2$,

$$\begin{aligned} (\{g_n\}, \{h_n\})_{l^2} &= \sum_{n=1}^{\infty} g_n h_n, \quad \textit{and} \\ \|\{g_n\}\|_{l^2} &= \left(\sum_{n=1}^{\infty} g_n^2\right)^{\frac{1}{2}}. \end{aligned}$$

With these definitions, the isomorphism from $L^2(U)$ to $l^2$ preserves inner products and norms. Additionally, the partial sums

$$s_N := \sum_{n=1}^{N} g_n \phi_n$$

converge in $L^2(U)$ to $g$, i.e.

$$g = \sum_{n=1}^{\infty} g_n \phi_n.$$

Note that this defines the inverse of the isomorphism mentioned above.

We now have the tools to define the Hilbert spaces $H^s(U)$, which are going to be subspaces of regular functions from $L^2(U)$. For $s \geq 0$,

$$H^s(U) := \left\{ f \in L^2(U) : \sum_{n=1}^{\infty} (1+\lambda_n)^s f_n^2 < \infty \right\}.$$

We define inner products on these spaces in terms of their generalized Fourier coefficients, those in terms of the eigenfunctions $\phi_n$ :

$$(g,h)_s = \sum_{n=1}^{\infty} (1+\lambda_n)^s g_n h_n, \quad g,h \in H^s(U).$$

The corresponding induced norms on these spaces are

$$\|g\|_s = \left[ \sum_{n=1}^{\infty} (1+\lambda_n)^s g_n^2 \right]^{\frac{1}{2}}, \quad g \in H^s(U).$$

Note that for $s = 0$, we have that

$$H^0(U) = \left\{ g \in L^2(U) : \sum_{n=1}^{\infty} g_n^2 < \infty \right\}.$$

But since this is true for all functions in $L^2(U)$, we see that $H^0(U) = L^2(U)$. Then

$$(g,h)_0 = \sum_{n=1}^{\infty} g_n h_n = \int_U gh,$$

i.e. the $L^2(U)$ inner product and the $H^0(U)$ inner product are the same. Moreover, this shows the aforementioned isometry between the two spaces $L^2(U)$ and $l^2$,

$$\|g\|_0 = \|\{g_n\}\|_{l^2}.$$

### 2.2. The Hilbert Spaces: $H^{-s}$

We now define the spaces, $H^{-s}(U)$ for $s > 0$. Note that functions in $H^{-s}(U)$ may no longer be in $L^2(U)$. In fact, they may not even be functions in the classical sense. We instead look at generalized functions in the space of distributions on $U$, denoted $D'(U)$.

$$H^{-s}(U) := \left\{ \gamma \in D'(U) : \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \gamma_n^2 < \infty \right\}$$

Now, the fact that the eigenfunctions $\phi_n \in H^s(U)$ for all $s \geq 0$ ensures that the Fourier coefficients of $\gamma \in H^{-s}(U)$ are meaningful, i.e.

$$\gamma_n = \gamma(\phi_n) = \int_U \gamma \phi_n = \langle \gamma, \phi_n \rangle_{D'(U) \times D(U)}.$$

In other words, the Fourier coefficients $\gamma_n$ are well defined under the theory of distributions, with $\gamma$ acting on $\phi_n \in D(U)$, the set of test functions on $U$. The

inner product on $H^{-s}(U)$ is defined

$$(\gamma, \eta)_{-s} := \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \gamma_n \eta_n,$$

with the induced norm

$$\|\gamma\|_{-s} = \left[ \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \gamma_n^2 \right]^{\frac{1}{2}}.$$

### 2.3. Isomorphisms

We now show that the aforementioned spaces $H^s(U)$ and $H^{-s}(U)$, for a fixed $s$, are isomorphic. To do so, let $s \in \mathbb{R}$ and define the spaces of sequences $h^s$ as

$$h^s := \left\{ \{f_n\} \in l^\infty : (1+\lambda_n)^{s/2} f_n \in l^2 \right\}.$$

Note that $h^0 = l^2$. If we define the obvious inner product on $h^s$, we preserve the Hilbert space structure between the spaces $h^s$ and $H^s(U)$. In other words, the natural isomorphism between the function spaces $H^s(U)$ and $h^s$ is isometric. Letting $s \geq 0$,

$$\begin{aligned} E &: H^s(U) \to h^s, \\ E(g) &= \{(g, \phi_n)_0\} = \{g_n\}. \end{aligned}$$

Then

$$E^{-1}\left(\{g_n\}\right) = \sum_{n=1}^{\infty} g_n \phi_n.$$

For $s > 0$,

$$\begin{aligned} \hat{E} &: H^{-s}(U) \to h^{-s}, \\ \hat{E}(\gamma) &= \left\{ \langle \gamma, \phi_n \rangle_{D'(U) \times D(U)} \right\} = \{\gamma_n\}, \end{aligned}$$

and

$$\hat{E}^{-1}\left(\{\gamma_n\}\right) = \sum_{n=1}^{\infty} \gamma_n \phi_n.$$

For a fixed $s \geq 0$, we then define an isomorphism between the spaces $h^s$ and $h^{-s}$, in the following fashion.

$$\begin{aligned} j &: h^s \to h^{-s}, \\ j\left(\{g_n\}\right) &: = \{(1+\lambda_n)^s g_n\} \end{aligned}$$

Letting $j\left(\{g_n\}\right) = \{\gamma_n\}$, note that

$$\begin{aligned} \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \gamma_n^2 &= \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} (1+\lambda_n)^{2s} g_n^2 \\ &= \sum_{n=1}^{\infty} (1+\lambda_n)^{s} g_n^2 < \infty, \end{aligned}$$

i.e., $\{\gamma_n\}$ is indeed an element of $h^{-s}$. It is easy to see that $j$ is an isomorphism, with inverse $j^{-1}\{\gamma_n\} = \{g_n\}$ defined by

$$\{g_n\} = \left\{(1+\lambda_n)^{-s}\gamma_n\right\}.$$

Then (again, for a fixed $s \geq 0$) the isomorphism between $H^s(U)$ and $H^{-s}(U)$ is simply defined in terms of $E$ and $j$.

$$\begin{aligned} J &: H^s(U) \to H^{-s}(U) \\ J(g) &: = \left(\hat{E}^{-1} \circ j \circ E\right)(g) \end{aligned}$$

For $s \geq 0$, we can define an isomorphism between $h^s$ and $h^0$, and use this to define an isomorphism between $H^s(U)$ and $H^0(U)$.

$$\begin{aligned} j^{1/2} &: h^s \to h^0, \\ j^{1/2}(\{g_n\}) &: = \left\{(1+\lambda_n)^{s/2} g_n\right\}, \\ J^{1/2} &: H^s(U) \to H^0(U), \\ J^{1/2}(g) &: = \left(E^{-1} \circ j^{1/2} \circ E\right)(g). \end{aligned}$$

The isomorphism between $H^{-s}(U)$ and $H^0(U)$ is similarly defined:

$$\begin{aligned}
\hat{j}^{1/2} &: h^{-s} \to h^0, \\
\hat{j}^{1/2}(\{\gamma_n\}) &: = \left\{(1+\lambda_n)^{s/2}\gamma_n\right\}, \\
\hat{J}^{1/2} &: H^{-s}(U) \to H^0(U), \\
\hat{J}^{1/2}(g) &: = \left(\hat{E}^{-1} \circ \hat{j}^{1/2} \circ \hat{E}\right)(\gamma).
\end{aligned}$$

With these definitions, and noting that $s \in \mathbb{R}$, we see that

$$Jg = \gamma \Rightarrow J^{1/2}g = \hat{J}^{1/2}\gamma.$$

### 2.4. Duality

For a fixed $s \geq 0$, we denote the action of an arbitrary element $\eta \in H^{-s}(U)$ on $g \in H^s(U)$ as the duality pairing $\langle g, \eta \rangle_{H^s \times H^{-s}}$, and define this below. With $g = \sum_{n=1}^{\infty} g_n \phi_n$,

$$\begin{aligned}
\eta(g) &:= \sum_{n=1}^{\infty} g_n \eta(\phi_n) \\
&= \sum_{n=1}^{\infty} g_n \eta_n \\
&= \sum_{n=1}^{\infty} (1+\lambda_n)^{s/2} g_n (1+\lambda_n)^{-s/2} \eta_n
\end{aligned}$$

$$= \sum_{n=1}^{\infty} G_n H_n \qquad \left(j^{1/2}\left[\{g_n\}\right] = \{G_n\}, \quad \hat{j}^{1/2}\left[\{\eta_n\}\right] = \{H_n\}\right)$$

Noting that $\{G_n\}, \{H_n\} \in l^2$ correspond to $G = \sum_{n=1}^{\infty} (1+\lambda_n)^{s/2} g_n \phi_n \in H^0(U)$ and $H = \sum_{n=1}^{\infty} (1+\lambda_n)^{-s/2} \eta_n \phi_n \in H^0(U)$, we have that

$$\eta(g) = \langle g, \eta \rangle_{H^s \times H^{-s}} = \int_U GH = (G,H)_0 \in \mathbb{R}.$$

This shows that $\eta(g)$ is well defined. Moreover, we employ the Cauchy-Schwarz inequality to see that

$$\begin{aligned} |\eta(g)| &= \left| \sum_{n=1}^{\infty} (1+\lambda_n)^{s/2} g_n (1+\lambda_n)_n^{-s/2} \eta \right| \\ &\leq \left[ \sum_{n=1}^{\infty} (1+\lambda_n)^{s} g_n^2 \right]^{\frac{1}{2}} \left[ \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \eta_n^2 \right]^{\frac{1}{2}} \\ &= \|G\|_0 \|H\|_0 = \|g\|_s \|\eta\|_{-s} \\ &= c \|g\|_s \end{aligned}$$

[**15**].

We have verified that $\eta \in H^{-s}(U)$ is a continuous linear functional on $H^s(U)$, i.e. $\eta$ is an element of the dual space of $H^s(U)$, denoted $[H^s(U)]'$. Now since the set of test functions, $C_c^{\infty}(U)$, is dense in $H^s(U)$ for every $s \geq 0$, and has as a dual the set of distributions, $D'(U)$, we see that the dual of any space $H^s(U)$ is

a subspace of the distributions, consisting of those distributions which satisfy the condition given in the definition of $H^{-s}(U)$ above.

Now, using the defined duality pairings and isomorphisms mentioned above, we can conveniently write various equivalent expressions for a given duality pairing or inner product.

$$\begin{aligned}
\langle g, \eta \rangle_{H^s \times H^{-s}} &= \sum_{n=1}^{\infty} \eta_n g_n \\
&= \sum_{n=1}^{\infty} (1+\lambda_n)^s h_n g_n \qquad (j[\{h_n\}] = \{\eta_n\}) \\
&= (h, g)_s
\end{aligned}$$

$$\begin{aligned}
\langle g, \eta \rangle_{H^s \times H^{-s}} &= \sum_{n=1}^{\infty} g_n \eta_n \\
&= \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} \gamma_n \eta_n \qquad (j^{-1}[\{\gamma_n\}] = \{g_n\}) \\
&= (\gamma, \eta)_{-s} \ .
\end{aligned}$$

Thus we can write the following equivalencies, for example:

$$\langle g, \eta \rangle_{H^s \times H^{-s}} = \left(g, J^{-1}\eta\right)_s = (g, h)_s = \left(J^{1/2}g, J^{1/2}h\right)_0$$

$$= (G, H)_0$$

$$= \left(\hat{J}^{-1/2}\gamma, \hat{J}^{-1/2}\eta\right)_0 = (\gamma, \eta)_{-s} = (\gamma, Jh)_{-s}$$

$$= \langle \gamma, h \rangle_{H^s \times H^{-s}} .$$

Now let $\beta$ be an arbitrary element of $[H^s(U)]'$. Then by the Riesz Representation Theorem, there exists a unique element $g_\beta \in H^s(U)$ such that $\beta(u) = (u, g_\beta)_s$, $\forall u \in H^s(U)$ [**15**]. Then, defining $Jg_\beta = \gamma$, we have $\forall u \in H^s(U)$ that $(u, g_\beta)_s = \langle u, \gamma \rangle_{H^s \times H^{-s}}$. This defines an isomorphism between the spaces $[H^s(U)]'$ and $H^{-s}(U)$, i.e., $H^{-s}(U)$ is the dual of $H^s(U)$: $\langle u, \beta \rangle_{H^s \times (H^s)'} = \langle u, \gamma \rangle_{H^s \times H^{-s}}$.

### 2.5. Regularity

Denoting the dual spaces with a prime, we note the following containment between the following Hilbert spaces on the set $U$:

$$\cdots \supset H^{-2} = \left(H^2\right)' \supset H^{-1} = \left(H^1\right)' \supset H^0 \supset H^1 \supset H^2 \supset \cdots$$

This is precisely the containment observed with the Hilbert-Sobolev spaces typically denoted

$$W^{k,2}(U) = H^k(U) .$$

In fact, we have merely characterized these Hilbert-Sobolev spaces in terms of Fourier coefficients (versus containment of weak derivatives in $L^2(U)$), with the

inclusion of certain boundary conditions built into the space by the generated set of eigenfunctions [9].

For a function $g \in H^s(U)$, it can be shown that $\alpha-$order differentiation causes a $\alpha$-degree loss of regularity. For example,

$$g \in H^1(U) \Rightarrow \nabla g \in H^0(U) \quad and \quad \nabla^2 g \in H^{-1}(U).$$

Of course, integration has an opposite, smoothing, effect.

Finally, with $\gamma \in H^{-s}(U)$ and $g \in H^s(U)$, we interpret the integral $\int_U \gamma g$ as the duality pairing defined above:

$$\int_U \gamma g = \langle g, \gamma \rangle_{H^s \times H^{-s}}.$$

With the appropriate function spaces constructed, we can determine the appropriate setting for our problem.

CHAPTER 3

# FORMULATION OF THE INVERSE PROBLEM

Our objective here is to identify a single unknown coefficient in a parabolic PDE from measurements made on the boundary of the domain. In particular, we consider the identification of $a(x)$ in (1.1), specifically

$$\partial_t u(x,t) = \nabla \cdot [a(x) \nabla u(x,t)], \quad (x,t) \in U_T.$$

Coefficient recovery from observed boundary data is a problem widely studied in a number of contexts; see [**1, 11**]. We let $U$ denote a bounded open set in $\mathbb{R}^n$ with smooth boundary $\Gamma$, which is composed of two non-overlapping complementary parts, $\Gamma^D$ and $\Gamma^N$:

$$\begin{aligned} U_T &= U \times (0,T), \qquad (3.1) \\ \Gamma &= \Gamma^D \cup \Gamma^N = \text{boundary of } U, \\ \Gamma_T &= \Gamma \times (0,T). \end{aligned}$$

For $n = 1, 2$, this PDE governs dynamic groundwater flow for an isotropic, homogeneous confined aquifer, where

$$\begin{aligned} u(x,t) &= \text{hydraulic head in } U, \\ a(x) &= \text{transmissivity in } U. \end{aligned}$$

The direct problem, with spatial boundary $\Gamma = \Gamma^N \cup \Gamma^D$ $\left(\Gamma^N \cap \Gamma^D = \varnothing\right)$ is

$$\begin{aligned} \partial_t u(x,t) &= \nabla \cdot [a(x) \nabla u(x,t)], \quad (x,t) \in U_T \qquad (3.2) \\ u(x,0) &= 0, \quad x \in U \\ u(x,t) &= f_D(x,t), \quad (x,t) \in \Gamma_T^D \\ a(x) \nabla u(x,t) \cdot n &= f_N(x,t), \quad (x,t) \in \Gamma_T^N. \end{aligned}$$

Now suppose $v = v(x,t)$ solves (3.2) in the case $a(x) = a_0 = \text{constant}$, and $u = u(x,t)$ solves (3.2) in the case $a(x) = a_0 + \tau(x)$. Then $z(x,t) = u(x,t) - v(x,t)$ solves

$$\partial_t z - \nabla \cdot (a \nabla z) = \nabla \cdot (\tau \nabla v), \quad (x,t) \in U_T \qquad (3.3)$$

with homogeneous conditions

$$z(x,0) = 0, \quad x \in U$$

$$z(x,t) = 0, \quad (x,t) \in \Gamma_T^D$$

$$[a(x)\nabla u(x,t) - a_0 \nabla v(x,t)] \cdot n = 0, \quad (x,t) \in \Gamma_T^N.$$

The derivation of this "difference problem" is located in Appendix 1.1.

### 3.1. The Integral Equation

We now multiply Equation (3.3) by a smooth function $w(x,t)$,

$$\int\int_{U_T} w\partial_t z - \int\int_{U_T} w\nabla \cdot (a\nabla z) = \int\int_{U_T} w\nabla \cdot (\tau \nabla v),$$

and integrate by parts.

$$\begin{aligned} & \int_U \left[ wz|_0^T - \int_0^T z\partial_t w dt \right] dx \\ & - \int_0^T \left[ \int_\Gamma wa\nabla z \cdot n ds - \int_U a\nabla w \cdot \nabla z dx \right] dt \\ = & \int_0^T \left[ \int_\Gamma w\tau \nabla v \cdot n ds - \int_{U_T} \tau \nabla w \cdot \nabla v dx \right] dt. \end{aligned}$$

Then, requiring $w(x,T) = 0$, applying boundary conditions for $u$ and $v$, and further integrating by parts, we have

$$\begin{aligned} & \int\int_{U_T} z\left[-\partial_t w - \nabla\cdot(a\nabla w)\right] dxdt \\ & + \int_0^T \int_{\Gamma^D} w(a_0\nabla v - a\nabla u)\cdot n ds dt + \int_0^T \int_{\Gamma^N} za\nabla w\cdot n ds dt \\ = & -\int\int_{U_T} \tau\nabla w\cdot\nabla v dx dt. \end{aligned} \tag{3.4}$$

### 3.2. The Adjoint Problem

Now suppose that $w(x,t)$ solves the associated adjoint problem:

$$\begin{aligned} -\partial_t w - \nabla\cdot(a\nabla w) &= 0, \quad (x,t)\in U_T \\ w(x,T) &= 0, \quad x\in U \\ w(x,t) &= g^*(x,t), \quad (x,t)\in\Gamma_T^D \\ a(x)\nabla w(x,t)\cdot n &= h^*(x,t), \quad (x,t)\in\Gamma_T^N. \end{aligned}$$

Then the integral identity (3.4) can be rewritten as

$$\begin{aligned} & \int_0^T \int_{\Gamma^D} g^*(a_0\nabla v - a\nabla u)\cdot n ds dt + \int_0^T \int_{\Gamma^N} zh^* ds dt \\ = & -\int\int_{U_T} \tau\nabla w\cdot\nabla v dx dt. \end{aligned} \tag{3.5}$$

At this point, we can choose adjoint inputs $g^*$ and $h^*$, to specify (3.5) as an expression relating the coefficient differences to various observation differences:

(1) With $h^* = 0$, (3.5) reduces to

$$\int_0^T \int_{\Gamma^D} g^*(a\nabla u - a_0 \nabla v) \cdot n ds dt = \int_U \tau \int_0^T \nabla w \cdot \nabla v dt dx.$$

At this point, we can rewrite this integral identity in inner product and operator notation as

$$(g^*, G_N(a) - G_N(a_0))_{L^2(\Gamma_T^D)} = \left(\tau, \int_0^T \nabla w \cdot \nabla v dt\right)_{L^2(U)}, \qquad (3.6)$$

where

$$G_N(a) = a(x) \nabla u(x,t,a), \quad (x,t) \in \Gamma_T^D.$$

(2) With $g^* = 0$, (3.5) reduces to

$$\int_0^T \int_{\Gamma^N} (v-u) h^* ds dt = \int_U \tau \int_0^T \nabla w \cdot \nabla v dt dx.$$

This can be rewritten as

$$(h^*, G_D(a_0) - G_D(a))_{L^2(\Gamma_T^N)} = \left(\tau, \int_0^T \nabla w \cdot \nabla v dt\right)_{L^2(U)}, \qquad (3.7)$$

where $G_D(a) = u(x,t,a), \quad (x,t) \in \Gamma_T^N$.

Note that (3.6)and (3.7) involve equations with observation differences on the left and coefficient differences on the right; i.e. they relate the two differences. If $\Gamma = \Gamma^N$, then $\Gamma^D = \varnothing$ and we impose a Neumann condition and observe Dirichlet data over the entire boundary. Conversely, if $\Gamma = \Gamma^D$, then $\Gamma^N = \varnothing$ and we impose a Dirichlet condition and observe Neumann data over the entire boundary. If $\Gamma^D \neq \varnothing$ and $\Gamma^N \neq \varnothing$, then mixed conditions are imposed and we can make observations on only one portion of the boundary or its entirety, at our option. In the latter case, we can rewrite (3.5) as

$$(g^*, \Delta G_N)_{L^2(\Gamma_T^D)} + (h^*, \Delta G_D)_{L^2(\Gamma_T^N)} = \left(\tau, \int_0^T \nabla w \cdot \nabla v dt\right)_{L^2(U)}.$$

We note that this is a well posed initial boundary value problem, given that the coefficient $a \in L^\infty(U)$ is such that $a(x) \geq k_1 > 0$ for $x \in U$. Then this has a unique solution $u = u(x,t,a)$ such that $u \in L^2[(0,T) : H^1(U)] \cap C[[0,T] : H^0(U)]$, with $f_D(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$ and $f_N(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$.

### 3.3. The Spatial Setting

Revisiting the function space discussion in Chapter 2, we must ensure that our weak formulation of the problem takes place in the proper setting for a valid

solution. Since we have a mixed boundary value problem, it is sufficient (but not necessary) to build the spaces $H^s(U)$ using a family of eigenfunctions, $\{\phi_n\}$, generated with mixed boundary conditions. For $s \geq 1$, we set $H^s(U)$ as the appropriate solution space: $u, v, w \in H^s(U)$. Established trace operator theory demands that $f_D, g^* \in H^{s-1/2}\left(\Gamma^D\right)$ and $f_N, h^* \in H^{s-1-1/2}\left(\Gamma^N\right)$ necessitating minimum regularities of $H^{1/2}\left(\Gamma^D\right)$ and $H^{-1/2}\left(\Gamma^N\right)$, respectively [7]. Moreover, we see that $\nabla u, \nabla v, \nabla w \in H^{s-1}(U)$, and hence are contained in $H^0(U)$.

Determining the appropriate space for $\nabla w \cdot \nabla v$ requires a bit more work. We make use of the following lemmas.

**Lemma 1.** $f, g \in L^2(U) \Rightarrow fg \in L^1(U)$

**Proof.** Note that we can't directly use the Cauchy-Schwarz inequality, as it concerns the expression $\left|\int_u fg\right|$ and we must show $\int_u |fg|$ is finite. However, we can appeal to the familiar argument used in the proof of the Cauchy-Schwarz inequality: Letting $\alpha > 0$, consider the integral $\int_U (|f| + \alpha |g|)^2$ and note that

$$\begin{aligned} & \int_U (|f| + \alpha |g|)^2 \\ = & \int_U \left(|f|^2 + 2\alpha |f||g| + \alpha^2 |g|^2\right) \\ = & \left(\int_U |g|^2\right)\alpha^2 + \left(2\int_U |f||g|\right)\alpha + \int_U |f|^2 . \end{aligned}$$

Because this quadratic function of $\alpha$ is nonnegative, its discriminant is nonpositive:

$$\left(2\int_U |f|\,|g|\right)^2 - 4\left(\int_U |g|^2\right)\int_U |f|^2 \leq 0.$$

This implies that

$$\left(\int_U |fg|\right)^2 \leq \int_U |g|^2 \int_U |f|^2.$$

Then, since $f, g \in L^2(U)$, we have that each integral is finite. Thus, $\int_U |fg| < \infty$, i.e., the product $fg \in L^1(U)$. □

Now, in the setting of our problem, we note that the solution to

$$-\nabla^2 \phi_n(x) = \lambda_n \phi_n(x), \quad x \in U$$

under certain boundary conditions produces eigenvalues of the form $\lambda_n \sim n^2$. In this case, we make the following claim.

**Lemma 2.** *$f \in L^1(U) \Rightarrow f \in H^{-s}(U)$, for all $s \geq 1$.*

**Proof.** $f \in L^1(U)$ implies $\{f_n\} \in l^\infty$, or $|f_n| \leq c_f$, for all $n$. Then

$$\sum_{n=1}^{\infty} (1+\lambda_n)^{-s} f_n^2 \leq c_f^2 \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} < \infty,$$

since $\lambda_n \sim n^2 \Rightarrow \sum_{n=1}^{\infty} (1+\lambda_n)^{-s} < \infty$. Then $\left\{(1+\lambda_n)^{-s/2} f_n\right\} \in l^2$, or $f \in H^{-s}(U)$, for $s \geq 1$. □

So, with $w, v \in H^s(U)$, we have that $\partial_i w, \partial_i v \in H^{s-1}(U) \subset H^0(U)$, which implies that $\nabla w \cdot \nabla v$ is linearly comprised of products of $L^2(U)$ functions. Using the claims above, we have that $\partial_i w \partial_i v \in L^1(U)$, and thus $\nabla w \cdot \nabla v \in L^1(U)$. Then $\int_0^T \nabla w \cdot \nabla v dt \in H^{-s}(U)$ for $s \geq 1$, allowing $\tau \in H^s(U)$.

We can then rewrite (3.5) using duality pairings and minimum regularities, for $s \geq 1$:

$$\left\langle \tau, \int_0^T \nabla w \cdot \nabla v dt \right\rangle_{H^S(U) \times H^{-S}(U)} \tag{3.8}$$
$$= \int_0^T \left[ \langle g^*, \triangle G_N \rangle_{H^{1/2}(\Gamma^D) \times H^{-1/2}(\Gamma^D)} + \langle h^*, \triangle G_D \rangle_{H^{-1/2}(\Gamma^N) \times H^{1/2}(\Gamma^N)} \right] dt.$$

Similarly, we can rewrite (3.6) and (3.7) using duality pairings, instead of inner products. Note that the terms $\triangle G_N$ represent the flow difference observations on the Dirichlet boundaries, and $\triangle G_D$ the state difference observations on the Neumann boundaries.

### 3.4. One-Dimensional Formulation

Although the above formulation is for the general n-dimensional spatial setting, much of our work in this paper will be done with just a single dimension. As such,

we detail the problem formulation for this setting, highlighting the characteristics associated with various boundary conditions.

Specifying the above work, we have the following differential equation with homogeneous initial condition.

$$\begin{aligned} \partial_t u(x,t) &= \partial_x \left[ a(x) \partial_x u(x,t) \right], \quad (x,t) \in U_T \\ u(x,0) &= 0, \quad x \in U \end{aligned}$$

We'll allow the type of boundary conditions to vary according to experimental preference. In particular, we will impose either state (Dirichlet) or flow (Neumann) conditions, or a mix, on the boundary. Our associated comparison problem is

$$\partial_t v(x,t) = a_0 \partial_{xx} v(x,t), \quad (x,t) \in U_T.$$

As discussed in the general case above, we define $\tau(x) = a(x) - a_0$ and $z(x,t) = u(x,t) - v(x,t)$ and construct the difference equation

$$\partial_t z - \partial_x (a \partial_x z) = \partial_x (\tau \partial_x v).$$

In any case, the associated initial and boundary conditions satisfied by $z(x,t)$ are homgeneous. Note that recovering $\tau(x)$ is now the goal, since it is related to $a(x)$ by a known constant.

Employing the adjoint method, we multiply the difference equation above by a smooth function $w(x,t)$

$$\int\int_{U_T} w\partial_t z - \int\int_{U_T} w\partial_x\left(a\partial_x z\right) = \int\int_{U_T} w\partial_x\left(\tau\partial_x v\right)$$

and integrate by parts. Applying the homogeneous final condition on $w$, we arrive at

$$\begin{aligned} & -\int\int_{U_T} z\partial_t w dx dt \qquad (3.9) \\ & -\int_0^T \left[a(1)\, w(1,t)\,\partial_x z(1,t) - a(0)\, w(0,t)\,\partial_x z(0,t) - \int_U a\partial_x w\partial_x z dx\right] dt \\ = & \int_0^T \left[\tau(1)\, w(1,t)\,\partial_x v(1,t) - \tau(0)\, w(0,t)\,\partial_x v(0,t) - \int_U \tau\partial_x w\partial_x v dx\right] dt. \end{aligned}$$

Specific boundary conditions now necessitate separate analysis for each case: mixed, Dirichlet, and Neumann. In all cases, we adopt the notation $B_0$ or $B_1$, respectively, for the imposed boundary conditions at $x = 0$ and $x = 1$ in the direct problem. Similarly, we use the notation $G_0$ and $G_1$ for the corresponding observation operators. These selected conditions also determine the type of boundary

conditions present in the associated adjoint problem. Detailed computations for each case are located in Appendix 2.

### 3.4.1. Mixed Boundary Conditions

We first choose mixed conditions for the direct problem, specifically a Dirichlet condition at $x = 0$ and Neumann at $x = 1$. We define $B_0[u] = u(0,t)$ and $B_1[u] = a(1)\partial_x u(1,t)$, and note that for $B_1[v]$, $a(1) = a_0$. The associated difference problem is

$$\partial_t z - \partial_x (a\partial_x z) = \partial_x (\tau \partial_x v)$$

$$z(x,0) = z(0,t) = 0,$$

$$a(1)\partial_x u(1,t) - a_0 \partial_x v(1,t) = 0.$$

After integrating (3.9) by parts, we expand $\tau(x)$ and use the fact that $z = u - v$. Rewriting, we have

$$-\int\int_{U_T} z\partial_t w dx dt - \int_0^T a(1) w(1,t) [\partial_x u(1,t) - \partial_x v(1,t)] dt \qquad (3.10)$$
$$+ \int_0^T a(0) w(0,t) [\partial_x u(0,t) - \partial_x v(0,t)] dt + \int\int_{U_T} (a\partial_x w) \partial_x z dx dt$$

$$= \int_0^T [a(1) - a_0] w(1,t) \partial_x v(1,t) dt - \int_0^T [a(0) - a_0] w(0,t) \partial_x v(0,t) dt$$
$$- \int_0^T \int_U \tau \partial_x w \partial_x v dx dt.$$

Integrating by parts again, we see that

$$\int \int_{U_T} (a\partial_x w) \partial_x z dx dt$$
$$= \int_0^T \left[ a(1) z(1,t) \partial_x w(1,t) - a(0) z(0,t) \partial_x w(0,t) - \int_U z \partial_x (a \partial_x w) dx \right] dt.$$

Using this result and regrouping terms, we rewrite (3.10) as

$$\int \int_{U_T} z [-\partial_t w - \partial_x (a \partial_x w)] dx dt$$
$$- \int_0^T w(1,t) [a(1) \partial_x u(1,t) - a(1) \partial_x v(1,t)] dt$$
$$+ \int_0^T w(0,t) [a(0) \partial_x u(0,t) - a(0) \partial_x v(0,t)] dt$$
$$+ \int_0^T a(1) \partial_x w(1,t) [u(1,t) - v(1,t)] dt$$
$$= \int_0^T [a(1) - a_0] w(1,t) \partial_x v(1,t) dt - \int_0^T [a(0) - a_0] w(0,t) \partial_x v(0,t) dt$$
$$- \int_0^T \int_U \tau \partial_x w \partial_x v dx dt.$$

Rearranging and cancelling terms, we have

$$\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dxdt \\
&+\int_0^T w\left(1,t\right)\left[a_0\partial_x v\left(1,t\right)-a\left(1\right)\partial_x u\left(1,t\right)\right]dt \\
&+\int_0^T w\left(0,t\right)\left[a\left(0\right)\partial_x u\left(0,t\right)-a_0\partial_x v\left(0,t\right)\right]dt \\
&+\int_0^T a\left(1\right)\partial_x w(1,t)\left[u\left(1,t\right)-v\left(1,t\right)\right]dt \\
=\ &-\int_0^T\int_U \tau\partial_x w\partial_x v dxdt.
\end{aligned}$$

Noting the second integral vanishes per the difference conditions, we arrive at the following useful integral identity.

$$\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dxdt+\int_0^T w\left(0,t\right)\left[a\left(0\right)\partial_x u\left(0,t\right)-a_0\partial_x v\left(0,t\right)\right]dt \\
&+\int_0^T a\left(1\right)\partial_x w(1,t)\left[u\left(1,t\right)-v\left(1,t\right)\right]dt=-\int_0^T\int_U \tau\partial_x w\partial_x v dxdt.
\end{aligned}$$

Now suppose that $w\left(x,t\right)$ solves the associated adjoint problem.

$$\begin{aligned}
-\partial_t w-\partial_x\left(a\partial_x w\right) &= 0 \qquad (3.11)\\
w\left(x,T\right) &= 0
\end{aligned}$$

$$B_0[w] = w(0,t) = g^*(t)$$

$$B_1[w] = a(1)\,\partial_x w(1,t) = h^*(t)$$

This gives us the identity

$$\int_0^T B_0[w]\,(G_0[a_0] - G_0[a])\,dt + \int_0^T B_1[w]\,(G_1[a_0] - G_1[a])\,dt$$
$$= \int_0^T \int_U \tau \partial_x w \partial_x v dx dt.$$

Note that our observation operators in this case are $G_0[a] = a(0)\,\partial_x u(0,t)$ and $G_1[a] = u(1,t)$.

### 3.4.2. Dirichlet Boundary Conditions

Choosing Dirichlet conditions on the entire boundary, this time we have $B_0[u] = u(0,t)$ and $B_1[u] = u(1,t)$, with associated difference problem boundary conditions $z(0,t) = z(1,t) = 0$. Integration by parts leads to an associated adjoint problem

$$-\partial_t w - \partial_x(a\partial_x w) = 0,$$

$$w(x,T) = 0,$$

$$B_0[w] = w(0,t) = g^*(t),$$

$$B_1[w] = w(1,t) = h^*(t),$$

and the integral identity

$$\int_0^T B_0[w](G_0[a_0] - G_0[a])\,dt + \int_0^T B_1[w](G_1[a] - G_1[a_0])\,dt$$
$$= \int\int_{U_T} \tau \partial_x w \partial_x v dx dt.$$

Note that the defined observation operators are $G_0[a] = a(0)\partial_x u(0,t)$ and $G_1[a] = a(1)\partial_x u(1,t)$.

### 3.4.3. Neumann Boundary Conditions

We next choose Neumann conditions on the boundary: $B_0[u] = a(0)\partial_x u(0,t)$ and $B_1[u] = a(1)\partial_x u(1,t)$. This time we use the adjoint problem

$$-\partial_t w - \partial_x(a\partial_x w) = 0,$$

$$w(x,T) = 0,$$

$$B_0[w] = a(0)\partial_x w(0,t) = g^*(t),$$

$$B_1[w] = a(1)\partial_x w(1,t) = h^*(t).$$

and have the integral identity

$$\int_0^T B_0[w]\left(G_0[a]-G_0[a_0]\right)dt+\int_0^T B_1[w]\left(G_1[a]-G_1[a_0]\right)dt$$
$$=\int\int_{U_T}\tau\partial_x w\partial_x v dxdt.$$

Note that $G_0[a]=u(0,t)$ and $G_1[a]=u(1,t)$.

We can write a general expression that highlights the relationship between the boundary data, $B$, in both the direct and adjoint problems, and the observed data $G$.

$$\int_0^T B_0\Delta G_0 dt+\int_0^T B_1\Delta G_1 dt=\int\int_{U_T}\tau\partial_x w\partial_x v dxdt,$$

where the $\Delta G_x$ are the observation differences on the appropriate portion of the boundary. This form of the identification makes use of observations at both ends of the interior. If only one or the other observation is available, we can cause the unavailable observation to drop out of the identification by choosing the appropriate adjoint input to be zero.

$$\int_0^T B_0\Delta G_0 dt=\int\int_{U_T}\tau\partial_x w\partial_x v dxdt$$
$$\int_0^T B_1\Delta G_1 dt=\int\int_{U_T}\tau\partial_x w\partial_x v dxdt$$

### 3.5. Identifiability

For simplicity, we consider identifying the unknown coefficient in one spatial dimension, which can then be generalized to higher dimensions. For convenience, we work with the intial (mixed) boundary value problem (3.12).

$$\begin{aligned} \partial_t u(x,t) - \partial_x \left[ a(x) \partial_x u(x,t) \right] &= 0 \quad (x,t) \in U_T \qquad (3.12) \\ u(x,0) &= 0, \quad x \in U \\ u(0,t) &= f(t), \quad t \in (0,T) \\ \partial_x u(1,t) &= 0, \quad t \in (0,T) \end{aligned}$$

The proof that follows can be generalized to Dirichlet and Neumann conditions. We note that identifiability in this problem can be shown by the method of Carleman estimates [**13**]. Here we provide an alternate method which employs the adjoint techniques used in the subsequent recovery process, and more importantly, can be modified to the case where $a$ is state dependent as well: $a = a(u)$.

Of course we must impose conditions on the coefficient $a = a(x)$ sufficient to guarantee that the problem is parabolic; e.g. it is enough to suppose that for positive constants $\alpha_1 > \alpha_0$, we have $\alpha_1 \geq a(x) \geq \alpha_0$ a.e., $x \in U$. In this case we will say that the coefficient $a(x)$ is admissible. This problem has a unique

solution $u = u(x,t;a)$ for every admissible $a(x) \in L^\infty(U)$ and each input $f(t)$ in $C(0,T)$. For a fixed input $f(t)$ in $C(0,T)$, we can consider the inverse problem of identifying the unknown diffusivity function $a(x)$ from the boundary observation $a(0)\,\partial_x u(0,t;a)$. Since $u$ is the solution of a parabolic initial boundary value problem, it is smooth with respect to $x$ and $t$ on the domain. Hence we can define the observation operator by $G[a] = a(0)\,\partial_x u(0,t;a) \in C(0,T) \subset L^2(0,T)$, where $G : L^\infty(U) \to L^2(0,T)$. It is evident that $G$ is not a linear mapping; in particular, $G[0]$ is not even defined. It is also evident that if $\partial_x u(x,t)$ vanishes on any open subset of $U_T$ then $a(x)$ can assume any values whatever on this set without affecting the observation values. Then, in order for $a(x)$ to be identifiable from the observation $G[a]$, the boundary input $f(t)$ must be chosen such that the gradient $\partial_x u(x,t)$ vanishes on no open subset of $U_T$. That $f$ can be chosen in this fashion follows from the following Lemma.

**Lemma 3.** *Suppose that for $a = a(x)$ an admissible diffusivity coefficient, $u = u(x,t)$ solves the IBVP (3.12) above for a boundary input $f(t)$ satisfying $f \in C^1(0,T)$ such that $f'(t) > 0$ for $t > 0$. Then $\partial_x u \in C(U_T)$ is such that for each $t^* \in (0,T)$ and every open interval $(x_1,x_2) \subset U$, $\partial_x u(x,t^*) < 0$ for $x_1 \le x \le x_2$.*

**Proof.** Multiply the differential equation (3.12) by $\partial_x \phi(x,t)$, where $\phi = \phi(x,t)$ is a smooth function, and integrate over the domain.

$$\int\int_{U_T} \{\partial_t u(x,t) - \partial_x [a(x)\,\partial_x u(x,t)]\}\,\partial_x \phi(x,t)\,dxdt = 0$$

If we integrate by parts and use the fact that $\partial_x \partial_t \phi(x,t) = \partial_t \partial_x \phi(x,t)$, we arrive at

$$\begin{aligned} &\int\int_{U_T} (\partial_t \phi + a\partial_{xx}\phi)\,\partial_x u dxdt + \int_U u\partial_x \phi|_0^T dx \\ &- \int_0^T u\partial_t \phi|_0^1 dt - \int_0^T a\partial_x u\partial_x \phi|_0^1 dt = 0. \end{aligned} \tag{3.13}$$

(Details are in Appendix 3.) We require $\phi(x,t)$ to solve the following adjoint problem.

$$\begin{aligned} \partial_t \phi + a\partial_{xx}\phi &= F(x,t), \qquad (x,t) \in U_T \\ \phi(x,T) &= 0, \qquad x \in U \\ \phi(0,t) &= \phi(1,t) = 0, \qquad t \in (0,T) \end{aligned}$$

These conditions imply that $\partial_x \phi(x,T) = \partial_t \phi(0,t) = \partial_t \phi(1,t) = 0$. Then, using the homogeneous conditions from the direct problem, (3.13) reduces to

$$\int\int_{U_T} F(x,t)\,\partial_x u dx dt = -\int_0^T a(0)\,\partial_x u(0,t)\,\partial_x \phi(0,t)\,dt.$$

The maximum principle, applied to the adjoint problem, implies that if $F(x,t)$ is any function satisfying $F(x,t) \geq 0$ for $0 < x < 1$ and $0 < t < T$, then $\phi(x,t) < 0$ on this same set and $\phi(0,t) = 0$ for $0 < t < T$; it then follows that $\partial_x \phi(0,t) < 0$ on this same time domain [**6**]. Similarly, since $f'(t) > 0$ for $t > 0$ and $u(x,0) = 0$, we have that $\partial_x u(0,t) < 0$ for $0 < t < T$. All this taken together implies that for $F(x,t) \geq 0$,

$$\int_0^T a(0)\,\partial_x u(0,t)\,\partial_x \phi(0,t)\,dt > 0$$

and thus

$$\int\int_{U_T} F(x,t)\,\partial_x u dx dt < 0.$$

This is just the assertion that $\partial_x u(x,t) < 0$ in the sense of distributions, but since $u(x,t)$ is the solution of a parabolic initial boundary value problem, it possesses additional smoothness. In particular, the continuity of $\partial_x u(x,t)$ ensures it is negative in the usual pointwise sense as well. □

Now, as in the previous formulation, we let $v = v(x,t;a_0)$ be the solution of the direct problem with diffusivity coefficient $a_0 = a_0(x)$. Then denoting $z(x,t) = (u-v)(x,t)$ and $a(x) = a_0(x) + \tau(x)$, we can form the difference problem

$$\begin{aligned} \partial_t z(x,t) - \partial_x [a(x)\,\partial_x z(x,t)] &= \partial_x [\tau(x)\,\partial_x v], \qquad (x,t) \in U_T \qquad (3.14) \\ z(x,T) &= 0, \qquad x \in U \\ z(0,t) &= \partial_x z(1,t) = 0, \qquad t \in (0,T). \end{aligned}$$

We now let $w(x,t;g^*)$ denote the unique solution to the following backward parabolic problem

$$\begin{aligned} -\partial_t w(x,t) - \partial_x [a_0(x)\,\partial_x w(x,t)] &= 0, \qquad (x,t) \in U_T \qquad (3.15) \\ w(x,T) &= 0, \qquad x \in U \\ w(0,t) &= g^*(t), \qquad t \in (0,T) \\ \partial_x w(1,t) &= 0, \qquad t \in (0,T). \end{aligned}$$

Then multiplying our difference equation (3.14) by $w(x,t)$ and integrating by parts, we can show that

$$\int_0^T g^*(t)\left(G[a]-G[a_0]\right)dt = -\int\int_{U_T} \tau\partial_x w\partial_x v dxdt.$$

Given this, we can prove the following theorem.

**Theorem 4.** *Suppose $f$ in the problem above satisfies the requirements from the previous lemma. Let $a, a_0$ be two admissible diffusivity coefficients in $L^\infty(U)$ which are not identical in the sense that there is some open interval $J \subset U$ of positive measure where $\tau(x) = (a-a_0)(x) \geq \alpha > 0$ a.e. on $J$. Then $G[a] \neq G[a_0]$.*

**Proof.** Let $J = (x_1, x_2)$ and suppose that $G[a] = G[a_0]$. Then it follows from above that

$$\int\int_{U_T} \tau\partial_x w\partial_x v dxdt = 0$$

for all $g^* \in L^2(0,T)$. Now, if $w(x,t)$ solves the backward parabolic problem (3.15), then $\partial_t w(x,t)$ solves the same with $g^*(t)$ replaced by the derivative $dg^*/dt$. Then the fact that $\partial_x(\partial_t w) = \partial_t(\partial_x w)$ gives

$$\int\int_{U_T} \tau\partial_x v\partial_t(\partial_x w)\,dxdt = 0.$$

Now, since $v = v(x,t;a_0)$ is the smooth solution to a parabolic initial value problem, we can employ the mean value theorem to determine for some $t^* \in (0,T)$ that

$$\int\int_{U_T} \tau \partial_x v \partial_t (\partial_x w)\, dxdt = \int_U \tau(x)\, \partial_x v(x,t^*) \int_0^T \partial_t (\partial_x w)\, dxdt = 0.$$

Then

$$\int_U \tau(x)\, \partial_x v(x,t^*) [\partial_x w(x,T) - \partial_x w(x,0)] dx = 0$$

and $w(x,T) = 0$ gives

$$\int_U \tau(x)\, \partial_x v(x,t^*)\, \partial_x w(x,0)\, dx = 0.$$

We next define the smooth, positive function $r(x)$ as having its support in the interval $J$ such that

$$\int_0^1 \tau(x) r(x)\, dx = \int_J \tau(x) r(x)\, dx = M > 0$$

and let $V(x)$ denote the solution of

$$\begin{aligned} \partial_x v(x,t^*) \frac{dV}{dx} &= r(x), \qquad x \in J, \\ V(x_1) &= 0. \end{aligned}$$

In view of the previous lemma, the following function is well defined.

$$V(x) = \begin{cases} 0, & x \notin J \\ \int_{x_1}^{x} \frac{r(s)}{\partial_x v(s,t^*)} ds, & x \in J \end{cases}$$

In [**19, 20**] it is shown that the adjoint problem is approximately backward controllable, not just in the $L^2(U)$ topology, but in that of $C^1(U)$ as well. This implies that for any $\varepsilon > 0$, we can cause $w(x, 0; g^*)$ to satisfy

$$\begin{aligned} \sup_{x \in U} ||w(x,0;g^*) - V(x)||_{L^2(U)} &< \varepsilon, \\ \sup_{x \in U} \left|\left| \partial_x w(x,0;g^*) - \frac{dV}{dx} \right|\right|_{C^1(U)} &< \varepsilon, \end{aligned}$$

by an appropriate choice of smooth adjoint input $g^*(t)$ on the interval $(0, T)$. Then

$$\begin{aligned} & \int_U \tau(x)\, \partial_x v(x,t^*)\, \partial_x w(x,0)\, dx \\ = & \int_U \tau(x)\, \partial_x v(x,t^*) \left[ \partial_x w(x,0) - \frac{dV}{dx} + \frac{dV}{dx} \right] dx \\ = & \int_U \tau(x)\, \partial_x v(x,t^*) \left[ \partial_x w(x,0) - \frac{dV}{dx} \right] dx + \int_U \tau(x)\, \partial_x v(x,t^*) \frac{dV}{dx} dx \\ = & \int_U \tau(x)\, \partial_x v(x,t^*) \left[ \partial_x w(x,0) - \frac{dV}{dx} \right] dx + \int_J \tau(x)\, r(x)\, dx \\ \geq & M - K\varepsilon, \end{aligned}$$

where $K = ||\tau(x)\,\partial_x v(x,t^*)||_{L^\infty(U)}$. For $\varepsilon$ sufficiently small through the appropriate adjoint input $g^*$, this last expression can be forced to be positive, leading to the contradiction that implies $G[a] \neq G[a_0]$. □

We end this section by noting the increasing requirements on the terms in the initial boundary value problems. First, we ensure unique solutions by requiring the admissible coefficient (bounded away from zero) to be in $L^\infty(U)$ and the boundary terms $f_D(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$ and $f_N(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$. To guarantee identifiability of the coefficient via the observation operators noted, we must ensure the solution's gradient does not vanish on any measureable subset of the domain, rendering the coefficient values arbitrary on this set. This is accomplished by further controlling the boundary terms in some appropriate fashion, such as the sufficient conditions in Lemma 3. Finally, to achieve meaningful integrals in the identities developed in our adjoint methods, we have shown that the coefficients must have further regularity, i.e. must be elements of $H^s(U)$ for $s \geq 1$. In the recoveries that follow, we often relax this constraint, as much of the setting developed here proves overly restrictive. For example, it can be shown using semigroup theory that the solutions to our IBVPs are much smoother than indicated [7]. This translates to $\int_0^T \nabla w \cdot \nabla v dt$ being of greater regularity, which

in turn allows $\tau$ to be of lesser regularity and still recoverable by the developed techniques.

CHAPTER 4

# THE BACKUS-GILBERT METHOD

We now formulate a method to approximate $\tau$ from the overdetermined data on $\Gamma$. We select $N$ linearly independent functions for boundary value data in the adjoint problem. The choice for $N$ is arbitrary, and can be adjusted appropriately throughout the solution process. With this data in hand, we define the following, for convenience. (Note that we return to a generalized spatial dimension for this chapter).

$$K_j := \int_0^T \nabla w_j \cdot \nabla v dt$$

Denoting the scalars

$$\mu_j := \int_0^T \left[ \int_{\Gamma^D} B_D^* \triangle G_D + \int_{\Gamma^N} B_N^* \triangle G_N \right] dt$$

we can write the integral identities such as (3.8) in the general form

$$\mu_j = \langle \tau, K_j \rangle_{H^s \times H^{-s}} .$$

Letting $z_0 := (x_0, y_0)$ be an arbitrary fixed point in $U$, we employ the Backus-Gilbert ansatz,

$$\tau(z_0) = \sum_{j=1}^{N} \Lambda_j(z_0)\,\mu_j,$$

where $\Lambda_j(z_0)$ is assumed to be a scalar whose values vary with $z_0$ [**2, 8, 18**]. Note that

$$\begin{aligned}
\tau(z_0) &= \sum_{j=1}^{N} \Lambda_j(z_0)\,\mu_j \\
&= \sum_{j=1}^{N} \Lambda_j(z_0) \int_U \tau K_j dx \\
&= \sum_{j=1}^{N} \int_U \tau \Lambda_j(z_0)\, K_j dx \\
&= \int_U \tau \sum_{j=1}^{N} \Lambda_j(z_0)\, K_j dx.
\end{aligned}$$

Then

$$\begin{aligned}
\tau(z_0) &= \left\langle \tau(z), \sum_{j=1}^{N} \Lambda_j(z_0)\, K_j \right\rangle_{H^S \times H^{-S}} \\
&= \langle \tau(z), \delta(z - z_0) \rangle_{H^S \times H^{-S}},
\end{aligned}$$

where $\delta(z - z_0)$ denotes the Dirac delta distribution concentrated at $z = z_0$; hence

$$\sum_{j=1}^{N} \Lambda_j(z_0) K_j = \delta(z - z_0). \tag{4.1}$$

This indicates we need to express the delta distrubution in terms of the basis functions $K_j(x)$. To facilitate the discussion further, at this point we define the following mapping, with $\langle\ ,\ \rangle$ denoting an appropriate duality pairing.

$$\begin{aligned} A &: H^S(U) \to \mathbb{R}_N \\ Ax &= [\langle x, K_1\rangle, \ldots, \langle x, K_N\rangle] \end{aligned}$$

Here, $\mathbb{R}_N$ represents a row vector. We then formulate its transpose, $A^T : \mathbb{R}^N \to H^{-S}(U)$, with $b \in \mathbb{R}^N$ (a column vector) arbitrary.

$$\begin{aligned} (Ax, b)_{\mathbb{R}^N} &= ([\langle x, K_1\rangle, \ldots, \langle x, K_N\rangle], b)_{\mathbb{R}^N} \\ &= b_1 \langle x, K_1\rangle + \cdots + b_N \langle x, K_N\rangle \\ &= b_1 \int_U x K_1 dx + \cdots b_N \int_U x K_N dx \\ &= \int_U x \sum_{j=1}^{N} b_j K_j dx \\ &= \left\langle x, \sum_{j=1}^{N} b_j K_j \right\rangle_{H^S \times H^{-S}}. \end{aligned}$$

Then

$$A^T b = \sum_{j=1}^{N} b_j K_j.$$

Then in terms of the mapping $A$, we can rewrite (4.1) as

$$A^T \bar{\Lambda}(z_0) = \delta(z - z_0), \tag{4.2}$$

where $\bar{\Lambda} = [\Lambda_1, \ldots, \Lambda_N]^T$. Having no information about the existence or uniqueness of a solution to (4.2), we resort to an approximation of the unknown vector. As such, we determine the adjoint to the mapping $A^T$, $\left(A^T\right)^* : H^{-S}(U) \to \mathbb{R}^N$. Taking $x \in H^{-S}(U)$ arbitrary,

$$\begin{aligned}
\left(A^T b, x\right)_{-s} &= \left(\sum_{j=1}^{N} b_j K_j, x\right)_{-s} \\
&= (b_1 K_1 + \cdots + b_N K_N, x)_{-s} \\
&= (b_1 K_1, x)_{-s} + \cdots + (b_N K_N, x)_{-s} \\
&= b_1 (K_1, x)_{-s} + \cdots + b_N (K_N, x)_{-s} \\
&= \left(b, \left\{(K_1, x)_{-s}, \ldots, (K_N, x)_{-s}\right\}\right)_{\mathbb{R}^N}.
\end{aligned}$$

We see that

$$\left(A^T\right)^* x = \left[(K_1, x)_{-s}, \ldots, (K_N, x)_{-s}\right]^T.$$

Applying this adjoint to (4.2), we have the normal equation

$$\left(A^T\right)^* \left[A^T \bar{\Lambda}\left(z_0\right)\right] = \left(A^T\right)^* \left(\delta_{z_0}\right), \tag{4.3}$$

where $\delta_{z_0} = \delta\left(z - z_0\right)$. We now recharacterize each side of (4.3).

$$\begin{aligned}
\left(A^T\right)^* \left[A^T \bar{\Lambda}\right] &= \left(A^T\right)^* \left[\sum_{j=1}^{N} \Lambda_j K_j\right] \\
&= \left\{\left(K_1, \sum_{j=1}^{N} \Lambda_j K_j\right)_{-s}, \ldots, \left(K_N, \sum_{j=1}^{N} \Lambda_j K_j\right)_{-s}\right\} \\
&= \left\{\sum_{j=1}^{N} \Lambda_j \left(K_1, K_j\right)_{-s}, \ldots, \sum_{j=1}^{N} \Lambda_j \left(K_N, K_j\right)_{-s}\right\} \\
&= \left\{\sum_{j=1}^{N} \Lambda_j \left(K_i, K_j\right)_{-s}\right\}_{i=1}^{N}.
\end{aligned}$$

Then, with

$$\bar{K}_i = \left[\left(K_i, K_1\right)_{-s}, \ldots \left(K_i, K_N\right)_{-s}\right]^T \in \mathbb{R}^N,$$

we see that

$$\sum_{j=1}^{N} \Lambda_j \left(K_i, K_j\right)_{-s} = \left(\bar{\Lambda}, \bar{K}_i\right)_{\mathbb{R}^N}.$$

Then

$$\left(A^T\right)^* \left[A^T \bar{\Lambda}\right] = \left\{\left(\bar{\Lambda}, \bar{K}_i\right)_{\mathbb{R}^N}\right\}_{i=1}^{N}.$$

Similarly, the right side of (4.3) is rewritten as

$$\left(A^T\right)^* [\delta_{z_0}] = \left[(K_1, \delta_{z_0})_{-s}, \ldots, (K_N, \delta_{z_0})_{-s}\right]^T = \left\{(K_i, \delta_{z_0})_{-s}\right\}_{i=1}^N,$$

and the entire equation as

$$\left\{\left(\bar{\Lambda}, \bar{K}_i\right)_{\mathbb{R}^N}\right\}_{i=1}^N = \left\{(K_i, \delta_{z_0})_{-s}\right\}_{i=1}^N.$$

Furthermore, we note that

$$\left(\bar{\Lambda}, \bar{K}_i\right)_{\mathbb{R}^N} = \begin{bmatrix} (K_i, K_1)_{-s}, & \ldots & , (K_i, K_N)_{-s} \end{bmatrix} \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_N \end{bmatrix},$$

and so

$$\left\{\left(\bar{\Lambda}, \bar{K}_i\right)_{\mathbb{R}^N}\right\}_{i=1}^N = \begin{bmatrix} (K_1, K_1)_{-s} & \cdots & (K_1, K_N)_{-s} \\ \vdots & \ddots & \vdots \\ (K_N, K_1)_{-s} & \cdots & (K_N, K_N)_{-s} \end{bmatrix} \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_N \end{bmatrix}.$$

Thus we can rewrite (4.3) in matrix form,

$$\begin{bmatrix} (K_1, K_1)_{-s} & \cdots & (K_1, K_N)_{-s} \\ \vdots & \ddots & \vdots \\ (K_N, K_1)_{-s} & \cdots & (K_N, K_N)_{-s} \end{bmatrix} \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_N \end{bmatrix} = \begin{bmatrix} (K_1, \delta_{z_0})_{-s} \\ \vdots \\ (K_N, \delta_{z_0})_{-s} \end{bmatrix}, \tag{4.4}$$

or

$$K\bar{\Lambda} = \left\{(K_i, \delta_{z_0})_{-s}\right\}_{i=1}^{N}.$$

Note that if the functions $K_j$ are linearly independent, the matrix $K$ will have rank $N$ and thus be uniquely solveable. We now establish criteria for this linear independence.

**Theorem 5.** *Suppose the adjoint inputs $g_j^*(t)$ are linearly independent on $[0, T^*]$ for $T^* \geq T$. Suppose also that the inputs to the forward problem and the inputs to the adjoint problem are such that the gradients $\nabla v$ and $\nabla w_j$ vanish on no positive measure subset of $(0,1) \times (0,T)$ and each of these gradients is of one sign on this domain. Then the functions $K_j(x)$ are linearly independent on $[0,1]$.*

**Proof.** Suppose

$$\sum_{j=1}^{N} c_j K_j(x) = 0 \quad \forall x \in [0,1],$$

i.e.

$$\sum_{j=1}^{N} c_j \int_0^T \nabla w_j \cdot \nabla v dt = 0,$$

or

$$\int_0^T \nabla v \cdot \sum_{j=1}^{N} c_j \nabla w_j dt = 0.$$

Since each of the gradients is of one sign, the vanishing of the integral implies the vanishing of the continuous integrand at each $t \in (0,T)$. Then

$$\sum_{j=1}^{N} c_j \nabla w_j = 0 \quad \forall x \in [0,1] \quad \forall t \in (0,T).$$

So

$$\nabla \left[ \sum_{j=1}^{N} c_j w_j \right] = \nabla \left[ \sum_{j=1}^{N} w\left(x,t,c_j g_j^*\right) \right] = 0,$$

due to the linear dependence of the adjoint solution on its boundary conditions. Rewritten as

$$\nabla \left[ w\left( x,t,\sum_{j=1}^{N} c_j g_j^* \right) \right] = 0,$$

we have the implication that

$$w\left( x,t,\sum_{j=1}^{N} c_j g_j^* \right) = k \in \mathbb{R}$$

with respect to the spatial domain. Moreover, because $-\partial_t w = \nabla \cdot (a \nabla w)$, we have that $\partial_t w = 0$; thus the implication extends to the time domain as well. Now, since

$$w\left(x, T, \sum_{j=1}^{N} c_j g_j^*\right) = 0,$$

we have that $k = 0$. Then, via maximum/minimum principles [7], we have that $\sum_{j=1}^{N} c_j g_j^* = 0$. Because the $g_j^*$ are chosen linearly independent,

$$c_1 = \cdots = c_N = 0,$$

and we have the desired result. □

**Theorem 6.** *If the functions $K_j(x)$ are linearly independent, then the matrix*

$$\begin{bmatrix} (K_1, K_1)_{-s} & \cdots & (K_1, K_N)_{-s} \\ \vdots & \ddots & \vdots \\ (K_N, K_1)_{-s} & \cdots & (K_N, K_N)_{-s} \end{bmatrix}$$

*is positive definite.*

**Proof.** Suppose that the functions $K_j(x)$ are linearly independent and

$$\sum_{j=1}^{N} c_j K_j(x) \not\equiv 0 \text{ on } U.$$

Then $\exists c_j \neq 0$ and $\left(\sum_{j=1}^{N} c_j K_j(x), \sum_{j=1}^{N} c_j K_j(x)\right)_{-s} > 0$. This can be rewritten as

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \begin{bmatrix} (K_1, K_1)_{-s} & \cdots & (K_1, K_N)_{-s} \\ \vdots & \ddots & \vdots \\ (K_N, K_1)_{-s} & \cdots & (K_N, K_N)_{-s} \end{bmatrix} \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix} > 0,$$

which gives the desired result. □

We then note our approximation of $\tau(z_0)$ becomes a matter of solving (4.4), where the matrix $K$ is independent of $z_0$. This latter point is advantageous, since for any choice of $N$, we need only recalculate $\left\{(K_i, \delta_{z_0})_{-s}\right\}_{i=1}^{N}$ for various points $z_0 \in U$. This greatly simplifies refining the resolution of the approximation.

CHAPTER 5

# DIRECT IDENTIFICATION

As a preliminary test to see if the coefficients are recoverable in any sense, we work directly with the integral identities, without the Backus-Gilbert method, in one spatial dimension. This is only possible in the simplest case of recovering a constant coefficient. We perform identifications with each choice of imposed and observed boundary conditions (from Section 3.4) and compare results. First, we note an important point that will affect all subsequent computations. In the adjoint method previously detailed, our adjoint constructions in every case utilized the unknown coefficient $a(x)$. For example, the adjoint problem (3.11), under mixed boundary conditions, reads

$$\begin{aligned} -\partial_t w - \partial_x (a \partial_x w) &= 0, \\ w(x,T) &= 0, \\ w(0,t) &= g^*(t), \\ a(1)\, \partial_x w(1,t) &= h^*(t). \end{aligned}$$

We annotate the solution to this problem as $w = w(a)$, and realize since $a$ is unknown, our use of $w$ in the method is impossible. Instead we must use a estimated adjoint solution, $\hat{w} = \hat{w}(\hat{a})$, where $\hat{a}$ is some "sensible" function. We'll often choose $\hat{a} = a_0$, out of convenience.

With unspecified boundary conditions, we have the identity

$$\int_0^T B_0^* \Delta G_0 dt + \int_0^T B_1^* \Delta G_1 dt = \int \int_{U_T} \tau \partial_x w \partial_x v dx dt.$$

With $\tau(x) = a(x) - a_0 = \text{constant}$,

$$\int \int_{U_T} \tau(x) \partial_x w \partial_x v dx dt = \tau \int \int_{U_T} \partial_x w \partial_x v dx dt$$

and

$$\tau = \frac{\int_0^T B_0^* \Delta G_0 dt + \int_0^T B_1^* \Delta G_1 dt}{\int \int_{U_T} \partial_x w \partial_x v dx dt}.$$

Note that no indexing is necessary, i.e. $N = 1$ implies $\mu$ and $K(x)$ are a single value and function, respectively. To correlate with our previous development, we have

$$\mu = \int_0^T B_0^* \Delta G_0 dt + \int_0^T B_1^* \Delta G_1 dt,$$
$$K(x) = \int_0^T \partial_x w \partial_x v dt,$$

Table 5.1. Recovery values for $\hat{w} = \hat{w}(a_0)$

| Boundary conditions | $\tau$ |
|---|---|
| Mixed | 0.14792253143140 |
| Dirichlet | 0.40561141333667 |
| Neumann | 0.18237172075535 |

and so

$$\tau = \frac{\mu}{\int_U K(x)\,dx}.$$

As an example, we solve for $\tau$ under specified conditions, using

$$a = .6 \quad \text{(unknown coefficient)}$$

$$a_0 = .2 \quad \text{(comparison coefficient)}$$

$$g^*(t) = 1 - t \quad \text{(adjoint input, } x = 0\text{)}$$

$$h^*(t) = \sin(\pi t) \quad \text{(adjoint input, } x = 1\text{)}.$$

With $\hat{w} = \hat{w}(a_0)$, we compute the values in Table 5.1, noting the true value $\tau = .4$.

All the recoveries are the proper order of magnitude, but contain significant error. We see the Dirichlet result is the most accurate of the three cases. This was repeated with several different values for $a$ and $a_0$ with similar results. Of course, the errors associated with the recovery can be logically attributed to the use of $a_0$ in place of $a$ when computing the adjoint solution. Evidently, this has less effect with Dirichlet boundary conditions, since the approximate coefficient is not present in the adjoint boundary conditions. To see this, we look at the adjoint

boundary conditions , which mirror those of the direct problem. Here we view the mixed case:

$$\begin{aligned} \hat{w}(0,t) &= g^*(t), \\ a_0 \partial_x \hat{w}(1,t) &= h^*(t). \end{aligned}$$

Since we control $g^*$ and $h^*$, independent of the cases $w = w(a)$ or $\hat{w} = \hat{w}(a_0)$, the value $\mu$ is unaffected. However, we note that at the portion of the boundary $x = 1$,

$$\partial_x \hat{w}(1,t) = \frac{h^*(t)}{a_0}$$

contains error on the entire time domain. These gradient values are very small over most of the interior of $U$, and at $x = 1$ the value is incorrect. This translates to a significant error in the value of $K(x)$, since its computation is based in part on $\partial_x \hat{w}$. On the other hand, $\hat{w}(0,t) = g^*(t) = w(0,t)$.

Of course, using Dirichlet conditions, we have that

$$\begin{aligned} \hat{w}(0,t) &= g^*(t) = w(0,t), \\ \hat{w}(1,t) &= h^*(t) = w(1,t). \end{aligned}$$

Table 5.2. Recovery values for $w = w(a)$

| Boundary conditions | $\tau$ |
|---|---|
| Mixed | 0.39944372886666 |
| Dirichlet | 0.39989962262540 |
| Neumann | 0.39948707669281 |

Despite the fact that $\hat{w}$ exhibits error on the interior in all cases, the boundary values remain true in the Dirichlet case, enhancing the recovery for the closest approximation.

Of course, with Neumann conditions, we have error at both boundaries.

$$\begin{aligned} \partial_x \hat{w}(0,t) &= \frac{g^*(t)}{a_0} \\ \partial_x \hat{w}(1,t) &= \frac{h^*(t)}{a_0} \end{aligned}$$

Compare the identifications using $\hat{w} = \hat{w}(a_0)$ (Table 5.1) with those using $w = w(a)$ in Table 5.2.

Of course, numerical error with finite differences and approximate integration are still present, but as expected, we see significantly improved results.

Clearly, using $a_0$ to approximate the adjoint solution induces error. However, a series of recoveries based on $\hat{w}(a_0)$ as $a_0$ tends to the true value $a$, shows that the error in the approximation to $\tau$ also tends continuously to zero. We use the Dirichlet case only, and as above, fix $a = .6$.

Table 5.3. A series of recoveries

| $a_0$ | $\tau$ | actual | % relative error |
|---|---|---|---|
| .1 | .5 | .378483 | 24.30 |
| .2 | .4 | .365734 | 8.56 |
| .3 | .3 | .289819 | 3.39 |
| .4 | .2 | .197264 | 1.37 |
| .5 | .1 | .099551 | .45 |
| .6 | 0 | 0 | 0 |
| .7 | −.1 | −.100257 | .26 |
| .8 | −.2 | −.200785 | .39 |
| .9 | −.3 | −.301400 | .47 |
| 1 | −.4 | −.402026 | .51 |
| 1.1 | −.5 | −.502628 | .53 |

Of course, we get zero error when $a = a_0$, since the observed and computed data on the boundary is the same. In reality, error in observed data will introduce noise into the computation, and we'll get a nonzero result. As expected, we see that the relative error approaches zero as $a_0$ approaches the true coefficient value. Similar results were noted for the other cases. Interestingly, we note that for larger values of $a_0$, the error appears to be growing at a notably smaller rate. A possible explanation for this is mentioned in Chapter 9.

We now view the effect of using observations on only part of the boundary. In Table 5.4, we compare the above recoveries with both $g^*(t)$ and $h^*(t)$ nonzero, with the cases where one or the other is chosen zero. Again, we use $a = .6$ and $a_0 = .2$. Except for the mixed case with observations at $x = 0$ only, we see little effect of the reduction of data collected. In the mixed case, with $h^* = 0$, we have

Table 5.4. Recoveries under varying observations, $\hat{w} = \hat{w}(a_0)$

| Conditions | $\tau:\ h^* = 0$ | $\tau:\ g^* = 0$ | $\tau:\ h^*, g^* \neq 0$ |
|---|---|---|---|
| Mixed | 0.42701661442311 | 0.17508583153271 | 0.14792253143140 |
| Dirichlet | 0.36573369675596 | 0.38283267212683 | 0.40561141333667 |
| Neumann | 0.18160739891618 | 0.18281861892377 | 0.18237172075535 |

Table 5.5. Recoveries under varying observations, $w = w(a)$

| Conditions | $\tau:\ h^* = 0$ | $\tau:\ g^* = 0$ | $\tau:\ h^*, g^* \neq 0$ |
|---|---|---|---|
| Mixed | 0.39996772622512 | 0.39956798576402 | 0.39944372886666 |
| Dirichlet | 0.40013075454499 | 0.40002571863012 | 0.39989962262540 |
| Neumann | 0.39919310442742 | 0.39965802203946 | 0.39948707669281 |

that

$$\partial_x \hat{w}(1,t) = \frac{h^*(t)}{a_0} = 0,$$

ensuring that $\partial_x \hat{w}(1,t) = \partial_x w(1,t)$; i.e. there is no longer error at the boundaries, which evidently improves the recovery. This is the only one of the three cases where we can expect this improvement. Indeed, in the remaining situations, we see no significant change. We'll view this further when employing the Backus-Gilbert method outlined in Chapter 4.

Finally, we collect results using $w = w(a)$, and note excellent results in all cases, as seen in Table 5.5.

Although we bypass the Backus-Gilbert method with these direct identifications, we obtain valuable information: Clearly, data from observations on the boundary (or a portion thereof) is sufficient to recover an unknown coefficient,

albeit a simple constant. Knowing this provides the confidence to use the data in a more involved manner, with non-constant coefficients.

CHAPTER 6

# COMPUTATIONAL IMPLEMENTATION

## 6.1. Overview

We now return to the proposed Backus-Gilbert method, which requires the generation of observation data. To do so, we solve the forward problem with a known coefficient, via finite differences. Generating this data ($G_0$ and $G_1$) takes the place of the physical experiment. Once this is accomplished, we proceed to the recovery phase. After the constant coefficient $a_0$ is chosen, we solve the comparison and adjoint problems, also using finite differences. These methods are detailed below. The numerical PDE solutions achieved are stored in matrices, as are the chosen adjoint inputs $B_0^*[w_j]$ and $B_1^*[w_j]$. The corresponding data are then used to construct the integrands defining the constants

$$\mu_j = \int_0^T \left(B_0^* \triangle G_0 + B_1^* \triangle G_1\right) dt,$$

and the functions

$$K_j(x) = \int_0^T \partial_x w_j \partial_x v dt,$$

for $j = 1 \ldots N$. Trapezoidal integration is used for these computations. We then employ the first $d$ eigenfunctions $\{\phi_n\}$, which are associated with the one dimensional Laplacian operator,

$$-\phi_n''(x) = \lambda \phi_n(x)$$

and specificed homogeneous boundary conditions. These are numerically defined for use in determining the Fourier coefficients of the functions $K_j(x)$,

$$(K_j)_n = \int_U K_j(x)\, \phi_n(x)\, dx,$$

again via trapezoidal integration. These coefficients, along with the eigenvalues $\{\lambda_n\}_{n=1}^d$, are then used to numerically approximate the $H^{-s}(U)$ inner products

$$(K_i, K_j)_{-s} \approx \sum_{n=1}^{d} \lambda_n^{-s} (K_i)_n (K_j)_n,$$

for $i, j = 1 \ldots N$. These, in turn, comprise the elements of the matrix $K$. Similarly, we derive the Fourier coefficients for the functions $\delta_{x_0}$, although much simpler computationally, since

$$(\delta_{x_0})_n = \int_U \delta_{x_0} \phi_n(x)\, dx = \phi_n(x_0)$$

indicates we merely need to call the $n^{th}$ eigenfunction at the desired point in the domain. These coefficients are used to approximate the inner products

$$(K_j, \delta_{x_0})_{-s} \approx \sum_{n=1}^{d} \lambda_n^{-s} (K_j)_n (\delta_{x_0})_n = \sum_{n=1}^{d} \lambda_n^{-s} (K_j)_n \phi_n (x_0)$$

which comprise the elements of the vector

$$\delta (x_0) = \begin{bmatrix} (K_1, \delta_{x_0})_{-s} \\ \vdots \\ (K_N, \delta_{x_0})_{-s} \end{bmatrix}.$$

Now, because the matrix $K$ is symmetric positive definite with eigenvalues $0 < \alpha_1 < \cdots < \alpha_N$, we exploit the orthogonality of the matrix eigenvectors $\{v_j\}$ to solve the system $K\Lambda (x_0) = \delta (x_0)$. Accordingly, we compute

$$\Lambda (x_0) = \sum_{j=1}^{N} \frac{v_j^T \delta (x_0)}{\lambda_j v_j^T v_j} v_j.$$

then approximate $\tau (x_0) \approx \Lambda (x_0) \cdot \mu$. The corresponding Matlab code for these computations are detailed in Appendix 6.

## 6.2. Finite Differences

### 6.2.1. Numerical Solution of the Direct Problem

We use a simple, explicit finite differences scheme to solve the direct problem. Specifically, we approximate the time derivative with a forward difference and interior spatial derivatives with centered differences.

$$\begin{aligned} \frac{\partial u}{\partial t} &\approx \frac{u(x_0, t_0 + \Delta t) - u(x_0, t)}{\Delta t}, \\ \frac{\partial}{\partial x}\left[a(x)\,\partial_x u(x,t)\right] &\approx \frac{a\left(x + \frac{\Delta x}{2}\right)\left[u(x_0 + \Delta x, t_0) - u(x_0, t_0)\right]}{\Delta x^2} \\ &\quad - \frac{a\left(x - \frac{\Delta x}{2}\right)\left[u(x_0, t_0) - u(x_0 - \Delta x, t_0)\right]}{\Delta x^2}. \end{aligned}$$

Mixed and Neumann boundary conditions utilize ghost points and either backward or forward differences, as appropriate. We select a discretization of the spatial domain. Then, due to the presence of the terms

$$a\left(x + \frac{\Delta x}{2}\right), a\left(x - \frac{\Delta x}{2}\right)$$

we numerically define $a(x)$ more finely than the rest of the spatial data. Then, indexing time with subscript $m$ and space with superscript $n$, the approximate PDE is

$$\frac{u_{m+1}^n - u_m^n}{\Delta t} \approx \frac{a_{n+\frac{1}{2}}\left[u_m^{n+1} - u_m^n\right] - a_{n-\frac{1}{2}}\left[u_m^n - u_m^{n-1}\right]}{\Delta x^2}.$$

Solving for $u_{m+1}^n$, we have the explicit algorithm

$$u_{m+1}^n \approx \frac{\Delta t}{\Delta x^2}\left(a_{n+\frac{1}{2}}\left[u_m^{n+1}-u_m^n\right]-a_{n-\frac{1}{2}}\left[u_m^n-u_m^{n-1}\right]\right)+u_m^n.$$

Of course, this algorithm is unstable for large timesteps, which we quantify here. Setting $A := \max_{x\in U}\{a(x)\}$, established theory requires that $0 < \frac{A\Delta t}{\Delta x^2} \leq \frac{1}{2}$ to ensure stability [**23**]. Thus, after selecting $\Delta x$, we compute the coarsest time mesh possible, $\Delta t = \frac{\Delta x^2}{2A}$. Note this changes with the coefficient to be recovered. Even though this is unknown, this presents no conflict, since this algorithm is merely for data generation.

Of course, the comparison problem uses the same algorithm, but the constant coefficient allows simplification. We note here that we intentionally use a different spatial grid for the direct problem than in any other portion of the recovery process – this is to add some reality to the process of data generation for our theoretical work. (The different spatial grids correspond to different time grids as well, due to stability issues mentioned above.) As such, we initially produce observation values via the direct solution, which are then interpolated or extrapolated (both linearly) to the recovery spatial grid size. The new values are then used to compute $\triangle G_0$, $\triangle G_1$, and subsequently the scalars $\mu_j$.

As a check to this numerical scheme, let's visually compare the results with an analytical solution, the latter achieved using eigenfunction expansion on the comparison problem:

$$\begin{aligned} \partial_t v(x,t) &= a_0 \partial_{xx} v(x,t) \qquad (6.1) \\ v(x,0) &= \partial_x v(1,t) = 0 \\ v(0,t) &= t, \end{aligned}$$

where $a_0 = 1$. The simple nature of this problem makes finding an analytical solution a relatively easy matter. Expanding in terms of the eigenpairs

$$\begin{aligned} \phi_n(x) &= \sqrt{2}\sin\left[(n-1/2)\pi x\right], \qquad (6.2) \\ \lambda_n &= (n-1/2)^2 \pi^2, \end{aligned}$$

for $n \in \mathbb{N}$, we compute

$$v(x,t) = \begin{cases} t, & x = 0 \\ \sum_{n=1}^{\infty} c_n \lambda_n \int_0^t e^{-\lambda_n (t-s)} s ds \phi_n(x), & x \neq 0 \end{cases}.$$

(See Appendix 5.1 for details of this derivation.) We use Maple to plot several profiles in Figure 6.1, and compare with our numerical Matlab results in Figure

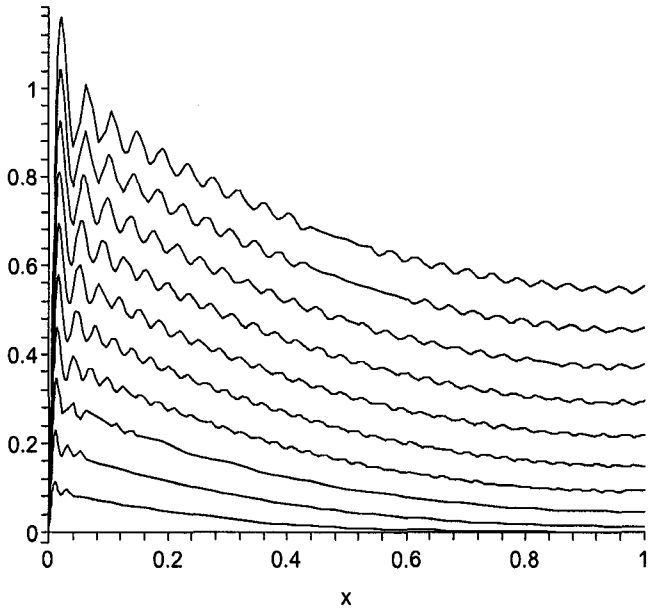


Figure 6.1. Comparison solution using eigenfunction expansion

6.2. These latter finite difference profiles were generated with $\Delta t = \frac{1}{1800}$ and $\Delta x = \frac{1}{30}$. Of course, we note the oscillatory Gibbs phenomena and the fact that there is a disconnect at $x = 0$ due to the eigenfunction expansion technique.

### 6.2.2. Numerical Solution of the Adjoint Problem

The algorithm construction for the adjoint problem is similar to the forward problem, and again uses interior spatial centered differences. However, due to the backwards parabolic nature of this PDE, a backwards difference was used for the

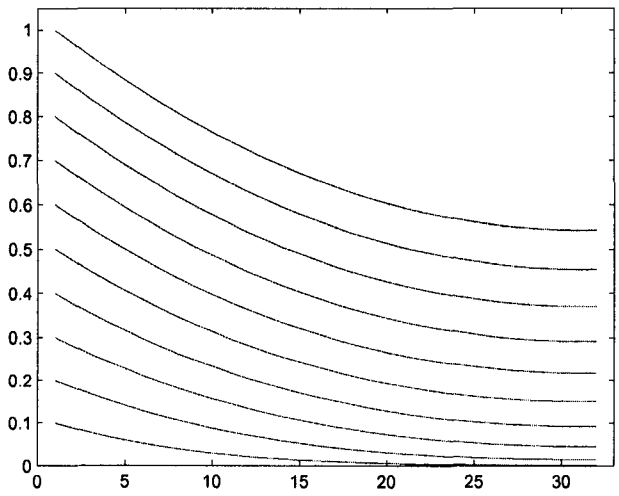


Figure 6.2. Comparison solution using finite differences

time derivative:

$$-\frac{\partial \hat{w}}{\partial t} \approx -\left[\frac{\hat{w}\left(x_0, t_0\right) - \hat{w}\left(x_0, t - \Delta t\right)}{\Delta t}\right].$$

Then the approximate PDE is

$$\frac{\hat{w}_{m-1}^n - \hat{w}_m^n}{\Delta t} \approx \frac{\hat{a}_{n+\frac{1}{2}}\left[\hat{w}_m^{n+1} - \hat{w}_m^n\right] - \hat{a}_{n-\frac{1}{2}}\left[\hat{w}_m^n - \hat{w}_m^{n-1}\right]}{\Delta x^2}.$$

Solving for $\hat{w}_{m-1}^n$, we have the explicit approximation

$$\hat{w}_{m-1}^n \approx \frac{\Delta t}{\Delta x^2}\left(\hat{a}_{n+\frac{1}{2}}\left[\hat{w}_m^{n+1} - \hat{w}_m^n\right] - \hat{a}_{n-\frac{1}{2}}\left[\hat{w}_m^n - \hat{w}_m^{n-1}\right]\right) + \hat{w}_m^n,$$

with the same stability issues mentioned above.

We check to this numerical scheme with the following problem, visually comparing the results with an analytical solution, the latter achieved using eigenfunction expansion:

$$-\partial_t w(x,t) = a_0 \partial_{xx} w(x,t) \tag{6.3}$$

$$\begin{aligned} w(x,1) &= \partial_x w(1,t) = 0 \\ w(0,t) &= t-1, \end{aligned}$$

with $a_0 = 1$. Expanding in terms of the eigenpairs (6.2), we compute

$$w(x,t) = \begin{cases} t-1, & x = 0 \\ \sum_{n=1}^{\infty} c_n \lambda_n \int_t^1 e^{-\lambda_n (s-t)} (s-1) ds \phi_n(x), & x \neq 0 \end{cases}.$$

(See Appendix 5.2 for details of this derivation.) Using Maple, we plot several profiles, as seen in Figure 6.3. The corresponding numerical profiles, again with time/space steps $\Delta t = \frac{1}{1800}$ and $\Delta x = \frac{1}{30}$, are shown in Figure 6.4.

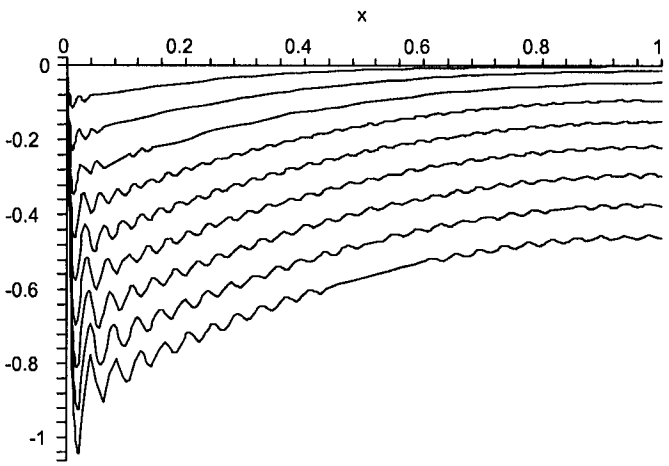


Figure 6.3. Adjoint solution using eigenfunction expansion

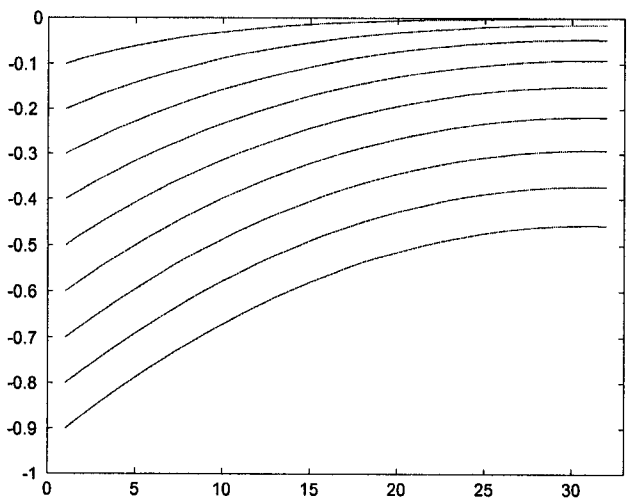


Figure 6.4. Adjoint solution using finite differences

CHAPTER 7

# PRELIMINARY RESULTS

## 7.1. General Results

As mentioned in the direct identification process (Chapter 5), the method described requires the solution of an adjoint problem whose coefficient is the unknown coefficient we are attempting to recover. Since this is clearly impossible in any real recovery algorithm, we must consider the effect of using an approximate coefficient in this problem. For now, however, the artificial recoveries in this section are based on using the true coefficient in the adjoint problem. As such, we generate the true adjoint solution $w = w(a)$ versus some approximation: $\hat{w} = \hat{w}(a_0)$, for example. We start with a constant coefficient recovery, and view the results for $N = 1, \ldots, 10$. Additionally, we use the following adjoint inputs.

$$g_j^*(t) = \begin{cases} 1, & 0 \leq t \leq \frac{j-1}{N} \\ \frac{N(t-1)}{j-N-1}, & \frac{j-1}{N} < t \leq 1 \end{cases}$$

$$h_j^*(t) = \sin(j\pi t)$$

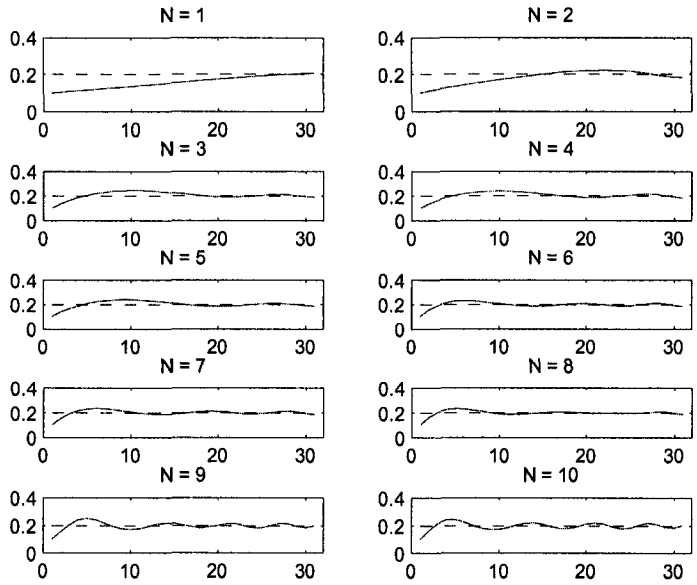


Figure 7.1. Constant recovery with mixed conditions

With $a = .2$, $a_0 = .1$ and mixed boundary conditions, we achieve the results shown in Figure 7.1, notably most accurate with subspace dimension $N = 8$, and less accurate thereafter as oscillations increase. (In this and all subsequent figures, the horizontal axes represent the appropriate independent variable – in this case, the spatial variable $x$.) Similar results were obtained for Dirichlet and Neumann boundary conditions.

Interestingly, with the non-constant linear function $a(x) = .5x + .1$, we note a significantly lower subspace dimension is necessary for a successful recovery. This

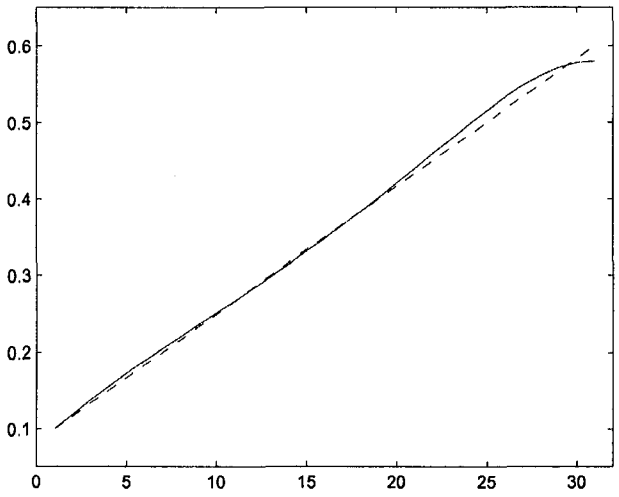


Figure 7.2. Linear recovery, Dirichlet conditions, $N = 3$

time we use $a_0 = a(0)$ (for reasons detailed in Section 7.2). With $N = 3$ and Dirichlet boundary conditions, we have the recovery shown in Figure 7.2.

Using the same adjoint inputs, we next recover the piecewise constant

$$a(x) = \begin{cases} .1, & 0 \leq t \leq \frac{1}{2} \\ .3, & \frac{1}{2} < t \leq 1 \end{cases}.$$

Figure 7.3 shows the results using Dirichlet boundary conditions – note that with $N = 1$ and 2, the recoveries are nearly linear. As above, recoveries exhibit increased oscillations with higher subspace dimensions. In this case, the recovered coefficient associated with $N = 7$ differed the least from the true coefficient in the $L^2(0,1)$ norm. With $N > 7$, oscillations increased and the results degraded

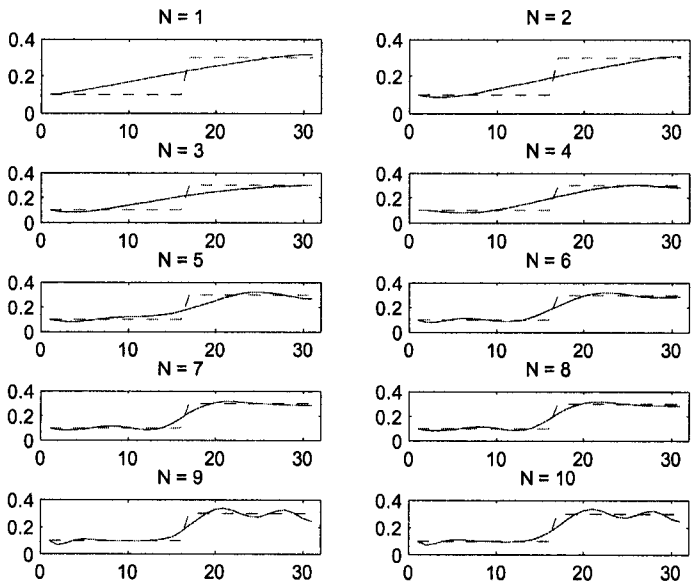


Figure 7.3. Piecewise constant "ramp" recovery, Dirichlet conditions

slightly, more notable when $N > 10$ (not shown). We obtained similar results with mixed and Neumann conditions.

We now keep the piecewise constant nature of the coefficient, but make it a non-monotonic function.

$$a(x) = \begin{cases} .1, & 0 \leq t \leq \frac{1}{3} \\ .3, & \frac{1}{3} < t < \frac{2}{3} \\ .1, & \frac{2}{3} \leq t \leq 1 \end{cases} .$$

Imposing Neumann conditions this time, we note the recoveries in Figure 7.4. It is interesting to see that the recovery at $x = 0$ and $x = 1$ exhibits little error, even

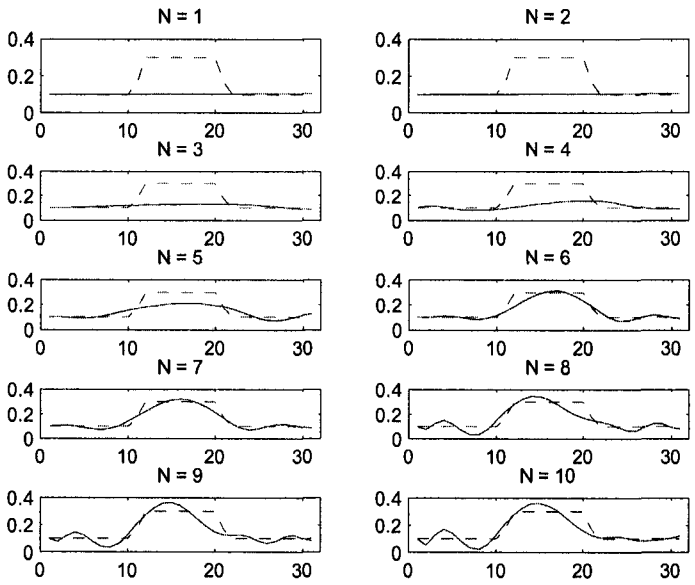


Figure 7.4. Piecewise constant "step" recovery, Neumann conditions

with lower subspace dimensions. This time, however, the near linear nature of the low subspace recoveries promote worse results on the interior. Higher dimensions are required to allow interior accuracy, which is optimal this time with $N = 7$. Mixed and Dirichlet conditions produced very similar results.

We next attempt to recover a similar coefficient, without the symmetry noted above. We offset the maximum region, and make it "thinner" as well.

$$a(x) = \begin{cases} .1, & 0 \leq t \leq \frac{1}{5} \\ .3 & \frac{1}{5} < t < \frac{2}{5} \\ .1, & \frac{2}{5} \leq t \leq 1 \end{cases} .$$

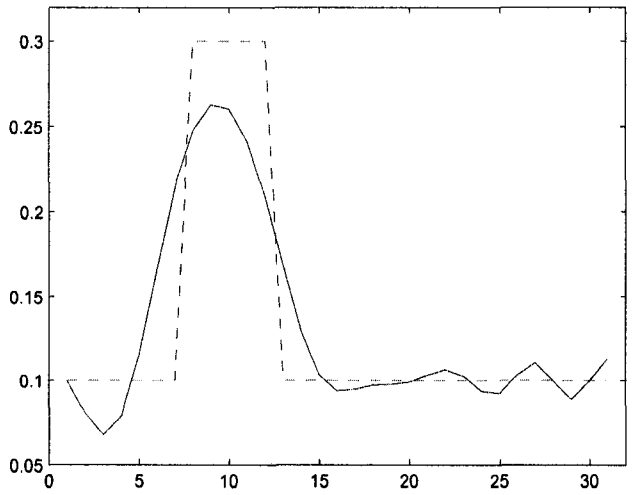


Figure 7.5. A thinner, asymmetric coefficient, $N = 11$

We note that even with a larger subspace dimension, $N = 11$, the recovery method has difficulty with the maximum value on the interior. The recovery shown in Figure 7.5 was performed with Dirichlet boundary conditions, although mixed and Neumann led to similar results once again. We attempt a similar recovery, this time with an even "thinner" maximum.

$$a(x) = \begin{cases} .1, & 0 \leq t \leq \frac{1}{5} \\ .3, & \frac{1}{5} < t < \frac{3}{10} \\ .1, & \frac{3}{10} \leq t \leq 1 \end{cases} .$$

As expected, the recovery – shown in Figure 7.6 – has even more difficulty ($N = 9$, Dirichlet conditions shown).

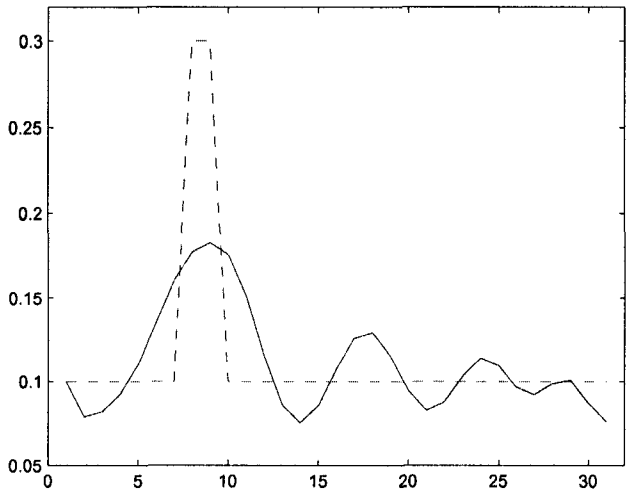


Figure 7.6. An even thinner step

We try two additional piecewise constant coefficients, $a_1$ and $a_2$, to further gauge the effectiveness of the recovery method.

$$a_1(x) = \begin{cases} .1, & 0 \le t \le \frac{1}{3} \\ .2 & \frac{1}{3} < t < \frac{2}{3} \\ .5, & \frac{2}{3} \le t \le 1 \end{cases}, \qquad a_2(x) = \begin{cases} .1, & 0 \le t \le \frac{1}{3} \\ .5 & \frac{1}{3} < t < \frac{2}{3} \\ .3, & \frac{2}{3} \le t \le 1 \end{cases}$$

The success is obvious in both cases, and much improved over the two previous recoveries. The best recovery of $a_1$ was accomplished with Dirichlet conditions and $N = 8$ (Figure 7.7), while the best for $a_2$ was with Neumann conditions and $N = 7$ (Figure 7.8). The other boundary conditions produced accurate results as well, and not notably different.

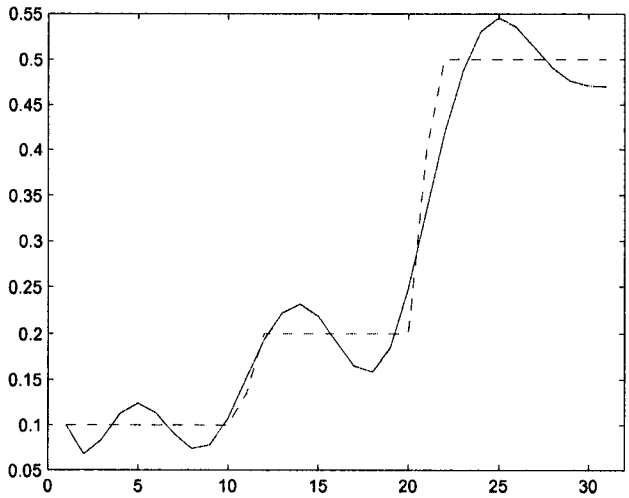


Figure 7.7. A recovery of $a_1$

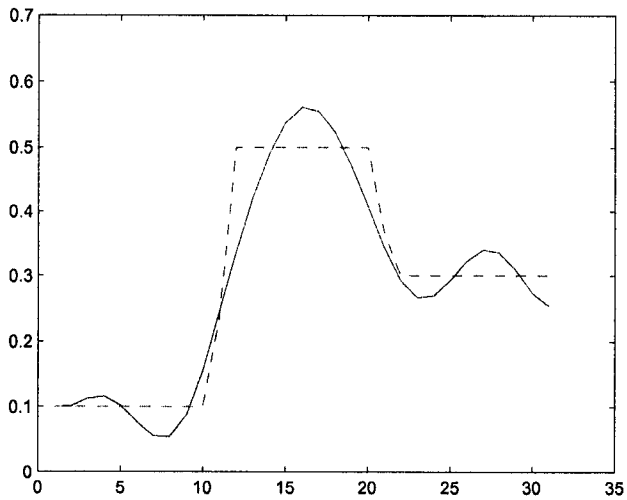


Figure 7.8. A recovery of $a_2$

Clearly the method detailed in Chapter 4 works, albeit in this artificial setting using the true coefficient to recover itself. We've noted the effect of the subspace dimensions on the recovery. Additionally, it seems that the choice of boundary conditions contributes little to the success of the recovery – most of the above results held similarly under the various boundary conditions. Before using an approximate coefficient in the adjoint problem, we continue with this controlled, artificial recovery to note the effects of further changes to the process. Specifically, we change the eigenfunctions, adjoint inputs, the Hilbert space dimension, and grid settings.

## 7.2. Eigenfunction Analysis

In each recovery above, we used, as a basis for $H^s(U)$, the orthonormal set of eigenfunctions given by $\phi_n(x) = \sqrt{2}\sin\left[\left(n-\frac{1}{2}\right)\pi x\right]$, generated by the boundary value problem

$$\begin{aligned} -\phi''(x) &= \lambda\phi(x), \qquad (7.1) \\ \phi(0) &= \phi'(1) = 0. \end{aligned}$$

We now look at the following alternate sets of eigenfunctions, generated by the changed boundary conditions in (7.1).

$$\begin{aligned}
\phi'(0) &= \phi(1) = 0 \Rightarrow \phi_n(x) = \sqrt{2}\cos\left[\left(n - \frac{1}{2}\right)\pi x\right] \\
\phi(0) &= \phi(1) = 0 \Rightarrow \phi_n(x) = \sqrt{2}\sin(n\pi x) \\
\phi'(0) &= \phi'(1) = 0 \Rightarrow \phi_n(x) = \sqrt{2}\cos(n\pi x)
\end{aligned}$$

Note the effect of these functions in the recovery process: Recall from Chapter 2 that the $H^{-s}(U)$ inner product of elements $\gamma, \eta$ is given by the sum

$$(\gamma, \eta)_{-s} := \sum_{n=1}^{\infty} (1 + \lambda_n)^{-s} \gamma_n \eta_n,$$

with the coefficients given by

$$\gamma_n = \langle \gamma, \phi_n \rangle_{D'(U) \times D(U)} = \int_U \gamma \phi_n(x)\, dx.$$

Then specifically, $\delta(x_0) \in H^{-s}(U)$ has coefficients

$$[\delta(x_0)]_n = \int_U \delta(x_0)\, \phi_n(x)\, dx = \phi_n(x_0),$$

which adopt the characteristics of the eigenfunctions on the boundary $x_0 = 0$ and $x_0 = 1$. Then

$$(K_i, \delta(x_0))_{-s} = \sum_{n=1}^{\infty} (1+\lambda_n)_n^{-s} (K_i)_n \phi_n(x_0),$$

which comprise the vector

$$\delta(x_0) = \begin{bmatrix} (K_1, \delta_{x_0})_{-s} \\ \vdots \\ (K_N, \delta_{x_0})_{-s} \end{bmatrix}$$

in the $N$ dimensional symmetric linear system $K\Lambda(x_0) = \delta(x_0)$. Considering the mixed conditions leading to the eigenfunctions $\phi_n(x) = \sqrt{2}\sin\left[\left(n - \frac{1}{2}\right)\pi x\right]$ used in the recoveries in Section 7.1, we see that $K\Lambda(0) = \delta(0) = 0$ gives $\Lambda(0) = 0$, since the nullspace of $K$ is trivial. Thus

$$\tau(0) = \sum_{j=1}^{N} \Lambda(0)\mu_j = 0.$$

Then, by assuming knowledge of $a(0)$ and setting $a_0$ equal to this value, we achieve an exact recovery at $x = 0$: $a(0) = a_0 + \tau(0)$.

Similarly, approximating $\phi'(1) = 0$ with a forward difference gives $||\delta(1)|| \approx ||\delta(1 - \triangle x)||$, measured in the Euclidean $\mathbb{R}^N$ norm. Then, dependent upon the

condition of the matrix $K$, we could have $||\Lambda(1)|| \approx ||\Lambda(1-\triangle x)||$. Then $\tau'(1) \approx 0$, and so the recovered $a'(1) \approx 0$ as well. Accordingly, we see that the recovered coefficient adopts these characteristics of the selected eigenfunctions.

Using this information, we perform the recovery of

$$a(x) = \begin{cases} .1, & 0 \leq t \leq \frac{1}{2} \\ .3, & \frac{1}{2} < t \leq 1 \end{cases}$$

using the three alternate sets of eigenfunctions. With $\phi_n(x) = \sqrt{2}\cos\left[\left(n-\frac{1}{2}\right)\pi x\right]$, we assume knowledge of the coefficient at $x = 1$ and set $a_0 = a(1)$. The results are as indicated theoretically above: The recovery at $x = 1$ is exact, while $a'(0) \approx 0$. The associated recoveries shown in Figure 7.9 (using mixed boundary conditions in the PDE) are as successful as those achieved with the first set of eigenfunctions.

We view the same coefficient recovery using mixed conditions and the eigenfunctions $\phi_n(x) = \sqrt{2}\sin(n\pi x)$. We note in this case, it should be equally advantageous to assume knowledge of $a(0)$ or $a(1)$. In fact, knowing both would provide a distinct advantage, if we chose to solve the comparison problem for a non-constant coefficient $a_0$. We don't pursue that here, and choose $a_0 = a(0)$. Although the general shape is loosely identifiable in Figure 7.10, the recovery is notably degraded.

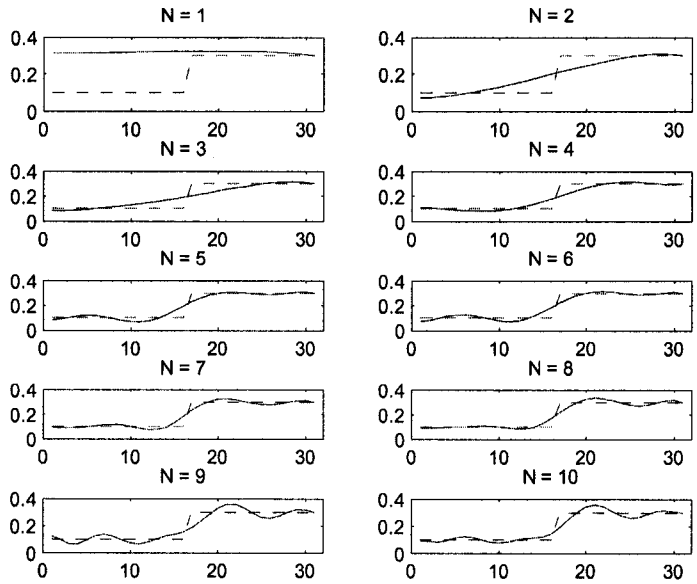


Figure 7.9. Using the eigenfunctions $\phi_n(x) = \sqrt{2}\cos\left[\left(n - \frac{1}{2}\right)\pi x\right]$

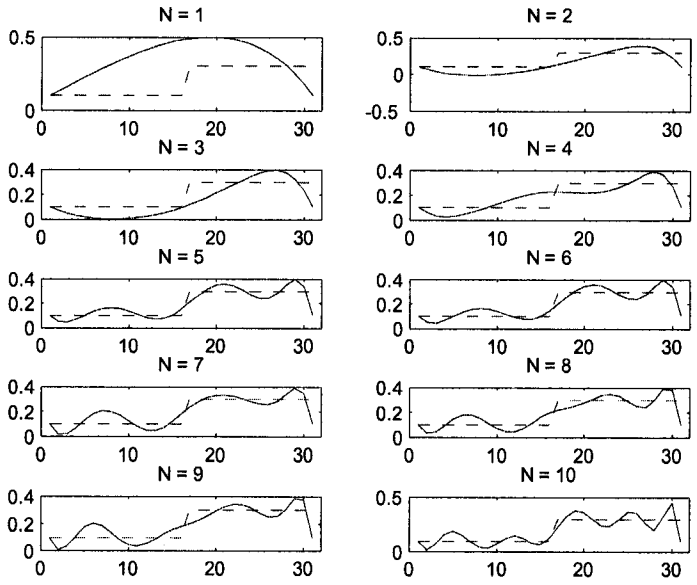


Figure 7.10. Using the eigenfunctions $\phi_n(x) = \sqrt{2}\sin(n\pi x)$

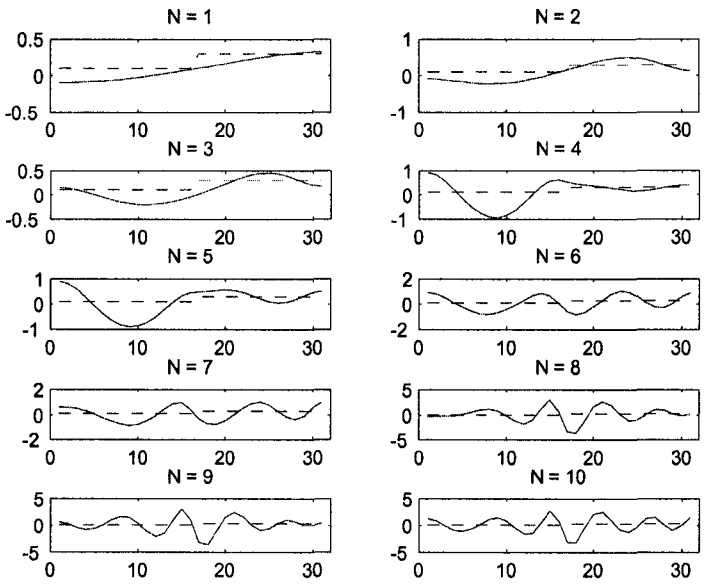


Figure 7.11. Using the eigenfunctions $\phi_n(x) = \sqrt{2}\cos(n\pi x)$

Finally, we view the recovery with the remaining eigenfunction set, $\phi_n(x) = \sqrt{2}\cos(n\pi x)$, which affects the slopes at each boundary value. The results in this case are severly degraded, as seen in Figure 7.11. Evidently, this latter case allows too much flexibility in the endpoint recovery, and adversely affects the interior recovery as well. On the other hand, controlling the state at each boundary value is disadvantageous, if we choose a constant comparison coefficient and the actual values are not equal. The eigenfunctions associated with mixed conditions allow an exact recovery at the endpoint with $a$ known, while giving the needed flexibility at the other value, leading to the best recoveries.

## 7.3. Adjoint Inputs

The adjoint inputs used in Section 7.1, $g_j^*$ and $h_j^*$, were the same for every recovery. Of course, there is an unlimited number of available inputs to use, but some are more convenient to use than others. Specifically, we use three criteria for selecting inputs.

(1) Linear independence: As shown in Chapter 4, this is a sufficient condition for the solvability of the linear system $K\Lambda = \delta$.

(2) $g_j^*(1) = h_j^*(1) = 0$ : With Dirichlet boundary conditions, these match the final condition in the adjoint problem, $w(x,1) = 0$. With Neumann conditions, these match the zero slope given by the final condition as well.

(3) Indexed: This leads to convenient coding.

We look at two cases in some detail. First, we examine the inputs used in Section 7.1, namely

$$\begin{aligned} g_j^*(t) &= \begin{cases} 1, & 0 \le t \le \frac{j-1}{N} \\ \frac{N(t-1)}{j-N-1}, & \frac{j-1}{N} < t \le 1 \end{cases} \\ h_j^*(t) &= \sin(j\pi t). \end{aligned} \tag{7.2}$$

With Dirichlet boundary conditions and $N = 5$, we have that the adjoint solution oscillates at the right boundary (due to $h_j^*$), and is decreasing at the left (due to

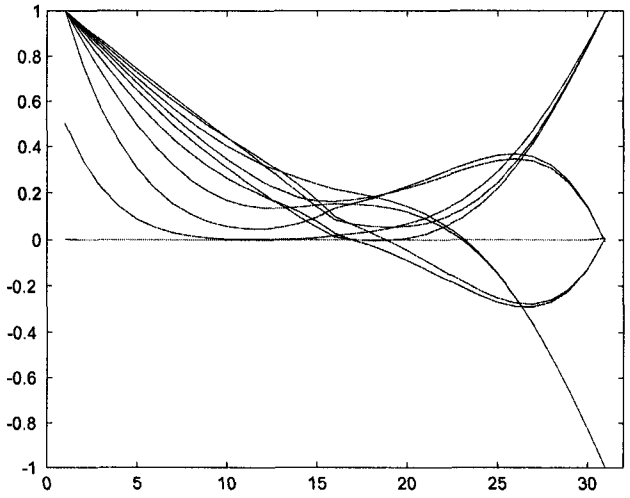


Figure 7.12. Adjoint solution profiles using inputs (7.2)

$g_j^*$). With the coefficient

$$a(x) = \begin{cases} .1, & 0 \le t \le \frac{1}{2} \\ .3, & \frac{1}{2} < t \le 1 \end{cases},$$

we have adjoint solution profiles for $w_5(x,t)$ noted in Figure 7.12. The inputs lead to five linearly independent functions $K_j(x)$ used in the recovery method. These are shown in Figure 7.13.

Compare these functions with those produced by the following adjoint inputs,

$$g_j^*(t) = (1-t)^j, \quad h_j^*(t) = (1-t)^{1/j},$$

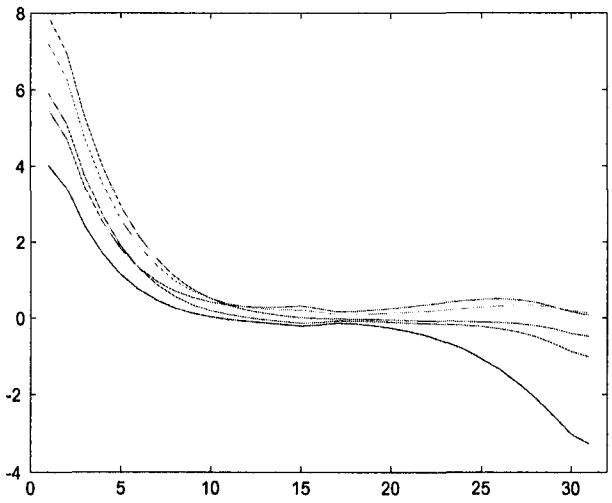


Figure 7.13. The linearly independent functions $K_j(x)$ using inputs (7.2)

this time with Neumann conditions (thus controlling the slope at each boundary value) and $N = 5$. Profiles for $w_5(x,t)$ and the functions $K_j(x)$ are shown in Figures 7.14 and 7.15. Although the constrasting adjoint inputs in these two examples produce linearly independent functions $K_j(x)$, the nature of each are significantly different. One might surmise that the subsequent recoveries will differ as well. However, this is not the case, as seen in Figure 7.16, where the recovery using inputs (7.2) is annotated as the Dirichlet recovery; the recovery using inputs (7.3) is annotated as the Neumann recovery. Other choices and combinations for adjoint inputs and boundary conditions led to similar results. These two examples

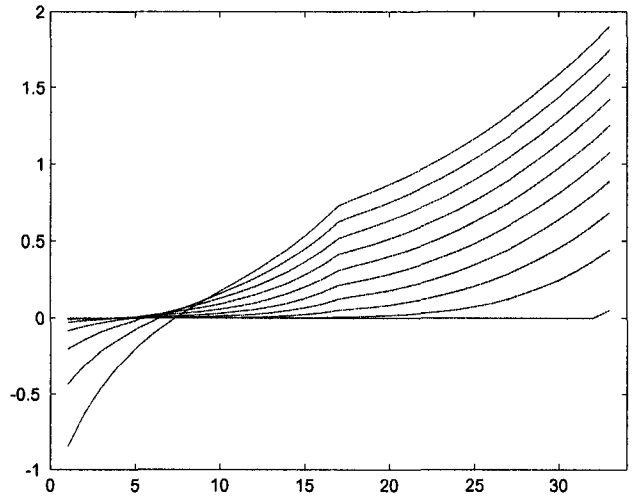


Figure 7.14. Adjoint solution profiles using inputs (7.3)

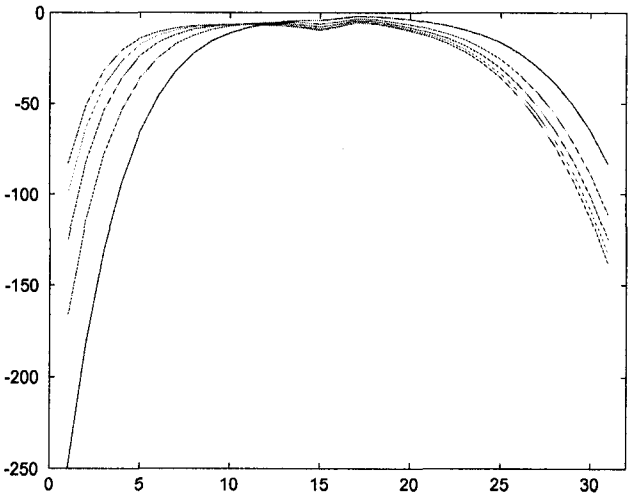


Figure 7.15. The linearly independent functions $K_j(x)$ using inputs (7.3)

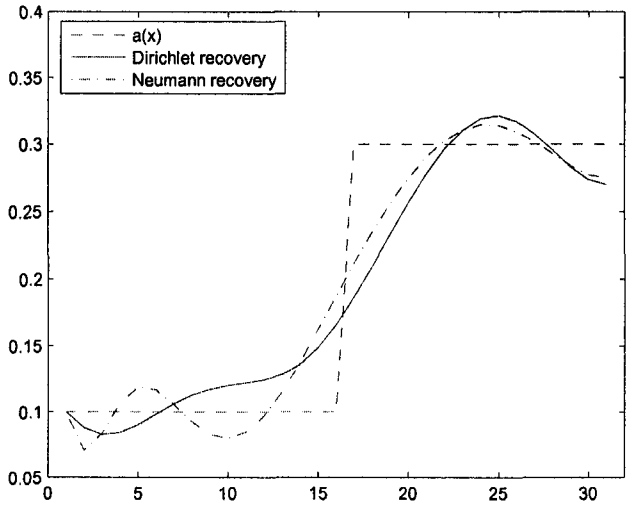


Figure 7.16. A comparison of adjoint input types

were chosen simply to highlight the fact that the selected inputs are less important to the recovery than initially thought.

Another perhaps more interesting question is how a zero adjoint input at one end might affect the recovery. This is equivalent to limiting our data observations to just a portion of the boundary. Per the identification results presented in Chapter 5, we realize that recovery does not require the dual observations we've employed thus far, at least theoretically. We view this effect in Figure 7.17, recovering the same coefficient as in the previous example. Here we used mixed boundary conditions and subspace dimension $N = 7$. We alternately set $g_j^* = 0$ with $h_j^* \neq 0$ and $h_j^* = 0$ with $g_j^* \neq 0$ for the single observation tests. Clearly, a successful

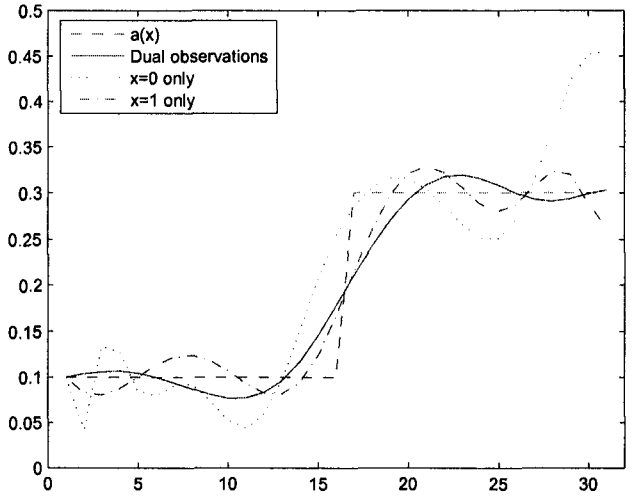


Figure 7.17. Dual versus single observations

recovery is accomplished in either case. As expected, recording observations on the entire boundary led to slightly better results.

## 7.4. Hilbert Space

We briefly look at the Hilbert Space setting for the recoveries: $H^s(U)$ and $H^{-s}(U)$. To this point, we've been using $s = 1$, but now view the effects of using different values. We recover the same coefficient as previously, again use mixed conditions and $N = 7$, and look at the recoveries associated with $s = 0, 1/2, 1$. We note that $s = 0$ is the only one of these values that leads to any significant change, and that the recovery is improved with $s = 1/2, 1$. These latter values result

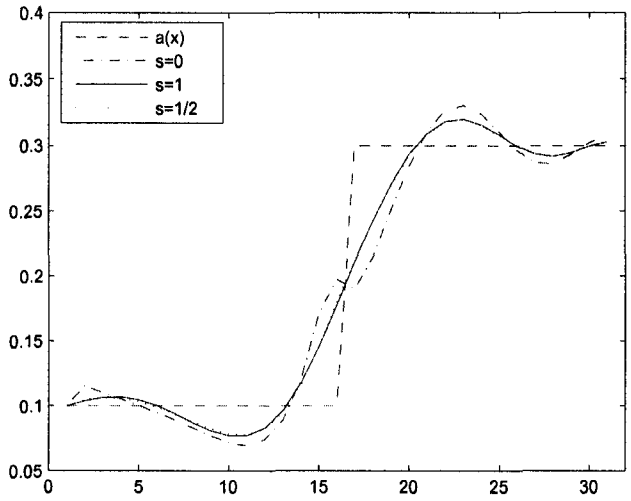


Figure 7.18. Varying $s$ in $H^s(U)$

in essentially the same recoveries, the difference which is nearly imperceptible in Figure 7.18. This is true for $s > 1$ as well (not shown).

## 7.5. Grid Selection

Finally, we briefly compare the effects of grid size on the subsequent recoveries. For convenience we use the same coefficient and parameters above, but change the spatial grid from 30 points to 20. Of course, this changes the time grid as well, in this case from 1800 to 800 points. The subsequent degraded recovery is seen in Figure 7.19.

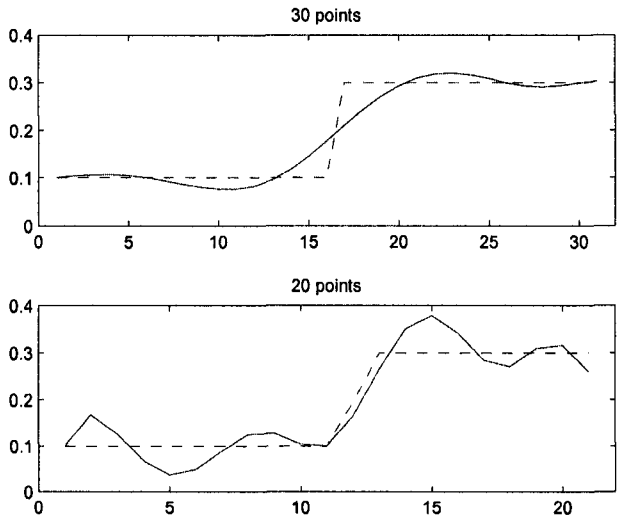


Figure 7.19. Grid comparison

### 7.6. Estimated Coefficient

We now perform a recovery using an estimated coefficient in the adjoint problem for various subspace dimensions and boundary conditions, keeping the other parameters used in Section 7.1. For convenience, we recover the "ramp-like" coefficient and let $\hat{w} = \hat{w}(a_0)$, where $a_0$ is the comparison problem coefficient. This removes the artificiality from the recovery process as shown, but of course introduces error in the problem. To see the effects of this realistic change, we first recover the coefficient using mixed conditions. In Figure 7.20 we see the typical near linear recoveries at lower subspace dimensions, but with notably poor results at $x = 1$. The oscillations induced at higher subspace dimensions make using

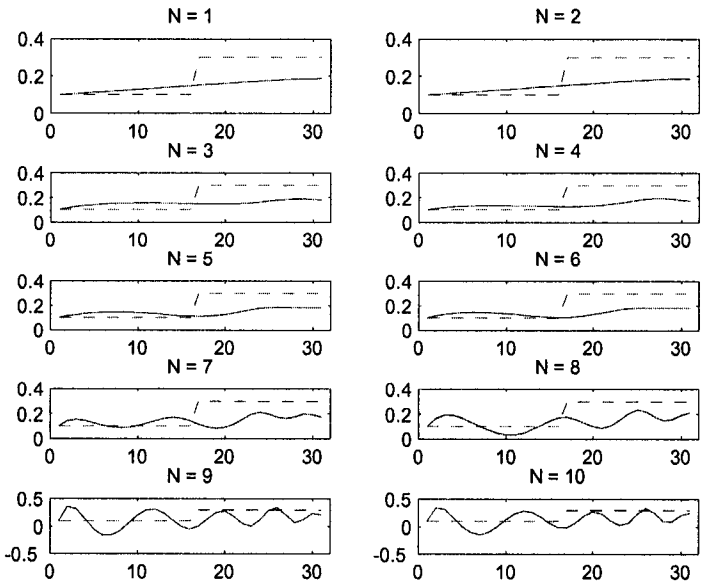


Figure 7.20. Mixed conditions, $\hat{w} = \hat{w}(a_0)$

$N > 10$ futile. We repeat the process with Dirichlet conditions, and note (Figure 7.21) better recoveries at $x = 1$, but with degraded results again. With Neumann conditions, we note (Figure 7.22) better results than the previous two, but with much room for improvement, as expected. This leads us to seek a better method for the actual recovery process, which is the subject of Chapter 8.

It is clear this modified Backus-Gilbert method works, in the sense that it provides a reasonable visual approximation of the unknown coefficient; to what extent is still in question. We can use the information gained in this section to guide the subsequent true recovery processes ahead, i.e. using an approximate

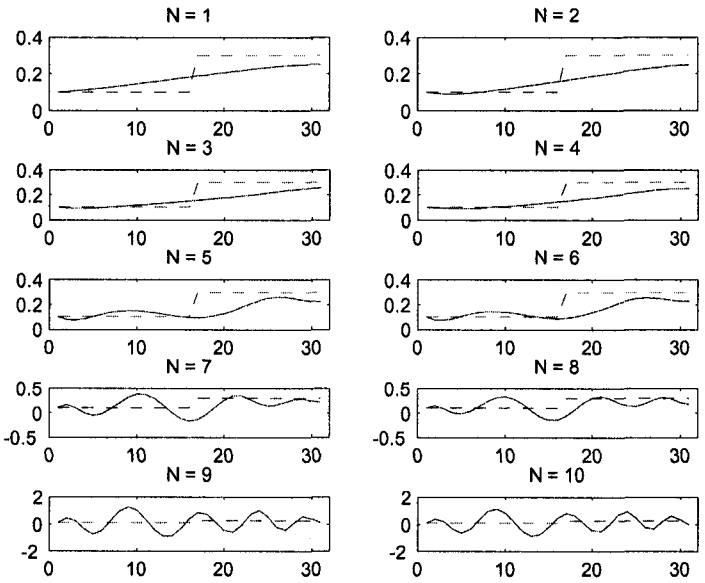


Figure 7.21. Dirichlet conditions, $\hat{w} = \hat{w}(a_0)$

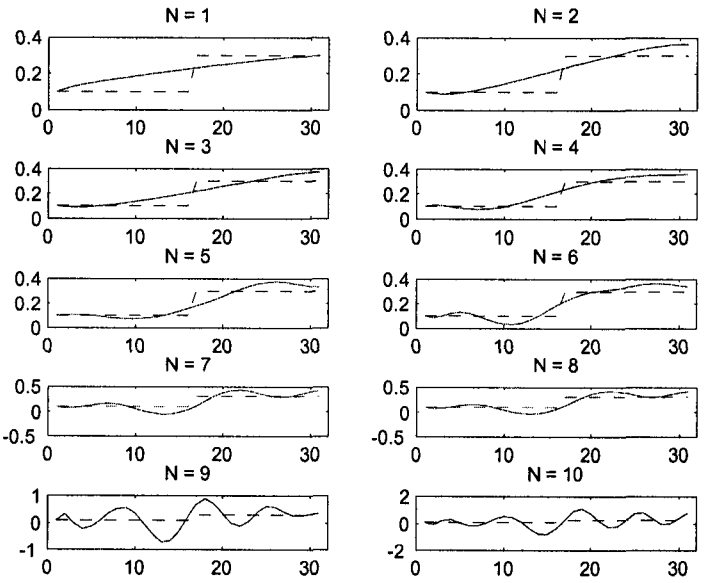


Figure 7.22. Neumann conditions, $\hat{w} = \hat{w}(a_0)$

adjoint $\hat{w} = \hat{w}(\hat{a})$, where $\hat{a}$ is $a_0$ or some other value – constant or otherwise. This obviously has a profound effect on the coefficient recovery.

CHAPTER 8

# RESULTS

The previous chapter contained somewhat unexpected results. Specifically, the success of the "artificial" recovery using $w = w(a)$, even with exceptionally ill-conditions systems, indicates that the choice of the approximate adjoint coefficient is more important than initially realized. More to the point, the successful recoveries with the true adjoint solution suggest the possibility of iterating the adjoint in some fashion, i.e. replacing the approximate coefficient with one increasingly more accurate – we hope – to the true coefficient. For if we can achieve a decent recovery with an initial constant coefficient, then maybe further improvement can be realized via this process, even with excessively high condition numbers of the $K$ matrices. This predictor-corrector technique has often been used as a successful regularization method, in countering the ill-posed nature of first-kind integral inverse problems [**16**].

## 8.1. The Process and Considerations

For reasons mentioned in Section 7.2, we use the eigenfunctions $\phi_n(x) = \sqrt{2}\sin\left[\left(n-\frac{1}{2}\right)\pi x\right]$, and make the assumption we can experimentally determine

the coefficient to be recovered at $x = 0$. We then set our comparison coefficient $a_0$ equal to $a(0)$ and start the process with the initial input to the adjoint problem $\hat{a}_1 = a_0$. Then we solve for the initial (approximate) adjoint solution $\hat{w}_1 = \hat{w}(\hat{a}_1)$, and use this in the remaining computations to achieve the recovery $\hat{a}_2$. Then, solving $\hat{w}_2 = \hat{w}(\hat{a}_2)$, we repeat, creating the discrete dynamical system

$$\begin{aligned} R(\hat{a}_n) &= \hat{a}_{n+1}, \quad n = 1, 2, \ldots \\ \hat{a}_0 &= a_0, \end{aligned}$$

where $R$ represents the complete recovery process. In testing the method, we check for accuracy by computing a normalized $L^2(U)$ difference between the recoveries at step $n$, and the true coefficient. For notational convenience, we denote

$$\beta_n := ||a - \hat{a}_n||_2 / ||a_0|| = ||a - \hat{a}_n||_2 / a_0.$$

With our concerns of stability using an explicit finite differences scheme, we must remain aware of the possibility of increasing the value $\max_{x \in U} \{\hat{a}_n\}$, which would necessitate a smaller time step, as detailed above. Thus after each iteration, we must peform a check for this value, and adjust the grid accordingly, or reduce values outside the realm of expected recovery.

Table 8.1. Recovery parameters

| | |
|---|---|
| Subspace dimension | $N$ |
| Boundary conditions | Mixed, Dirichlet, Neumann |
| Forcing terms | $f_0(t), \quad f_1(t)$ |
| Adjoint inputs | $g^*(t), \quad h^*(t)$ |
| Hilbert space setting | $s$ |
| Comparison coefficient | $a_0$ |

Of course, we may also generate negative values for the recovered coefficient as well, which can lead to algorithmic instabilities. One method of dealing with this, employed here, is to replace negative values with predetermined small positive values, creating an approximation to our approximation to be used in the next iteration. Although simple and computationally convenient, this provides no guarantee of further stability, as the new approximation may lead to additional instabilities in the adjoint solution.

## 8.2. Parameters

There are many parameters to adjust, listed in Table 8.1. We fix all but the subspace dimension and boundary conditions. In an attempt to facilitate future experimental considerations as well as the identifiablity criteria, we use constant forcing terms for Neumann conditions and linear, nonconstant terms for Dirichlet. Specifically, we assign the time dependent boundary conditions as listed in Table 8.2. As such, we're able to "shape" the comparison solution as monotonic functions

Table 8.2. Selected boundary conditions

| | |
|---|---|
| Mixed: | $f_0(t) = t, \quad f_1(t) = -5$ |
| Dirichlet: | $f_0(t) = t, \quad f_1(t) = -t$ |
| Neumann: | $f_0(t) = -5, \quad f_1(t) = -5$ |

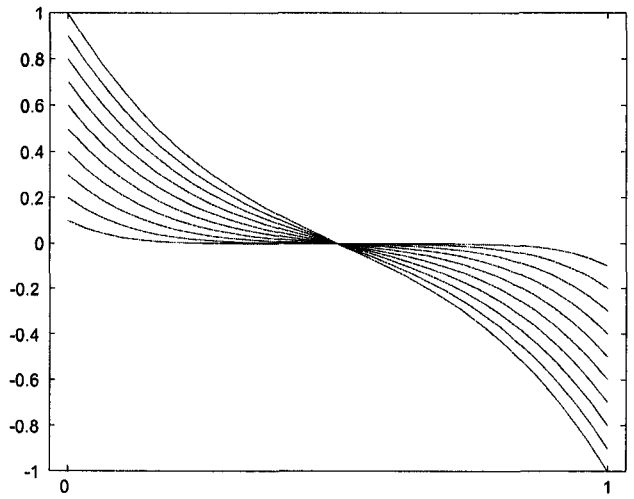


Figure 8.1. "Shaped" comparison solution profiles

over time, such that $\partial_x v$ is nonzero over much of the domain (see Figure 8.1). Of course, this directly affects the functions

$$K_j(x) = \int_0^T \partial_x v \partial_x w_j dt,$$

and thus the subsequent recovery. Logically, the same consideration could be given to the adjoint solution. However,these inputs must be linearly independent, which necessitates further complexity in the choices. We use the same adjoint inputs

as in Section 7.1, which work consistently well even though the adjoint gradient values are zero over some of the domain.

$$g_j^*(t) = \begin{cases} 1, & 0 \le t \le \frac{j-1}{N} \\ \frac{N(t-1)}{j-N-1}, & \frac{j-1}{N} < t \le 1 \end{cases}$$
$$h_j^*(t) = \sin(j\pi t).$$

Additionally, we continue to use the Hilbert space setting given by $s = 1$, and spatial grid steps of $\Delta x = \frac{1}{20}, \frac{1}{30}$ for data collection and recovery, respectively. With these parameters fixed, we vary the subspace dimensions, boundary conditions, and coefficient types, and view the resulting outputs. We look in some detail at the first set of recoveries, then present summaries thereafter.

### 8.3. "Ramp" Recovery

We first attempt the recovery of the "ramp-like" coefficient

$$a(x) = \begin{cases} .1, & 0 \le x \le \frac{1}{2} \\ .3, & \frac{1}{2} < x \le 1 \end{cases}.$$

Accordingly, we have for our initial adjoint coefficient, $a_0 = .1$. First, using mixed boundary conditions along with subspace dimension $N = 5$, we generate approximately 40 iterations, and view the results in Figure 8.2. The first iteration

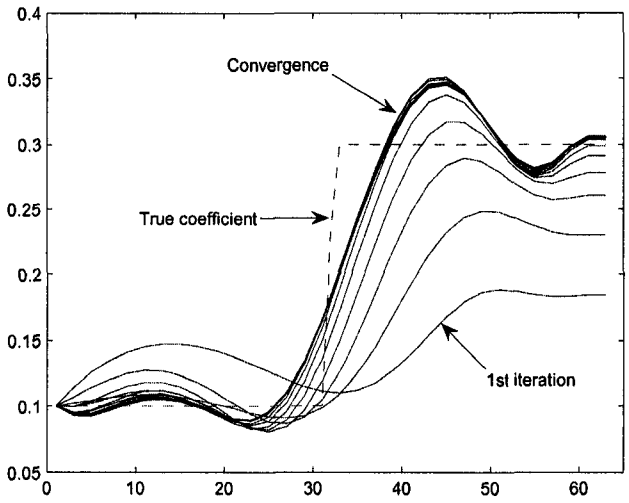


Figure 8.2. Mixed conditions, $N = 5$

is less than impressive, and differs significantly from the true coefficient, with $\beta_1 \approx .102717$. However, if we continue to iterate, the subsequent recoveries quickly begin to resemble the true coefficient shape quite nicely. In fact, we achieve the best recovery with iteration number 12, where $\beta_{12} \approx .028395$. As the iterative process continues, we have apparent convergence, with $\beta_{29} \approx .028552$. This is a neglible decline in accuracy from iteration 12 – plotting $\hat{a}_{12}$ versus $\hat{a}_{29}$ results in no discernible difference visually. Using subspace dimension $N = 6$, we achieve similar accuracy. Interestingly, with $N = 7$, the initial 10 iterations "bypass" more decent recoveries as $\max_{x \in U} \{\hat{a}(x)\}$ increases to approximately .75. The iterations then "settle" back to a closer recovery, with apparent convergence by about the

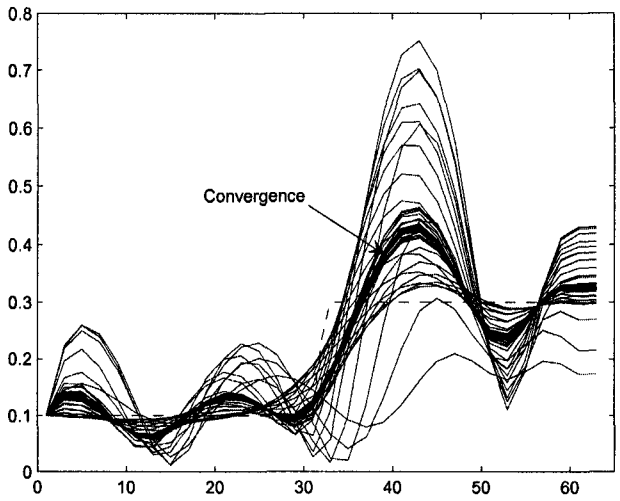


Figure 8.3. A larger variance with $N = 7$

$70th$ iteration, albeit to a degraded normalized difference of $\beta_{70} \approx .0532$. This is not suprising, as the entire Backus-Gilbert method is an approximating technique by design. This is seen in the Figure 8.3. Upon close inspection, it might be evident from this plot the convergence is not to the best approximation, which actually occurs at iteration number 16, with $\beta_{16} \approx .026940$. This recovery is shown in Figure 8.4. Incidentally, increased subspace dimensions results in unusable recoveries, as oscillations increase significantly.

We repeat the recovery with Dirichlet boundary conditions. This time, subspace dimension $N = 6$ produces the best result, nearly as accurate as with mixed boundary conditions, with $\beta_{18} \approx .029265$, visually noted in the Figure 8.5.

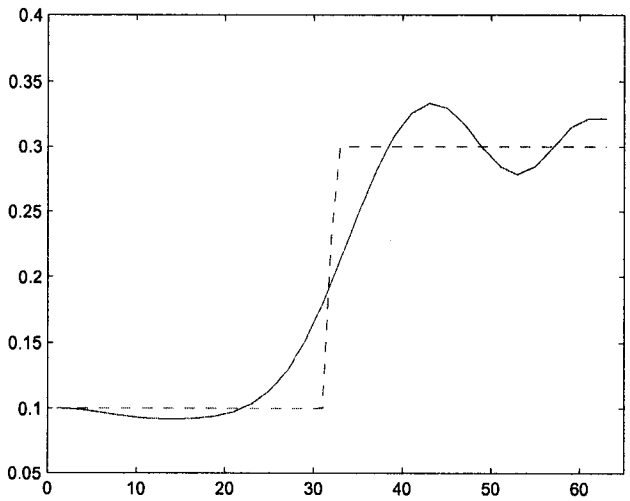


Figure 8.4. The best approximation at iteration 16

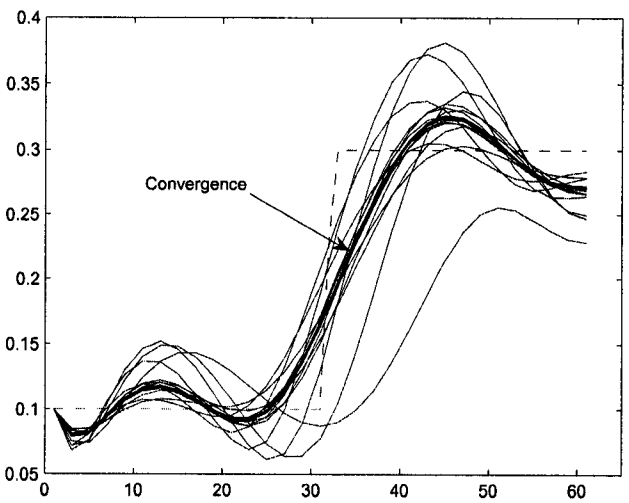


Figure 8.5. Dirichlet conditions, $N = 6$

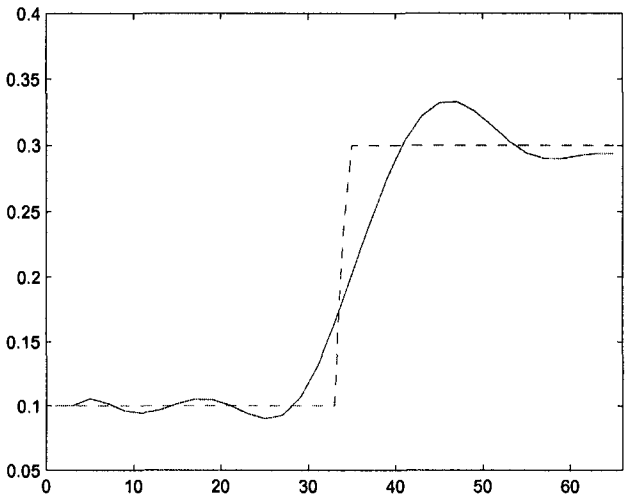


Figure 8.6. Neumann conditions, $N = 7$

Lastly, we perform the recovery using Neumann boundary conditions. The best recovery occurs with subspace dimension $N = 7$, and is just slightly better than the previous conditions, with $\beta_{30} \approx .025838$ (see Figure 8.6). Interestingly, with the higher subspace dimension $N = 8$, if we just look at the first few iterations, it appears the oscillatory behavior associated with will prohibit a decent recovery. We can see in the Figure 8.7 that the early iterated recoveries become too small (negative) and also exceed the maximum allowed value input in the Matlab code, and are adjusted accordingly for the next iteration. Although more iterations are necessary, we do achieve a decent recovery by iteration number 70. In Figure 8.8, iterations 60 through 80 are shown. In this case, $\beta_{80} \approx .026058$.

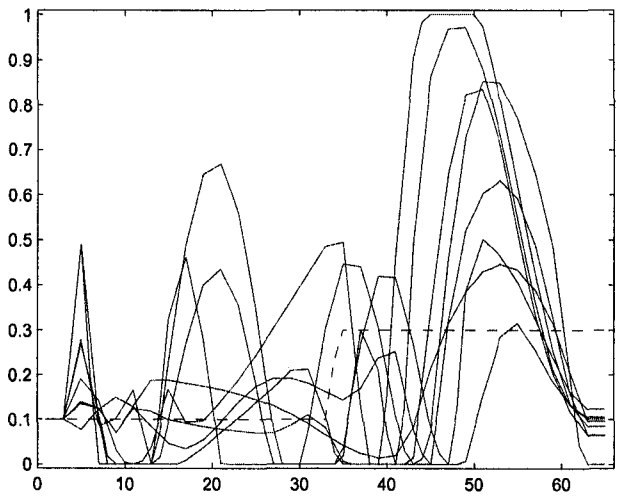


Figure 8.7. Early iterations with $N = 8$

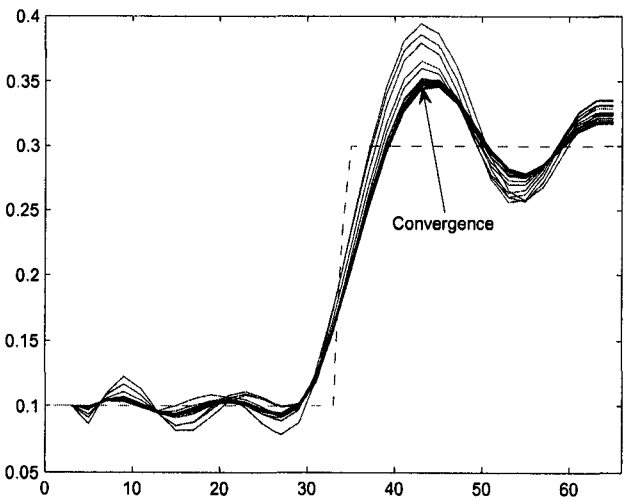


Figure 8.8. Later iterations with $N = 8$

## 8.4. "Step" Recovery

We now perform similar analyses on piecewise constant coefficients, with an increased value on a portion of the interior, referred to here as a "step".

$$a(x) = \begin{cases} .1, & 0 \leq x \leq x_1 \\ .3, & x_1 < x \leq x_2 \\ .1, & x_2 < x \leq 1 \end{cases} .$$

We'll work with both symmetric and asymmetric coefficients, beginning with the former, where $x_1 = \frac{1}{3}$ and $x_2 = \frac{2}{3}$. Because observations are made on the boundary and the true coefficient value is the same on the entire boundary, with $N = 1$ we expect a near constant solution. This is precisely the result, as the limited subspace dimension allows no chance of recovering the interior rise in the coefficient. In the mixed case, with much higher subspace dimensions, we achieved decent recoveries. We achieved the best recovery with $N = 8$ – sufficient "convergence" occured by the $11th$ iteration, with the results shown in Figure 8.9. In this case, $\beta_{11} \approx .035232$.

With Dirichlet boundary conditions, we achieve a similarly successful recovery, the best associated with subspace dimension $N = 7$. In this recovery, iterations approach the convergent solution in an orderly fashion, with the value $\max\{\hat{a}(x)\}$ in a neighborhood of $x = \frac{1}{2}$ increasing in an apparent monotonic fashion to an

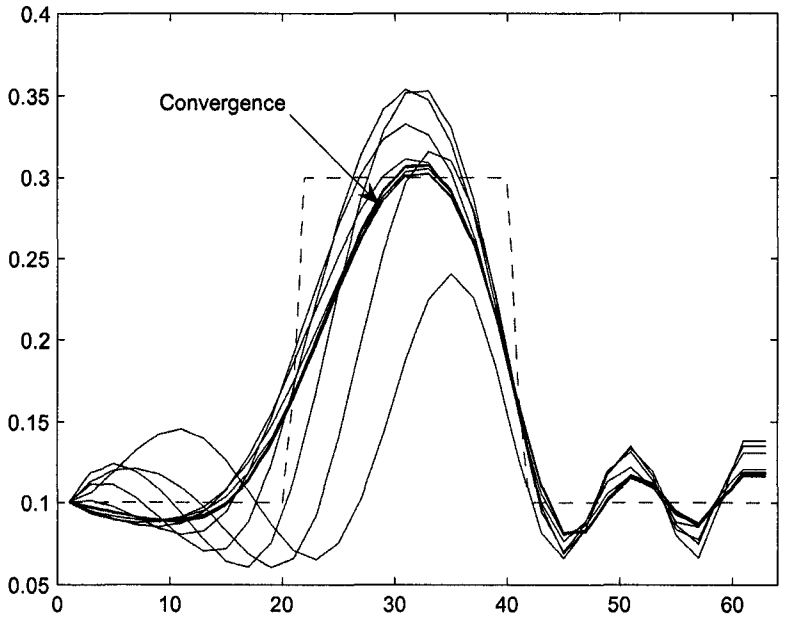


Figure 8.9. Mixed conditions, $N = 8$

eventual difference value $\beta \approx .034625$. This is seen in the Figure 8.10. Incidentally, $N = 8$ led to similar results, and $N = 9$ exhibited extreme oscillatory behavior.

Lastly, we use Neumann boundary conditions, achieving the best result with $N = 7$ and $\beta \approx .041901$. The iterations again exhibit the apparent monotonicity of the maximum value, as shown in Figure 8.11. The result, however, is notably degraded from those earlier.

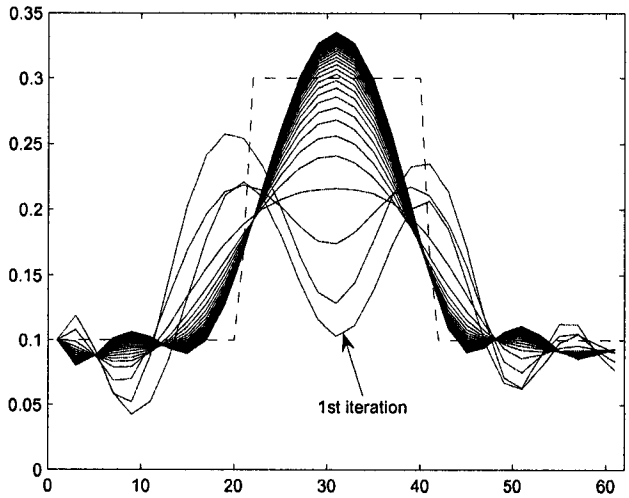


Figure 8.10. Dirichlet conditions, $N = 7$

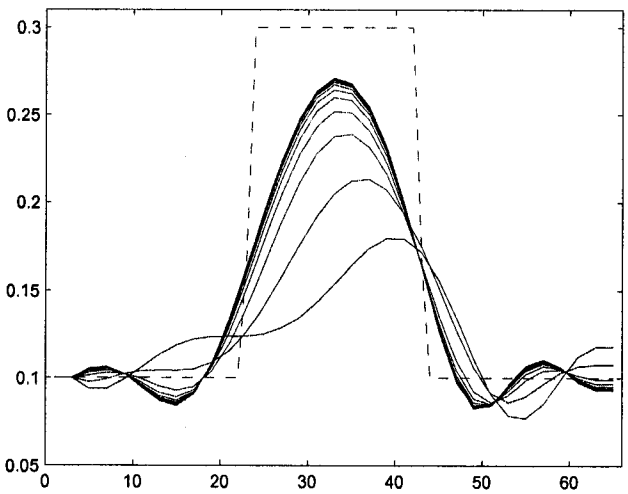


Figure 8.11. Neumann conditions, $N = 7$

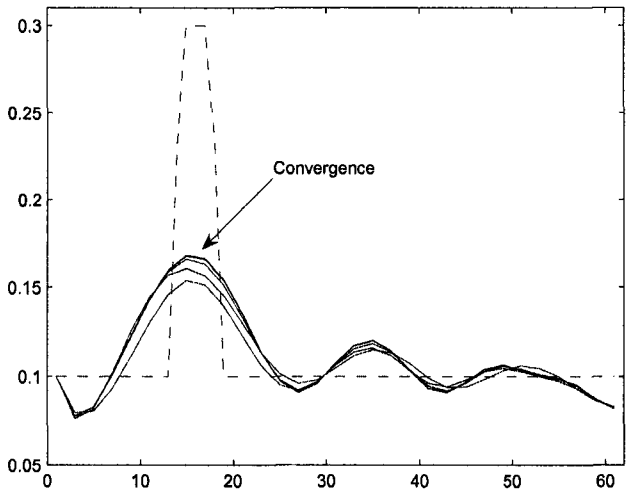


Figure 8.12. An asymmetric, thin step, near $x = 0$

We now offset and change the length of the "step" in the coefficient,

$$a(x) = \begin{cases} .1, & 0 \leq x \leq .2, \quad .3 < x \leq 1 \\ .3, & .2 < x \leq .3 \\ .1, & .3 < x \leq 1 \end{cases},$$

and merely show the best result (Figure 8.12), achieved with subspace dimension $N = 9$ and Dirichlet boundary conditions. The convergence was very fast (sufficient by iteration 5) but the results were less accurate than desired, with $\beta \approx .037561$. Perhaps we can contribute the poorer result to two factors:

(1) Closer proximity to the boundary, specifically $x = 0$, where the measured data and nature of the eigenfunctions used may tend to override the algorithmic attempts to recover the higher interior values.

(2) The smaller interval of the maxima.

To see the potential effect on the interval size, we perform a similar recovery with

$$a(x) = \begin{cases} .1, & 0 \leq x \leq .2, \quad .3 < x \leq 1 \\ .3, & .2 < x \leq .4 \\ .1, & .3 < x \leq 1 \end{cases},$$

using the same Dirichlet conditions that led to the best result above. The result, shown in Figure 8.13, better captures the higher conductivity present on the interior. Further increasing the interval size improves recovery of the difficult step, as shown in Figure 8.14. Placing the asymmetric step near $x = 1$ and reducing its length again, while keeping all else the same results in better recovery of the step itself, albeit it with increased oscillations elsewhere. The convergent solution is shown in the Figure 8.15.

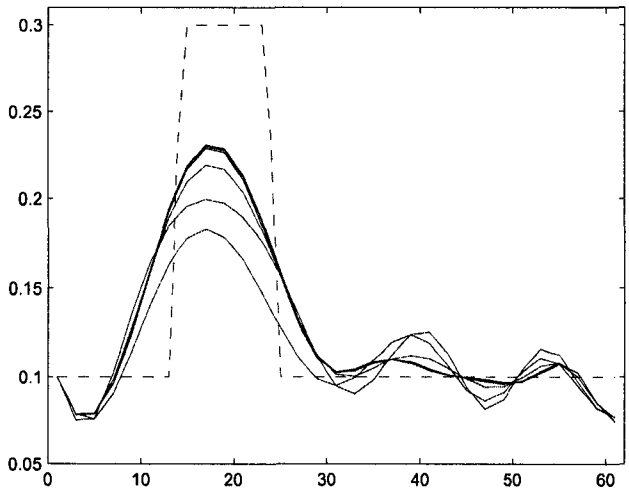


Figure 8.13. An asymmetric, less thin step

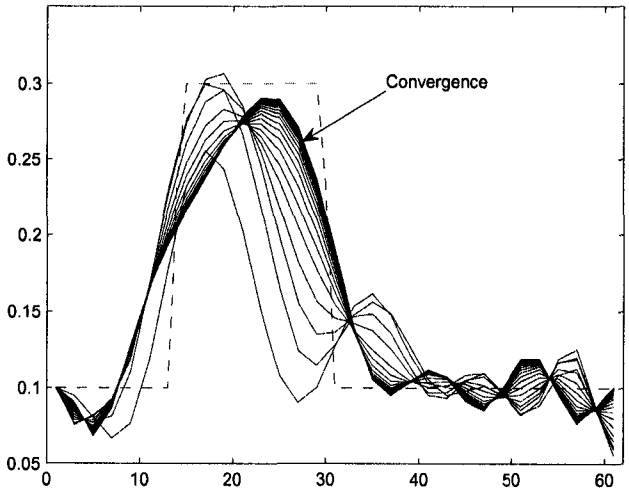


Figure 8.14. An asymmetric, wide step

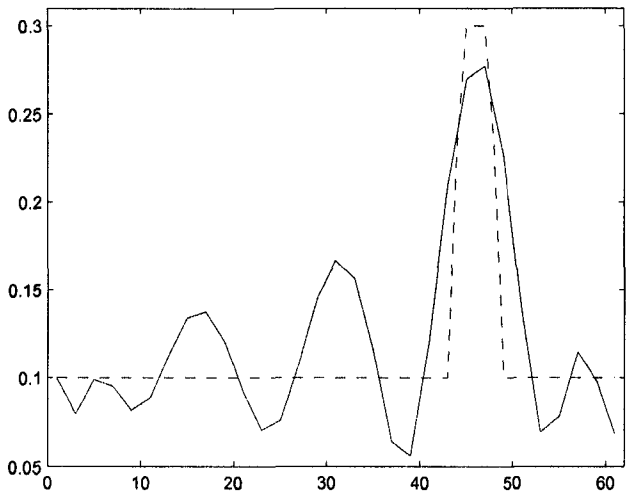


Figure 8.15. An asymmetric, thin step, near $x = 1$

As in the previous chapter, we attempt the recovery of a couple more difficult step functions, the first the piecewise constant, increasing coefficient

$$a_1(x) = \begin{cases} .1, & 0 \leq x \leq \frac{1}{3} \\ .2, & \frac{1}{3} < x \leq \frac{2}{3} \\ .5, & \frac{2}{3} < x \leq 1 \end{cases}.$$

From the subsequent recoveries, such as the one shown in Figure 8.16, we see the overall increasing trend is captured, but in reality we would have no knowledge of the interior conductivity, as the recovery does not indicate this portion in any notable fashion. This recovery was accomplished with Neumann boundary conditions and subspace dimension $N = 7$. The final step coefficient we'll recover

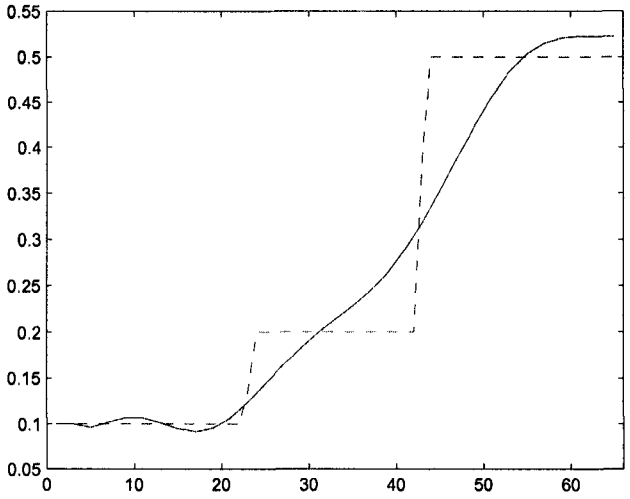


Figure 8.16. A more difficult recovery: $a_1$

is

$$a_2(x) = \begin{cases} .1, & 0 \leq x \leq \frac{1}{3} \\ .5, & \frac{1}{3} < x \leq \frac{2}{3} \\ .3, & \frac{2}{3} < x \leq 1 \end{cases} .$$

The method has an easier time with this recovery, and captures the general shape quite nicely, despite some erratic behavior in the initial iterations. The end result is shown in Figure 8.17, and was accomplished with Dirichlet boundary conditions and subspace dimension $N = 6$.

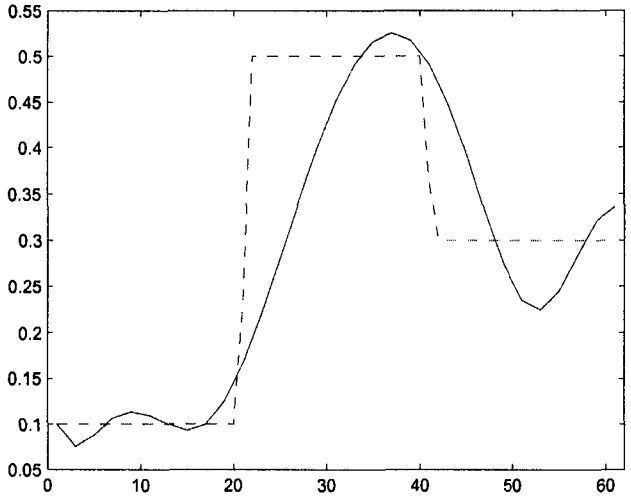


Figure 8.17. A more difficult recovery: $a_2$

## 8.5. Linear Recovery

We now attempt the recovery of two linear functions. Of course, from the results in Chapter 7, we expect a lower necessary subspace dimension, since these seem to result in near linear recoveries. We begin with the coefficient $a(x) = .2x + .1$ and find mixed boundary conditions most accurate, with the converged recovery exhibiting $\beta \approx .003442$. The recovery is shown in Figure 8.18, and was accomplished with subspace dimension $N = 1$. Subsequent iterations were monotonically increasing towards the actual coefficient. Dirichlet conditions led to similar accuracy, its best result corresponding to $N = 2$. Neumann conditions also captured the shape of the coefficient, but were considerably less accurate.

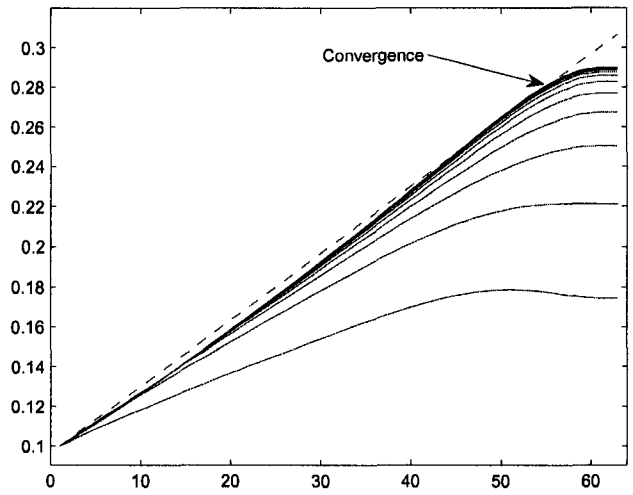


Figure 8.18. A linear coefficient, slope = .2

We increase the slope to .5 to see if this changes the capabilities of the algorithm, and find it does not. We achieve very similar results, the best with Dirichlet conditions with subspace dimension $N = 3$. The convergent solution is shown in Figure 8.19. Finally, we change to a negative slope: $a(x) = 1 - .5x$. The method is still successful, with similar results as the previous two, but this time with different iterative behavior. The successive iterates alternate "above" and "below" the true coefficient, as they converge towards it. This is shown in Figure 8.20.

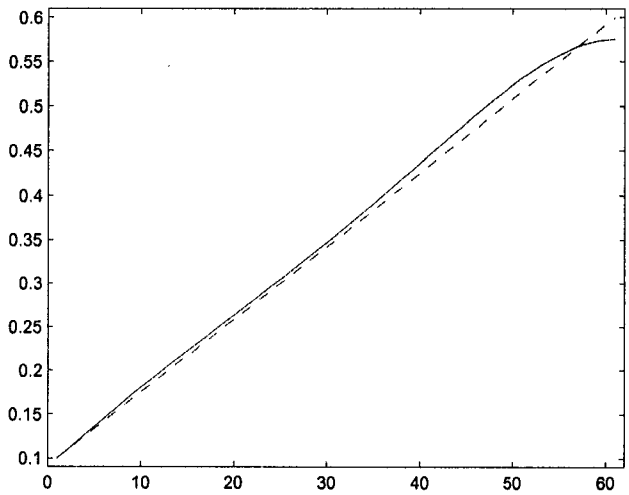


Figure 8.19. A linear coefficient, slope = .5

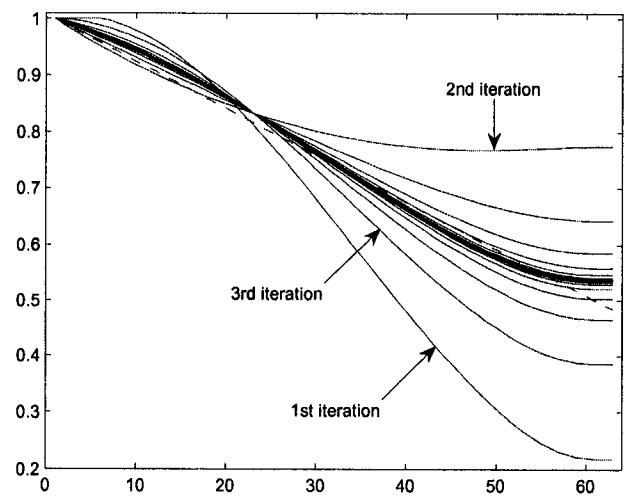


Figure 8.20. A linear coefficient, slope = $-.5$

CHAPTER 9

# EXPERIMENTAL CONSIDERATIONS

We now briefly consider aspects of this method as related to the collection of data in a physical experiment – used in determining the hydraulic conductivity of a column of soil, for example. We must take into account the the length of time required for experiments, the limitations of associated equipment, and the effects of minor errors in the data readings. Additionally, we consider the true nature of determining the recovery, and look at this first.

## 9.1. Realistic Determination

In our previous theoretical discourse, we have proceeded with complete knowledge of the coefficient to be recovered, using this to determine subspace dimensions and associated accuracies of the recoveries. Of course in reality, we may have very little knowledge of the coefficient itself, since the entire purpose of this inverse method is its determination. As mentioned, we assume we can experimentally determine the conductivity on part of the boundary, specifically choosing $x = 0$ in our work thus far. Beyond this and the admissibility criteria mentioned in the identifiability section, we assume no further information a priori. The coefficients

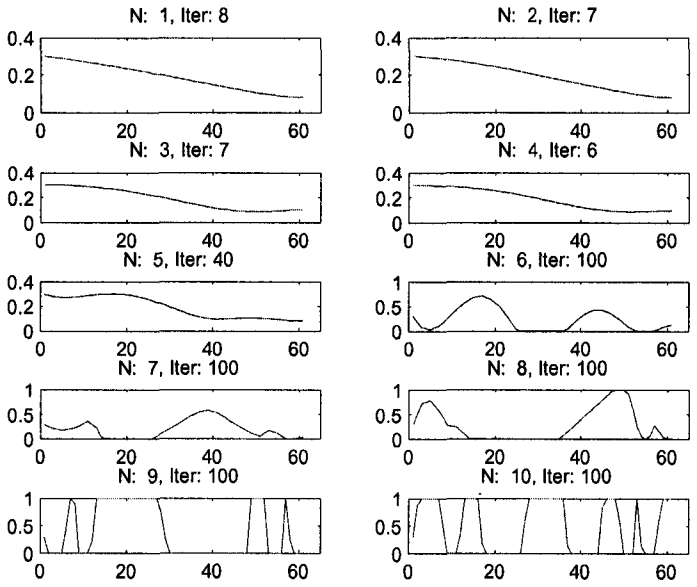


Figure 9.1. Recoveries with subspace dimension and iterations shown

to be recovered can be constant, linear, piecewise constant, etc., which could affect the success of the method with certain subspace dimensions.

With this in mind, we attempt a recovery using subspace dimensions $N = 1$ through $N = 10$, using the iterative "stop criteria" $||\hat{a}_n - \hat{a}_{n-1}||_2 / a_0 \leq .01$, chosen based on trial and error with the multitude of recoveries performed, including those in Chapter 8. Also used as a stop criteria is a limiting number of iterations, in this case 100, which experience has shown to be a sufficiently high number for assumed convergence in most instances. We show the results of the subsequent recovery, using Dirichlet conditions. Specifically, in Figure 9.1, we indicate the

subspace dimension used, as well as the number of iterations required to produce the output shown. We see that with the lower subspace dimensions, the degrees of freedom afforded for the recovery is limited and we have a smaller number of iterations required – this matches well with the previously accomplished iterative recoveries. By $N = 6$, we are maximizing the number of iterations allowed by the code, and the recovered coefficients are showing increased oscillatory behavior, to the point where certain values are truncated to preserve the both the stability of the numerical reconstruction, and the parabolic nature of the adjoint PDE. Based on this, and experience gained from prior work, we might surmise the recovery associated with $N = 5$ to be the most accurate. This is indeed the case, as shown with the actual coefficient in Figure 9.2. The recovery was repeated with mixed and Nuemann conditions, with similar results.

In general, we follow this process to ascertain the most effective recovery among those attempted: Start with $N = 1$ and continue to increase the subspace dimension until the required iterations reach the maximum value allowed by code. Of course, this is adjustable, but set as the typical point where oscillations increase to the extent they require truncation of the coefficient if continued iterations were needed. We take the previous recovery as the most accurate estimation of the true coefficient.

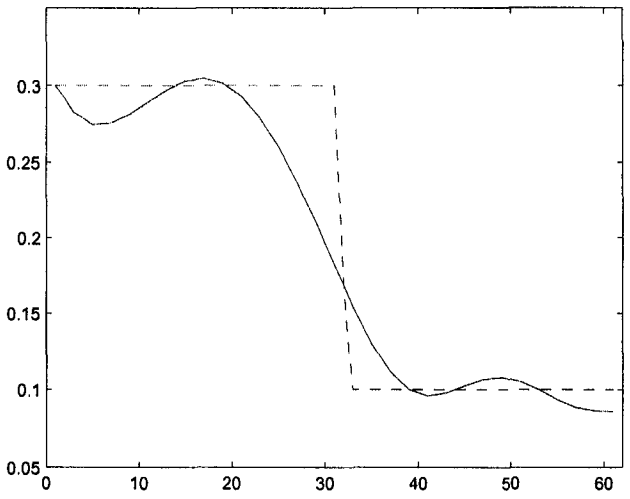


Figure 9.2. Recovery with $N = 5$ compared with actual coefficient

We perform a second recovery, again with Dirichlet conditions, and note the output in Figure 9.3. Again, we see relatively small numbers for iterations required up to a certain subspace dimension. Once we reach $N = 9$, the iterations increase from 17 to 100, with the associated recoveries exhibiting oscillatory behavior and truncated output. We would be correct to assume the recovery associated with $N = 8$ as a relatively close approximation to the true coefficient, as shown in Figure 9.4.

We look at another set of outputs, this time viewing only $N = 1$ through $N = 6$, since higher dimesions led to clearly diminished results. In Figure 9.5, the first four recoveries look relatively similar and require just a few iterations to

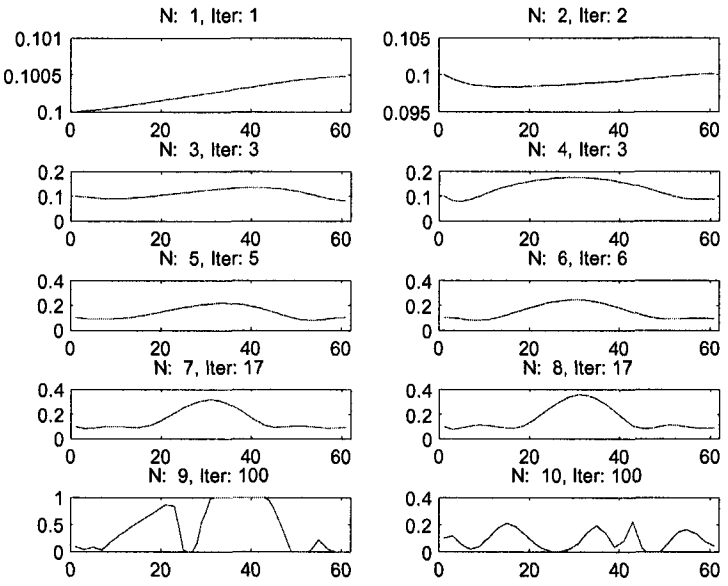


Figure 9.3. A different recovery

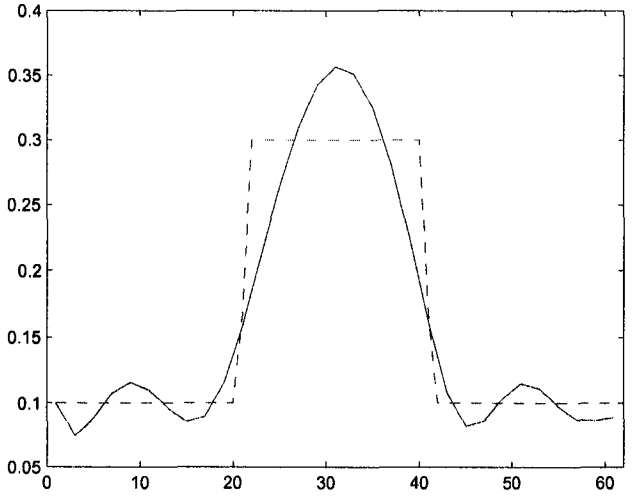

Figure 9.4. The best recovery, with $N = 8$

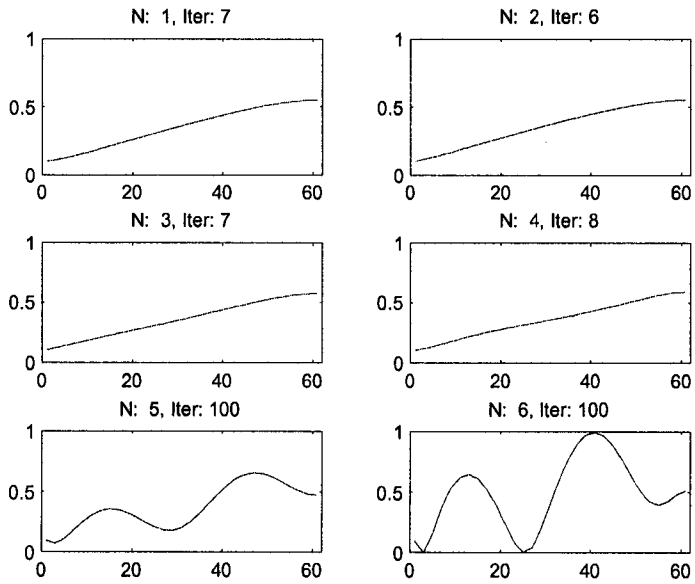


Figure 9.5. Another set of recoveries, this time $N = 1$ to 6 only

sufficiently converge. The iterations and oscillations increase with $N = 5$ and 6, but the output is not truncated this time. If we use iterations as our determining factor, we would choose the recovery associated with $N = 4$ as the best output. This is shown in Figure 9.6. In actuality, both the $N = 3$ and $N = 4$ recoveries are very accurate, with the former being just slightly better. As expected, the recovery of a linear function required a smaller subspace dimension than those previous.

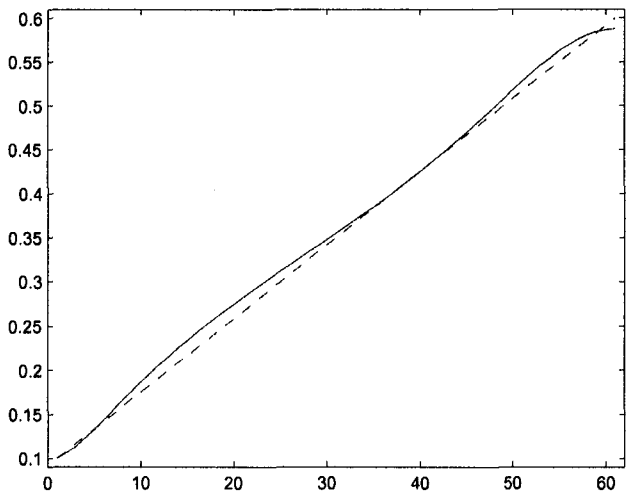


Figure 9.6. The best recovery, with $N = 3$

In the next example, we see similar output (Figure 9.7), and would choose the recovery associated with $N = 6$ as the best case, as the iterations increase significantly thereafter. Although the recovery of this linear function is most accurate for $N = 1$, the first six subspace dimensions give suprisingly accurate results – the additional freedom afforded by these higher subspaces does not notably degrade the recoveries until $N = 7$.

As a final example, we examine the outputs shown in Figure 9.8. This time we have two notable jumps in the number of iterations, between $N = 5$ and 6, and between $N = 6$ and 7. Since $N = 6$ produces a smooth solution, we might assume this is an accurate recovery. Figure 9.9 provides this confirmation.

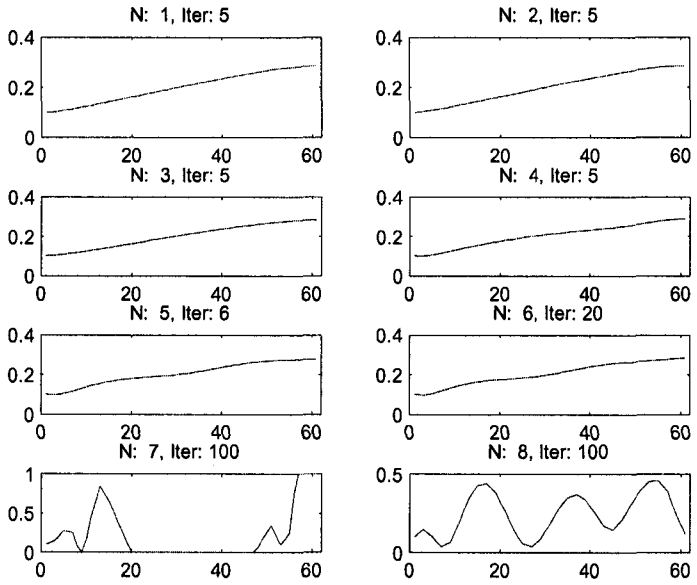


Figure 9.7. Decent recoveries through $N = 6$

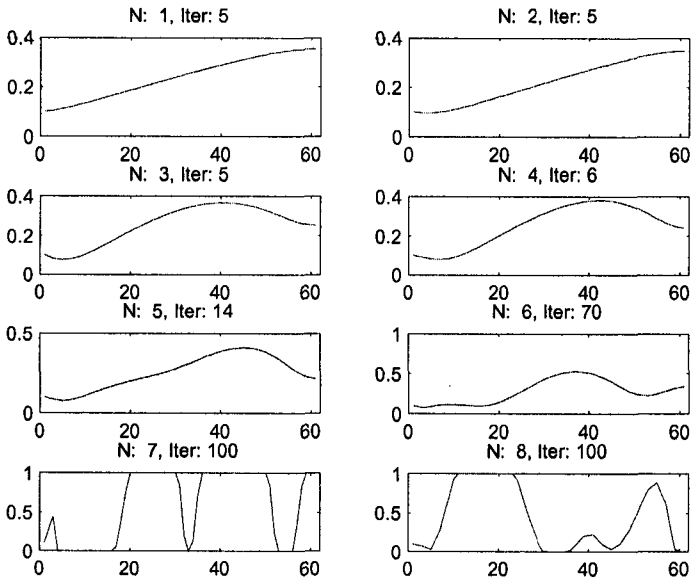


Figure 9.8. Two notable increases in iterations

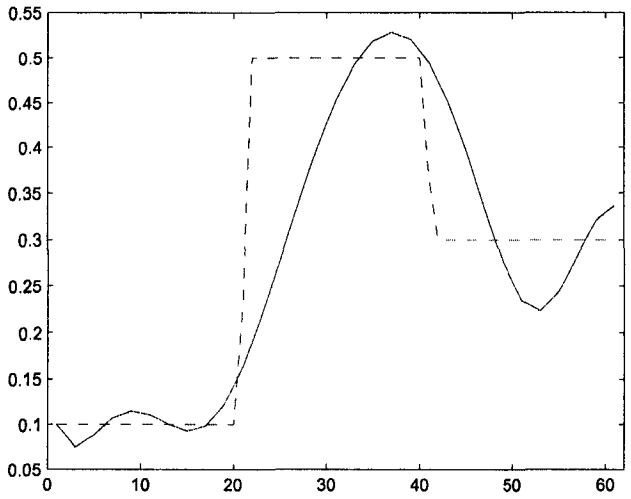


Figure 9.9. The best recovery, with $N = 6$

## 9.2. Time Scales

We note here that the failure to let the experiment proceed for a sufficient amount of time can lead to degraded results, since the effects of distinct coefficients minimally affect the observation data – minimally being the key word here, since the identifiability proof ensures some difference via the one-to-one observation operators. We show evidence of this by changing the magnitude of the coefficient to be recovered, i.e. the coefficient in the direct problem. This is equivalent to running an experiment for a longer or shorter amount of time. We see this simply

by considering the equation

$$\partial_t u = \partial_x \left( a \partial_x u \right), \quad t \in (0, T).$$

If we just increase the conductivity by a factor of $\alpha > 0$ on the same time domain, i.e.

$$\partial_t u = \partial_x \left( \alpha a \partial_x u \right), \quad t \in (0, T) \tag{9.1}$$

then we can rewrite as

$$\frac{1}{\alpha} \partial_t u = \partial_x \left( a \partial_x u \right), \quad t \in (0, T).$$

Now, with $\tau = \alpha t$,

$$\partial_t u \left( x, \tau \right) = \partial_\tau u \frac{d\tau}{dt} = \alpha \partial_\tau u,$$

we see that (9.1) is equivalent to

$$\partial_\tau u = \partial_x \left( a \partial_x u \right), \quad \tau \in (0, \alpha T). \tag{9.2}$$

Quite simply, increasing the conductivity by a factor of $\alpha$ and keeping the time scale the same (9.1) is equivalent to increasing the time scale by a factor of $\alpha$ and keeping the conductivity the same (9.2).

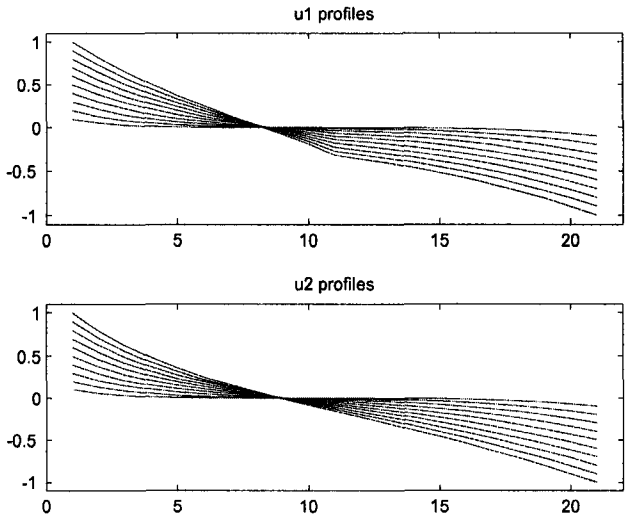


Figure 9.10. Notably distinct direct profiles

In this light, we view aspects of recoveries associated with our prototypical ramp coefficient and a related linear coefficient; specifically we compare

$$a_1 = \begin{cases} .1, & 0 \leq x \leq .5 \\ .3 & x > .5 \end{cases}$$

and

$$a_2 = .2x + .1.$$

We first view profiles (Figure 9.10) associated with the direct solutions involving each coefficient: $u_1 = u(x,t;a_1), u_2 = u(x,t;a_2)$. With solution $u_1$, we clearly see the effect of the discontinuity in $a_1$, not present with $u_2$. We record the

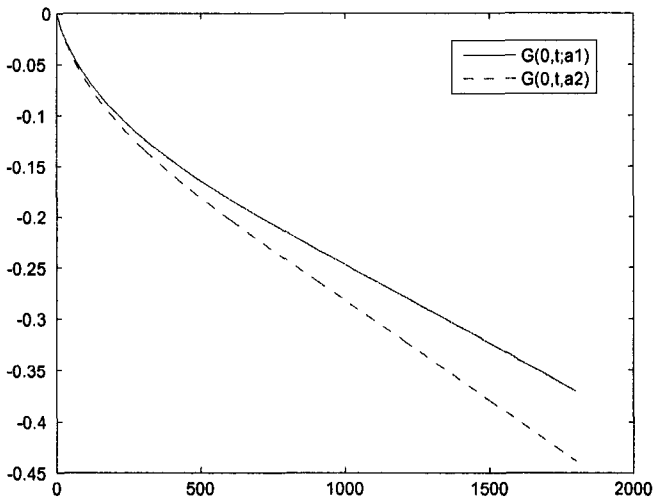


Figure 9.11. Observation differences at $x = 0$

observations on the boundary and note the differences in each as well. Using Dirichlet boundary conditions, at $x = 0$ we record the flow data: $G(0, t; a_1)$ and $G(0, t; a_2)$ (Figure 9.11). We repeat at $x = 1$ (Figure 9.12). The difference appears significant at $x = 0$ and less so at $x = 1$, but is sufficient in each case to result in notably different and reasonably accurate recoveries, as shown in Figure 9.13.

In contrast, we simulate running the experiment for a tenth of the time allotted above; i.e. we make $\alpha = .1$ and denote $a_3 = \alpha a_1$ and $a_4 = \alpha a_2$. Figure 9.14 shows the direct profiles for $u_3$ and $u_4$, which this time are not visibly different. Again we compare observations at $x = 0$ and $x = 1$ (Figures 9.15 and 9.16, respectively).

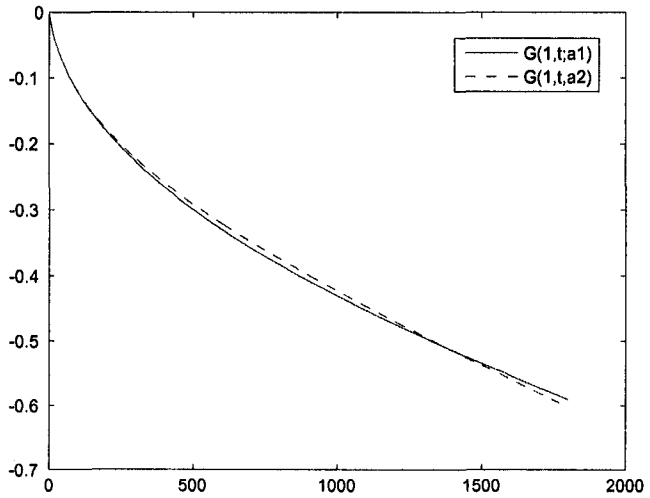


Figure 9.12. Observation differences at $x = 1$

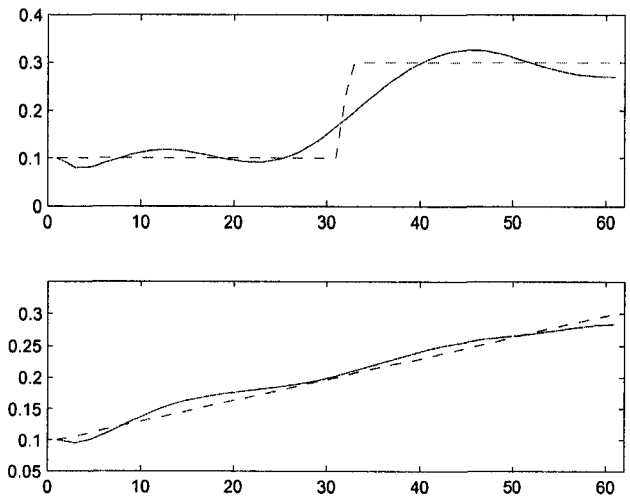


Figure 9.13. Distinct and accurate recoveries

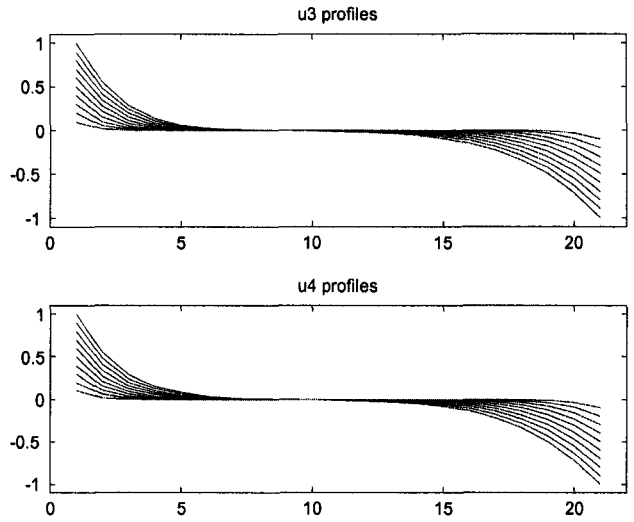


Figure 9.14. Similar direct profiles

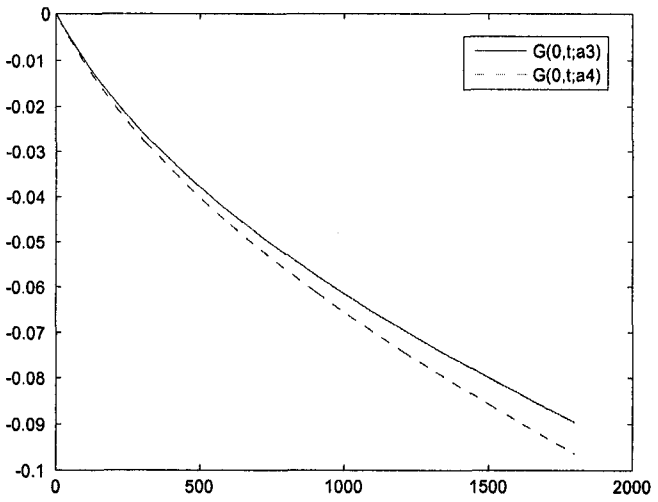


Figure 9.15. Observations at $x = 0$

This time we note both the degraded recoveries of each, but more importantly,

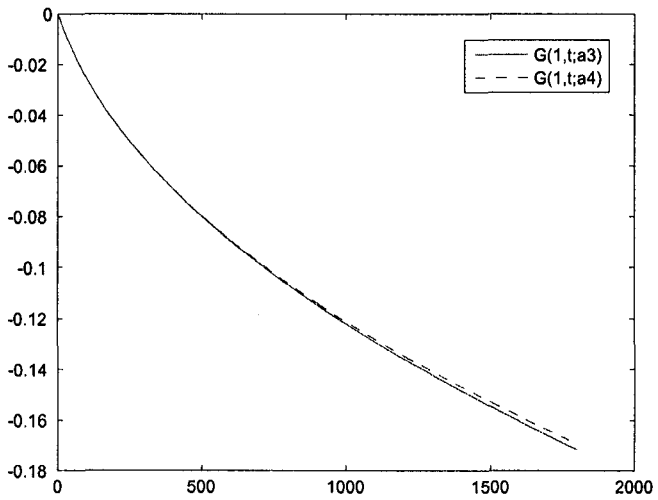


Figure 9.16. Observations at $x = 1$

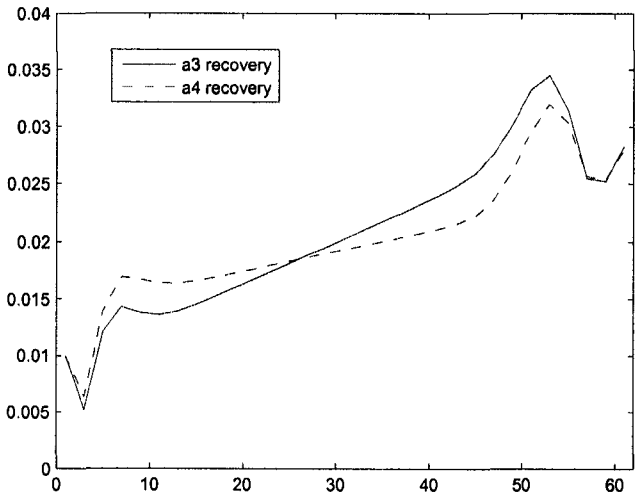


Figure 9.17. Similar, degraded recoveries

their similar structure, as shown in Figure 9.17. We note that the recovery of $a_3$

should more closely resemble the ramp, while $a_4$ should be linear. However, we lose the distinction present in the previous recoveries (Figure 9.13). Evidentally, the longer experiment allows differences in the coefficient to propagate sufficiently, to more greatly affect the observation data and the subsequent recovery.

## 9.3. Forcing Magnitudes

To counter possible limitations of data recording equipment, we note we can theoretically increase the magnitudes of the boundary forcing terms in the direct (and thus comparison) problems to similarly increase the associated observations noted. If done so symmetrically, i.e. the new forcing terms $\hat{f}_0$ and $\hat{f}_1$ are the same multiple $\alpha$ of the original functions ($\hat{f}_0 = \alpha f_0$ and $\hat{f}_1 = \alpha f_1$) then our recovery is unchanged while the observations increase in magnitude. This is due to the linear dependence of the solutions $u(x,t;f_0,f_1)$ to our PDEs on their boundary conditions:

$$u(c_0 f_0 + c_1 f_1) = c_0 u(f_0) + c_1 u(f_1).$$

As such, our new observation differences $\triangle\hat{G}_0$ and $\triangle\hat{G}_1$ are multiples of the same factor $\alpha$. Then $\hat{\mu}_j = \alpha\mu_j$. Similarly, since $v(x,t;\alpha f_0,\alpha f_1) = \alpha v(x,t;f_0,f_1)$,

$$\hat{K}_j = \int_0^T \partial_x w_j \partial_x (\alpha v)\, dt = \alpha K_j.$$

Then

$$\begin{aligned}\left(\hat{K}_i, \hat{K}_j\right)_{-s} &= \alpha^2 \left(K_i, K_j\right)_{-s}, \\ \left(\hat{K}_j, \delta_{x_0}\right)_{-s} &= \alpha \left(K_j, \delta_{x_0}\right)_{-s},\end{aligned}$$

and thus

$$\hat{K}\hat{\Lambda} = \hat{\delta} \Rightarrow \alpha^2 K\hat{\Lambda} = \alpha\delta \Rightarrow \hat{\Lambda} = \frac{1}{\alpha}K^{-1}\delta = \frac{1}{\alpha}\Lambda.$$

Then, with $\hat{\mu} = \alpha\mu$,

$$\hat{\tau} = \hat{\Lambda} \cdot \hat{\mu} = \left(\frac{1}{\alpha}\Lambda\right) \cdot \alpha\mu = \Lambda \cdot \mu = \tau.$$

Thus, if increased boundary forcing magnitudes are feasible in a laboratory setting, they can be used to bring the observation data to the level of sensitivity required by the data collection equipment, without (theoretically) changing the result provided by our algorithm.

## 9.4. Error Analysis

We now seek to analyze the effects of errors in the observed data on the resulting recovery. To do so, we denote the boundary observations $G_0(t, a)$ at $x = 0$ and $G_1(t, a)$ at $x = 1$, which are computed from the direct solution . We perturb each

within some selected window,

$$\begin{aligned} G' &= G\left[1+\varepsilon\left(t\right)\right], \\ -\varepsilon_0 &\leq \varepsilon\left(t\right) \leq \varepsilon_0, \end{aligned}$$

where the mean value of $\varepsilon\left(t\right)$ is equal to zero. Then the data $G_0'\left(t,a\right)$ and $G_1'\left(t,a\right)$ are interpolated/extrapolated to the recovery grid and used to find the new scalars $\mu_j'$, effectively constructing a new subspace for our projected approximation. Note the functions $K_j\left(x\right)$ in the solution process,

$$\left\langle \tau', K_j \right\rangle_{H^s \times H^{-s}} = \mu_j',$$

are independent of the observation data. With the new vector $\mu'$ constructed, we solve the linear system given in Equation (4.4),

$$K\Lambda = \left\{ \left(K_i, \delta_{z_0}\right)_{-s} \right\}_{i=1}^{N},$$

which prior to iteration produces the same vector $\Lambda$ as without error. We then solve

$$\tau' = \Lambda \cdot \mu_j',$$

and thus find $a_1'$. This recovery is then used in the subsequent iteration, and so forth, as previously.

With any level of error, the results can vary to some extent – we quantify this by taking the $L^2(U)$ difference between the recovery with no error ($\hat{a}$) and the recovery with error induced ($\hat{a}_e$), normalized by $a_0$ : $\hat{\beta} := ||\hat{a} - \hat{a}_e||_2 / a_0$. The trials lead to varying outputs, so we take an average result of a sufficient sampling.

We view a few results, fixing the subspace dimension and other parameters from previous successful recoveries, while allowing the boundary conditions and levels of induced error to vary. We first look at a ramp recovery under mixed conditions. Figure 9.18 shows the minimal effect of 1% error induced to the observations. In this case, the observations with and without error look nearly identical. The average difference $\hat{\beta}$ in outputs with 1% error was found to be approximately .22797. Figure 9.19 is indicative of this result. We increase the error to 3% and note the subsequent observations and recovery (Figures 9.20 and 9.21, respectively). This time the perturbation to the observations are notable, and the recovery is degraded accordingly, with an average $\hat{\beta} \approx .48477$.

We repeat the process with Dirichlet boundary conditions, with 1% and 3% error again induced, and note averages this time of $\hat{\beta} \approx .12384$ and .20661, respectively. See Figures 9.22 and 9.23.

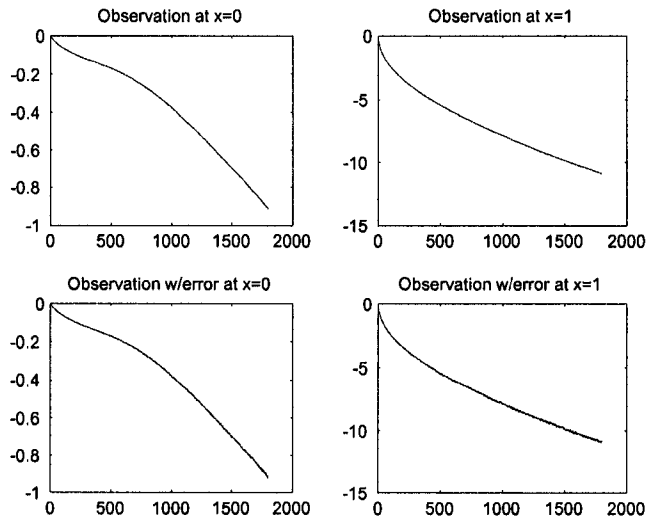


Figure 9.18. Observations with and without 1% error

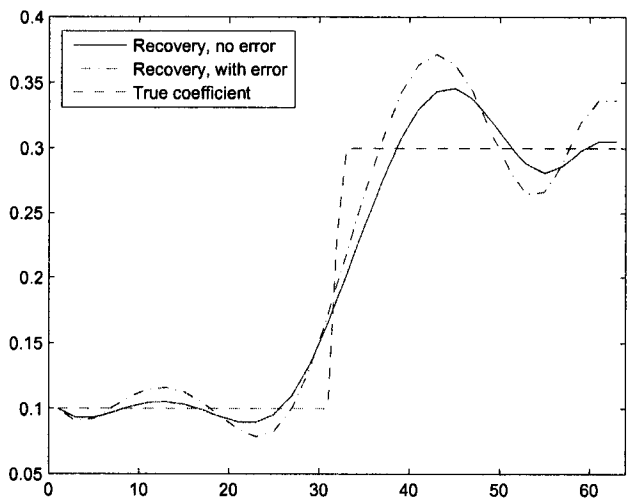


Figure 9.19. The recoveries, with and without 1% error

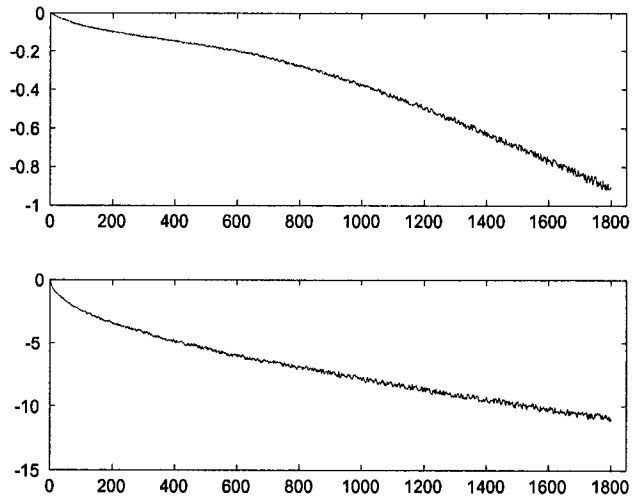


Figure 9.20. Observations with 3% error

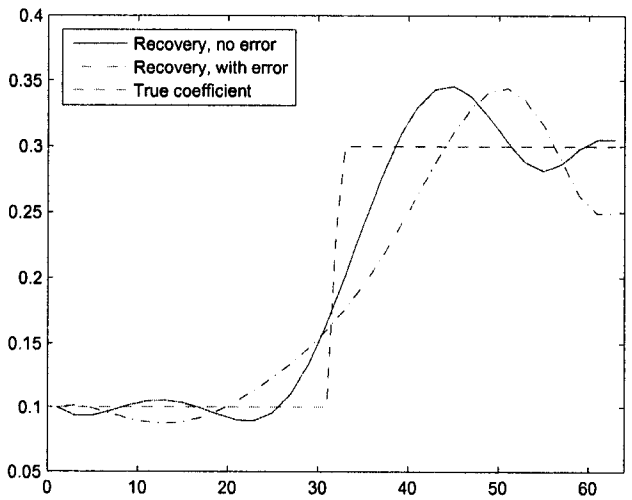


Figure 9.21. Recoveries with and without 3% error

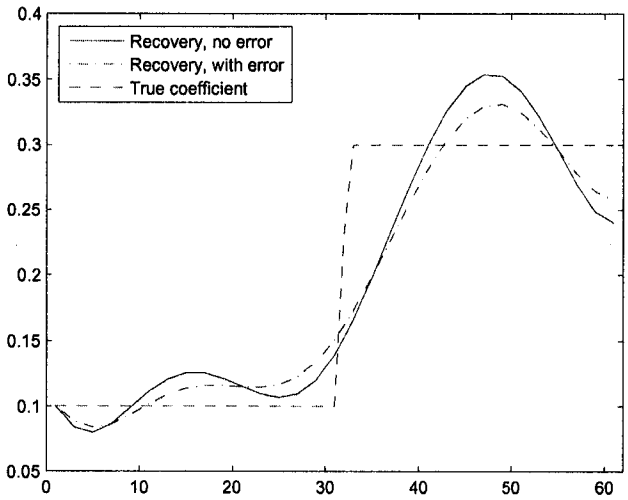


Figure 9.22. Dirichlet boundary conditions, 1% error

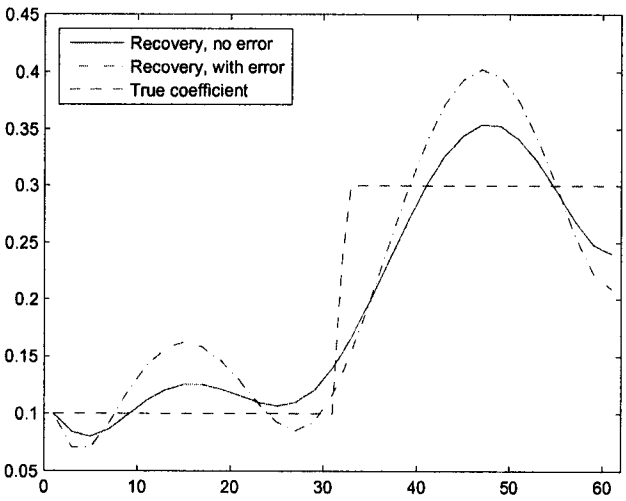


Figure 9.23. Dirichlet boundary conditions, 3% error

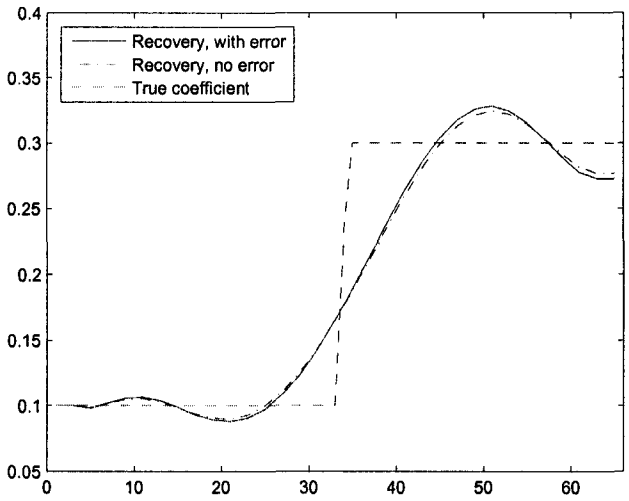


Figure 9.24. Neumann boundary conditions, 1% error

Lastly, we perform the error induced recoveries with Neumann conditions, and find notably improved averages $\hat{\beta} \approx .02237$ and .078305, for 1% and 3%, respectively. See Figures 9.24 and 9.25. We repeated the Neumann cases with 5% and 10% induced error to the observation data, and found the recoveries to be more robust than either other boundary condition.

We did note a positive correlation between the size of the subspace dimension used in the recovery and the effect of the error on that recovery. For example, using the above recovery with Neumann boundary conditions, we found the average $\hat{\beta}$ values listed in Table 9.1, indicating severely graded results with $N = 6$ and higher. Similar results were found for mixed and Dirichlet conditions.

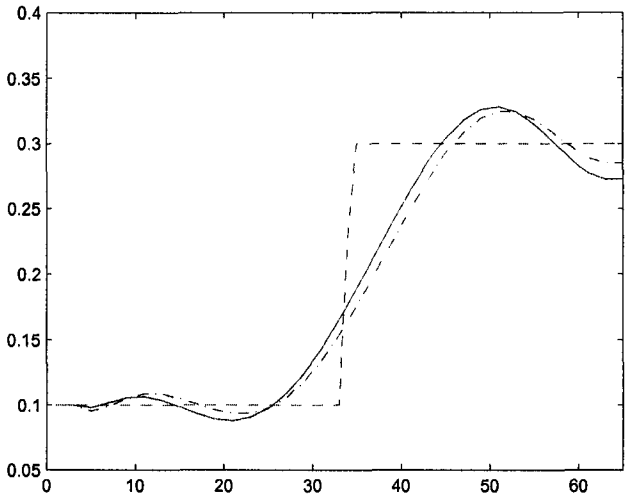


Figure 9.25. Neumann boundary conditions, 3% error

Table 9.1. Subspace dimension versus $\hat{\beta}$

| Subspace dimension/error | 1% | 3% |
|---|---|---|
| $N = 4$ | .01002 | .03102 |
| $N = 5$ | .02237 | .07831 |
| $N = 6$ | .79791 | 1.58394 |

Of course, this creates a dilemma for experimental data collection for use in the algorithm, in the recovery of conductivities which typically require a higher dimensional subspace for accuracy, as noted in the results chapter. Conversely, we should expect relative insensitivity to error for other recoveries, such as linear. As such, we test a linear recovery with error induced. Specifically, we recover the coefficient

$$a(x) = .5x + .1,$$

Table 9.2. Linear recoveries, boundary conditions versus $\hat{\beta}$

| Boundary conditions/error | 1% | 3% |
|---|---|---|
| Mixed | .07683 | .15514 |
| Dirichlet | .01180 | .03302 |
| Neumann | .00412 | .01527 |

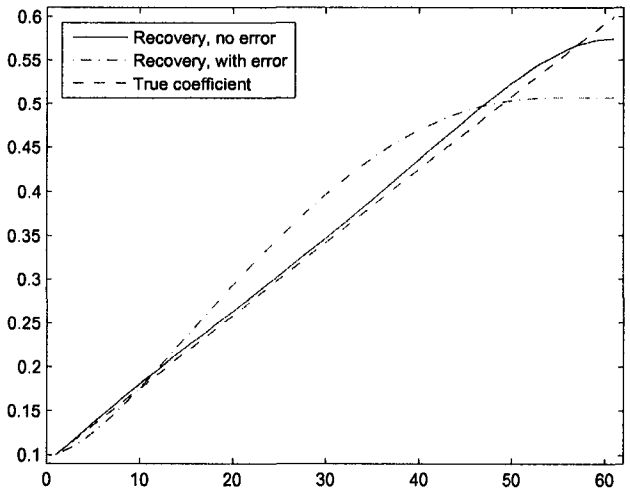


Figure 9.26. Linear recoveries, with and without 10% error

using subspace dimension $N = 5$. In Table 9.2, we list the average differences from the error free recoveries. With this extra stability, we can perform decent recoveries under higher levels of error. Figure 9.26 shows an atypically poor recovery under 10% error and Dirichlet conditions.

These facts persisted even under the ideal (and artificial) non-iterative algorithm using the true coefficient in the adjoint problem. Unfortunately, this negates any argument that attempts to correlate instabilities associated with induced error

to the iterative process alone. Although this may magnify the problem, clearly the matter is more basic, and will be the subject of future research aimed at determining the cause and fix. Perhaps some regularization method as documented in [**10**] is necessary to implement at this point.

CHAPTER 10

# A DIFFERENT PDE: q(x)

As mentioned in the introduction, we now briefly look at the recovery of a different coefficient, specifically $q(x)$ from Equation (1.2) with $s \equiv -1$ and $f \equiv 0$. We formulate the inverse problem and other theoretical considerations only, and omit associated computations. The intent here is to show that the method implemented can be used in the coefficient identification of a different parabolic PDE. We proceed similarly to our previous problem, basing the recovery of $q(x)$ from observed boundary data, and denoting the spatial and time domain components as in Equation (3.1).

The n-dimensional direct problem with mixed boundary conditions is given by (10.1).

$$\begin{aligned} \partial_t u(x,t) &= \nabla^2 u(x,t) + q(x)\, u(x,t), \quad (x,t) \in U_T \qquad (10.1) \\ u(x,0) &= 0, \quad x \in U \\ u(x,t) &= f_D(x,t), \quad (x,t) \in \Gamma_T^D \\ \nabla u(x,t) \cdot n &= f_N(x,t), \quad (x,t) \in \Gamma_T^N. \end{aligned}$$

Now suppose $v = v(x,t)$ solves (10.1) in the case $q(x) = q_0 = \text{constant}$, and $u = u(x,t)$ solves the same in the case $q(x) = q_0 + \tau(x)$. Then $z(x,t) = u(x,t) - v(x,t)$ solves (10.2):

$$\begin{aligned} \partial_t z - \nabla^2 z - qz &= \tau v, \quad (x,t) \in U_T \qquad (10.2) \\ z(x,0) &= 0, \quad x \in U \\ z(x,t) &= 0, \quad (x,t) \in \Gamma_T^D \\ \nabla z(x,t) \cdot n &= 0, \quad (x,t) \in \Gamma_T^N. \end{aligned}$$

The development of this difference problem is located in Appendix 1.2.

## 10.1. The Integral Equation

We now multiply equation (10.2) by a smooth function $w(x,t)$, and integrate by parts.

$$\begin{aligned} &\int_U \left[ wz|_0^T - \int_0^T z\partial_t w dt \right] dx - \\ &\int_0^T \left[ \int_\Gamma w \nabla z \cdot n ds - \int_U \nabla w \cdot \nabla z dx \right] dt - \int\int_{U_T} zqw dx dt \\ = &\int\int_{U_T} w\tau v dx dt. \end{aligned}$$

Then, requiring $w(x,T) = 0$, applying boundary conditions for $u$ and $v$, and further integrating by parts, we have

$$\begin{aligned} &\int\int_{U_T} z\left[-\partial_t w - \nabla^2 w - qw\right] dxdt \\ &+ \int_0^T \int_{\Gamma^D} w(\nabla v - \nabla u)\cdot n ds dt + \int_0^T \int_{\Gamma^N} (u-v)\nabla w \cdot n ds dt \\ = &\int\int_{U_T} w\tau v dx dt. \end{aligned} \tag{10.3}$$

## 10.2. The Adjoint Problem

Now suppose that $w(x,t)$ solves the associated adjoint problem:

$$\begin{aligned} -\partial_t w - \nabla^2 w - qw &= 0, \quad (x,t) \in U_T \\ w(x,T) &= 0, \quad x \in U \\ w(x,t) &= g^*(x,t), \quad (x,t) \in \Gamma_T^D \\ \nabla w(x,t)\cdot n &= h^*(x,t), \quad (x,t) \in \Gamma_T^N. \end{aligned}$$

Then the integral identity (10.3) can be rewritten as

$$\begin{aligned} &\int_0^T \int_{\Gamma^D} g^*(\nabla v - \nabla u)\cdot n ds dt + \int_0^T \int_{\Gamma^N} (u-v) h^* ds dt \\ = &\int\int_{U_T} w\tau v dx dt. \end{aligned} \tag{10.4}$$

At this point, we can choose adjoint inputs $g^*$ and $h^*$ to specify (10.4) to an expression relating the coefficient differences to various observation differences:

(1) With $h^* = 0$, (10.4) reduces to

$$\int_0^T \int_{\Gamma^D} g^*(\nabla v - \nabla u) \cdot n ds dt = \int \int_{U_T} w \tau v dx dt.$$

This can be rewritten as

$$(g^*, G_N(q_0) - G_N(q))_{L^2(\Gamma_T^D)} = \left(\tau, \int_0^T w v dt\right)_{L^2(U)}, \tag{10.5}$$

where $G_N(q) = \nabla u(x,t,q) \cdot n, \quad (x,t) \in \Gamma_T^D$.

(2) With $g^* = 0$, (10.4) reduces to

$$\int_0^T \int_{\Gamma^N} (u - v) h^* ds dt = \int \int_{U_T} w \tau v dx dt.$$

This can be rewritten as

$$(h^*, G_D(q) - G_D(q_0))_{L^2(\Gamma_T^N)} = \left(\tau, \int_0^T w v dt\right)_{L^2(U)}, \tag{10.6}$$

where $G_D(q) = u(x,t,q), \quad (x,t) \in \Gamma_T^N$.

Note that, as in our previous problem, (10.5) and (10.6) involve equations that relate coefficient differences to observation differences. We can make observations

on only one portion of the boundary or its entirety, at our option. In the latter case, we can rewrite (10.4) as

$$(g^*, \Delta G_N)_{L^2(\Gamma_T^D)} + (h^*, \Delta G_D)_{L^2(\Gamma_T^N)} = \left(\tau, \int_0^T wvdt\right)_{L^2(U)}.$$

We note that this is a well posed initial boundary value problem, given that the coefficient $q \in L^\infty(U)$ is such that $q(x) \leq k_2 < 0$ for $x \in U$. Then this has a unique solution $u = u(x,t,q)$ such that $u \in L^2[(0,T) : H^1(U)] \cap C[[0,T] : H^0(U)]$, with $f_D(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^D\right)\right]$ and $f_N(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$.

## 10.3. The Spatial Setting

The function space setting of the $q(x)$ problem closely resembles that associated with the previous identification, detailed in Section 3.3. Beyond the similarities, we simply note the lack of gradients in the integral $\int_0^T wvdt$ relaxes the setting constraints from previously. Now $w, v \in H^s(U)$ implies $\int_0^T wvdt \in H^s(U)$ for $s \geq 0$, allowing us to consider $\tau \in H^0(U)$.

$$\begin{aligned} &\left\langle \tau, \int_0^T wvdt \right\rangle_{H^S(U) \times H^{-S}(U)} \\ = &\int_0^T \left[ \langle g^*, \Delta G_N \rangle_{H^{1/2}(\Gamma^D) \times H^{-1/2}(\Gamma^D)} + \langle h^*, \Delta G_D \rangle_{H^{-1/2}(\Gamma^N) \times H^{1/2}(\Gamma^N)} \right] dt, \quad s \geq 0. \end{aligned}$$

## 10.4. Identifiability

We consider identifying the unknown coefficient $q(x)$ in the initial (mixed) boundary value problem (10.1). The proof that follows can be generalized to Dirichlet and Neumann conditions. The construction closely follows that given in Section 3.5, but in n spatial dimensions due to its reduced complexity.

If we suppose the data $f_D$ and $f_N$ are fixed and let the dependence of the solution on the coefficient $q$ be denoted by writing $u = u(x,t,q)$, then we can denote associated observation operators,

$$\begin{aligned} G_D[q] &= \nabla u(x,t,q) \cdot n, \quad (x,t) \in \Gamma_T^D, \\ G_N[q] &= u(x,t,q), \quad (x,t) \in \Gamma_T^N. \end{aligned}$$

These operators map $q \in L^\infty(U)$ to $L^2(\Gamma_T)$. Note that neither $G_D[0]$ nor $G_N[0]$ is zero, so neither is linear; however, both operators will be shown to be injective; i.e. the coefficient $q$ is identifiable from (either of) these observations. Note that the only point to considering $\Gamma$ to be composed of complementeary pieces is to admit the possiblity of mixed boundary conditions and, more importantly, to allow consideration of the case in which observations are made over only part of the boundary.

We note that identifiability in this problem can be shown by the method of Carleman estimates or by the ad-hoc scheme described in [**1**]. Here we provide an alternate method which employs the adjoint techniques used in the recovery process detailed in Chapter 4, and more importantly, can be modified to the case where $q$ is state dependent as well: $q = q(u)$.

We let $v = v(x, t, q_0)$ solve the same (comparison) IBVP with coefficient $q_0 = q_0(x)$, allowing the more general case of spatial dependence. We again denote $q(x) = q_0(x) + \tau(x)$. We take the difference of the direct and comparison problems, and denote the solution to this difference IBVP as $z = u - v$. We then multiply this PDE by the smooth function $w(x, t)$ which solves the adjoint problem

$$\begin{aligned} -\partial_t w - \nabla^2 w - qw &= 0, \qquad (10.7) \\ w(x, T) &= 0, \\ w(x, t) &= g^*(x, t), \quad (x, t) \in \Gamma_T^D \\ \nabla w(x, t) \cdot n &= h^*(x, t), \quad (x, t) \in \Gamma_T^N. \end{aligned}$$

Then, after integration by parts, we have the following identity, written in terms of the observation operators.

$$\int\int_{U_T} w\tau v dx dt \tag{10.8}$$

$$= \int_0^T \int_{\Gamma^D} g^*(x,t)\left(G_D[q_0] - G_D[q]\right) ds dt$$

$$+ \int_0^T \int_{\Gamma^N} \left(G_N[q] - G_N[q_0]\right) h^*(x,t)\, ds dt$$

In order for the operation operators to be injective, it will be necessary to impose some sort of conditions on the boundary inputs, $f_D$ and $f_N$ for the direct problem. For example, if the boundary input were such that the solution $v$ were to vanish on some positive measure subset of $U_T$, this would render the coefficient $q$ arbitrary on this set without influencing the observation values. It is a simple matter to impose sufficient conditions experimentally. For example, choosing

$$f_D \in C^1\left[\Gamma^D \times (0,T)\right],$$

$$f_N \in C^1\left[\Gamma^N \times (0,T)\right]$$

such that

$$\partial_t f_D(x,t) > 0, \tag{10.9}$$

$$f_N(x,t) < 0$$

implies, via maximum/minimum principles that $v = (x, t, q_0)$ does not vanish in this fashion [7]. We now prove that $G_D$ is injective; the proof for $G_N$ is similar.

**Theorem 7.** *Suppose that $f_D$ and $f_N$ satisfy (10.9) and that the coefficients $q_0, q_1 \in L^\infty(U)$ are distinct in the sense that there exists some open set $\Omega \subset U$ where $\tau(x) = q_1(x) - q_0(x) \geq q_* > 0$ almost everywhere in $\Omega$. Then $G_D[q_1] \neq G_D[q_0]$.*

**Proof.** Suppose that $G_D[q_1] = G_D[q_0]$ and choose the adjoint input $h^* = 0$. Clearly, if $w$ solves (10.7), then $\partial_t w$ solves the same with $g^*$ replace by $\partial_t g^*$. Hence it follows from (10.8) that

$$I = \int \int_{U_T} \tau(x)\, \partial_t w(x,t)\, v(x,t) dx dt = 0$$

for every $g^* \in C^1\left[0T : H^{1/2}\left(\Gamma^D\right)\right]$. Using the fact that $v$ is the solution to a parabolic problem and is therefore smooth on the interior of the solution domain, we can apply a mean value theorem to conclude that for some $t^* \in (0, T)$ we have

$$\begin{aligned} I &= \int_U \tau(x)\, v(x,t^*) \left[\int_0^T \partial_t w(x,t)\, dt\right] dx \qquad (10.10) \\ &= \int_U \tau(x)\, v(x,t^*) \left[w(x,T) - w(x,0)\right] dx \\ &= -\int_U \tau(x)\, v(x,t^*) w(x,0)\, dx = 0. \end{aligned}$$

We know the forward parabolic direct problem is approximately controllable per [**19, 20**], so under the change of variable $\tilde{t} = T - t$, the adjoint problem is "approximately backwards controllable", which is to say that the adjoint input $g^*$ can be chosen to guide $w(x,t)$ from the initial state at $t = T$ toward an arbitrary target function in $L^2(U)$ at $t = 0$. Choosing the target function to be the characteristic function on $\Omega$, $\chi_\Omega(x)$, it follows that for any $\varepsilon > 0$ there is a control $g^*$ such that $||w(x,0) - \chi_\Omega(x)||_{L^2(\Omega)} < \varepsilon$. Then the assumptions made on $\tau$ and on the data $f_D$ and $f_N$ ensure that there exist positive constants $M, m$ such that $M \geq \tau(x)\, v(x,t^*) \geq m > 0, \quad x \in \Omega$. Then

$$\begin{aligned} & \int_U \tau(x)\, v(x,t^*) w(x,0)\, dx \\ = & \int_U \tau(x)\, v(x,t^*) \left[w(x,0) - \chi_\Omega(x) + \chi_\Omega(x)\right] dx \\ = & \int_U \tau(x)\, v(x,t^*) \left[w(x,0) - \chi_\Omega(x)\right] dx + \int_\Omega \tau(x)\, v(x,t^*) dx. \end{aligned}$$

The goal at this point is to show that this expression must be non-zero. With this in mind, we must take into account that $v(x,t^*)$ may be of different signs on $\Omega$. If this is the case, we merely use the characteristic function on a smaller set $\hat{\Omega} \subset \Omega$, such that $v(x,t^*) > 0$ or $v(x,t^*) < 0\ \forall x \in \hat{\Omega}$. Assuming $v(x,t^*) > 0\ \forall x \in \Omega$, we

have that

$$\begin{aligned} & \int_U \tau(x)\, v(x,t^*)\left[w(x,0) - \chi_\Omega(x)\right] dx + \int_\Omega \tau(x)\, v(x,t^*) dx \\ \geq\ & -(M\varepsilon)\, meas(U) + (m)\, meas(U) \\ =\ & (m - M\varepsilon)\, meas(U) \end{aligned}$$

Then for $\varepsilon$ sufficiently small, $(m - M\varepsilon)\, meas(U) > 0$. Then we have that $I < 0$, contradicting (10.10); this shows that $G_D[q_1] = G_D[q_0]$ is not consistent with $\tau(x) > 0$. We can also show that $v(x,t^*) < 0\ \forall x \in \Omega$ leads to $I > 0$, and thus achieve the necessary inconsistency in this case as well. A similar argument shows $G_N[q_1] \neq G_N[q_0]$. □

We end this section by noting the increasing requirements on the terms in the initial boundary value problems. First, we ensure unique solutions by requiring admissible coefficients (bounded away from zero) to be in $L^\infty(U)$ and the boundary terms $f_D(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^D\right)\right]$ and $f_N(x,t) \in L^2\left[(0,T) : H^s\left(\Gamma^N\right)\right]$. To guarantee identifiability of the coefficient via the observation operators noted, we must ensure the solution does not vanish on any measureable subset of the domain, rendering the coefficient values arbitrary on this set. This is accomplished by further controlling the boundary terms in some appropriate fashion, such as the

sufficient conditions in (10.9). Finally, to achieve meaningful integrals in the identities developed in our adjoint methods, we have shown that the coefficient must have further regularity, i.e. must be an elements of $H^s(U)$ for $s \geq 0$. As before, this proves overly restrictive, since it can be shown using semigroup theory that the solutions to our IBVPs are much smoother than indicated [7]. This translates to $\int_0^T wvdt$ being of greater regularity, which in turn allows $\tau$ to be of lesser regularity and still recoverable by the developed techniques.

### 10.5. One-Dimensional Formulation

As in Section 3.4, we now specify the above formulation for $n = 1$. As such, we consider the following differential equation with homogeneous initial condition.

$$\begin{aligned} \partial_t u(x,t) &= \partial_{xx} u(x,t) + q(x)\, u(x,t), \quad (x,t) \in U_T \\ u(x,0) &= 0, \quad x \in U \end{aligned}$$

As before, we'll allow the type of boundary conditions to vary according to experimental preference, and will specify the functions at a later point. Again, we'll impose either Dirichlet, Neumann, or a mix of conditions on the boundary. Our comparison problem solves

$$\partial_t u(x,t) = \partial_{xx} u(x,t) + q_0 u(x,t), \quad (x,t) \in U_T,$$

where $q_0$ is constant. With $\tau(x)$ and $z(x,t)$ defined as in (10.2), we have the following difference equation.

$$\partial_t z - \partial_{xx} z - qz = \tau v$$

Of course, the associated initial and boundary conditions to this equation are homogeneous and specified below.

We multiply the difference equation above by a smooth function $w(x,t)$

$$\int\int_{U_T} w\partial_t z - \int\int_{U_T} w\partial_{xx} z - \int\int_{U_T} wqz = \int\int_{U_T} w\tau v$$

and integrate by parts. Applying a homogeneous final condition on $w$ leads to the following equation.

$$\begin{aligned} &\int\int_{U_T} z\left(-\partial_t w\partial_{xx} w - qw\right) dxdt \\ &- \int_0^T \left[w(1,t)\,\partial_x z(1,t) - w(0,t)\,\partial_x z(0,t)\right] dt \\ &+ \int_0^T \left[\partial_x w(1,t)\, z(1,t) - \partial_x w(0,t)\, z(0,t)\right] dt \\ = &\int\int_{U_T} \tau w v dxdt \end{aligned} \tag{10.11}$$

Specific boundary conditions now necessitate separate analysis for each case: mixed, Dirichlet, and Neumann. In all cases, it is again useful to note that the imposed boundary conditions in the direct problem have the effect of fixing the observation operators, and determine the type of boundary conditions present in the associated adjoint problem. These effects are detailed below.

### 10.5.1. Mixed Boundary Conditions

We first choose mixed conditions for the direct problem, specifically a Dirichlet condition at $x = 0$ and Neumann at $x = 1$: $B_0[u] = u(0,t)$, $B_1[u] = \partial_x u(1,t)$. Then $z(0,t) = \partial_x z(0,t) = 0$, and (10.11) reduces to

$$\begin{aligned} &\int\int_{U_T} z\left(-\partial_t w \partial_{xx} w - qw\right) dxdt \\ &+ \int_0^T w(0,t)\left[\partial_x u(0,t) - \partial_x v(0,t)\right] dt + \int_0^T \partial_x w(1,t)\left[u(1,t) - v(1,t)\right] dt \\ = &\int\int_{U_T} \tau w v dxdt \end{aligned}$$

Now if we choose $w(x,t)$ such that it solves the associated adjoint problem,

$$-\partial_t w - \partial_{xx} w - qw = 0,$$

$$w(x,T) = 0,$$

$$B_0[w] = w(0,t) = g^*(t),$$

$$B_1[w] = \partial_x w(1,t) = h^*(t),$$

then we have the identity

$$\int_0^T B_0[w](G_0[q] - G_0[q_0])\,dt + \int_0^T B_1[w](G_1[q] - G_1[q_0])\,dt$$
$$= \int\int_{U_T} \tau w v dx dt.$$

Our observation operators are $G_0[q] = \partial_x u(0,t)$ and $G_1[q] = u(1,t)$.

### 10.5.2. Dirichlet Boundary Conditions

Next we choose Dirichlet conditions on the boundary: $B_0[u] = u(0,t)$, $B_1[u] = u(1,t)$. The associated adjoint problem used is

$$-\partial_t w - \partial_{xx} w - qw = 0,$$

$$w(x,T) = 0,$$

$$B_0[w] = w(0,t) = g^*(t),$$

$$B_1[w] = w(1,t) = h^*(t),$$

which gives integral identity

$$\begin{aligned} &\int_0^T B_0[w](G_0[q]-G_0[q_0])\,dt+\int_0^T B_1[w](G_1[q_0]-G_1[q])\,dt \\ =&\int\int_{U_T}\tau w v dx dt. \end{aligned}$$

Note that $G_0[q]=\partial_x u(0,t)$ and $G_1[q]=\partial_x u(1,t)$.

### 10.5.3. Neumann Boundary Conditions

We now impose Neumann conditions at both boundaries: $B_0[u]=\partial_x u(0,t)$, $B_1[u]=\partial_x u(1,t)$. The constructed adjoint problem used is

$$-\partial_t w-\partial_{xx}w-qw=0,$$

$$w(x,T)=0,$$

$$B_0[w]=\partial_x w(0,t)=g^*(t),$$

$$B_1[w]=\partial_x w(1,t)=h^*(t),$$

which gives the integral identity

$$\begin{aligned} &\int_0^T B_0[w](G_0[q_0]-G_0[q])\,dt+\int_0^T B_1[w](G_1[q]-G_1[q_0])\,dt \\ =&\int\int_{U_T}\tau w v dx dt \end{aligned}$$

This time, $G_0[q] = u(0,t)$ and $G_1[q] = u(1,t)$.

We can generalize with the identity

$$\int_0^T B_0 \Delta G_0 dt + \int_0^T B_1 \Delta G_1 dt = \int\int_{U_T} \tau w v dx dt,$$

and have the freedom to make observations on all or just part of the boundary.. Setting $g^* = 0$ or $h^* = 0$ omits the need for certain data:

$$\begin{aligned} \int_0^T B_0[w] \Delta G_0 dt &= \int\int_{U_T} \tau w v dx dt, \\ \int_0^T B_1[w] \Delta G_1 dt &= \int\int_{U_T} \tau w v dx dt. \end{aligned}$$

Finally, we note that the Backus-Gilbert recovery method outlined in Chapter 4 is applicable for this problem as well, with just one notational difference:

$$K_j = \int_0^T w_j v dt.$$

The computational aspects of this recovery are nearly identical to those in the previous identification, with numerical differences where appropriate. The results from subsequent recoveries are omitted here, as they produced similar results.

CHAPTER 11

# TWO SPATIAL DIMENSIONS

In Chapter 4, we outlined our modified Backus-Gilbert algorithm in general, i.e. for $n$ spatial dimensions. However, to this point we've performed all computations with a single dimension, in the context of groundwater flow through a column of soil with spatially dependent diffusivity, or hydraulic conductivity. A natural extension is to consider a two dimensional region $U$, such as the unit square (for simplicity), with boundary $\Gamma$,

$$\begin{aligned} U &= (0,1) \times (0,1) \\ \Gamma &= (x,0) \cup (x,1) \cup (0,y) \cup (1,y), \end{aligned}$$

with $U_T$ and $\Gamma_T$ defined as in (3.1).

Here we merely and briefly highlight the major computational differences with implementing the algorithm, leaving the implementation itself as future work.

As an example, we can designate $\Gamma^D$ and $\Gamma^N$ as

$$\begin{aligned}\Gamma^D &= (0,y)\cup(1,y),\\ \Gamma^N &= (x,0)\cup(x,1),\end{aligned}$$

controlling the state on $\Gamma^D$ and the flow on $\Gamma^N$ to generate the parabolic IBVP with mixed boundary conditions:

$$\begin{aligned}\partial_t u(x,y,t) &= \nabla\cdot[a(x,y)\nabla u(x,y,t)], \quad (x,y,t)\in U_T\\ u(x,y,0) &= 0, \quad (x,y)\in U\\ u(x,y,t) &= f_D(x,y,t) \quad (x,y,t)\in\Gamma_T^D\\ a(x,y)\nabla u(x,y,t)\cdot n &= f_N(x,y,t), \quad (x,y,t)\in\Gamma_T^N.\end{aligned}$$

We first mention that the generation of observation data,

$$\begin{aligned}a(x,y)\nabla u(x,y,t)\cdot n, \quad (x,y,t) &\in \Gamma^D,\\ u(x,y,t), \quad (x,y,t) &\in \Gamma^N,\end{aligned}$$

in the theoretical setting requires a slightly more complex numerical approximation in two dimensions, using finite differences or elements. The same is true for

the approximation of the adjoint solution used to compute the adjoint gradients $\nabla w_j(x,y)$ which are in turn used to determine the spatial functions

$$K_j(x,y) = \int_0^T \nabla w_j(x,y) \cdot \nabla v(x,y)\, dt.$$

More significant computational complexities arise with the use of the eigenfunctions and eigenvalues, which are now doubly indexed. Generating a basis for $L^2(U)$ using the conditions

$$u_y(x,0) = u_y(x,1) = u(0,y) = u(1,y) = 0$$

and Sturm-Louiville problem

$$-\nabla^2 \Phi_{mn}(x,y) = \lambda_{mn} \Phi_{mn}(x,y)$$

leads to the eigenpairs

$$\begin{aligned} \Phi_{mn}(x,y) &= 2\sin(m\pi x)\cos(n\pi y), \\ \lambda_{mn} &= \left(m^2+n^2\right)\pi^2, \end{aligned} \tag{11.1}$$

where $m \in \mathbb{N}$ and $n \in \mathbb{N} \cup \{0\}$. Alternate bases can be used, with similar advantages and disadvantages noted in the one dimensional computations (Section

7.2). For the boundary conditions $u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0$,

$$\Phi_{mn}(x,y) = 2\sin(m\pi x)\sin(n\pi y),$$

$$\lambda_{mn} = \left(m^2 + n^2\right)\pi^2, \quad m,n \in \mathbb{N}$$

For $\partial_y u(x,0) = \partial_y u(x,1) = \partial_x u(0,y) = \partial_x u(1,y) = 0$,

$$\Phi_{mn}(x,y) = 2\cos(m\pi x)\cos(n\pi y),$$

$$\lambda_{mn} = \left(m^2 + n^2\right)\pi^2, \quad m,n \in \mathbb{N} \cup \{0\}.$$

For $\partial_y u(x,0) = u(x,1) = u(0,y) = \partial_x u(1,y) = 0$,

$$\Phi_{mn}(x,y) = 2\sin\left[\left(m - \frac{1}{2}\right)\pi x\right]\cos\left[\left(n - \frac{1}{2}\right)\pi x\right],$$

$$\lambda_{mn} = \left[\left(m - \frac{1}{2}\right)^2 + \left(n - \frac{1}{2}\right)^2\right]\pi^2, \quad m,n \in \mathbb{N}.$$

The generation of these pairs are detailed in Appendix 4.2.

Using the eigenpairs (11.1), we compute the inner products associated with the linear system $K\Lambda = \delta$ (4.4):

$$(K_i, K_j)_{-s} = \sum_{m=1}^{\infty}\sum_{n=0}^{\infty} \lambda_{mn}^{-s}(K_i)_{mn}(K_j)_{mn}.$$

The applicable Fourier coefficients are computed via the double integrals

$$(K_i)_{mn} = \int\int_U K_i(x,y)\,\Phi_{mn}(x,y)\,dxdy.$$

Of course, we'll approximate these inner products, necessitating a scheme to truncate the double summation appropriately. Similarly, with the vector $\delta$, we compute

$$(K_i, \delta_{x_0y_0})_{-s} = \sum_{m=1}^{\infty}\sum_{n=0}^{\infty} \lambda_{mn}^{-s}\,(K_i)_{mn}\,(\delta_{x_0y_0})_{mn},$$

where

$$(\delta_{x_0y_0})_{mn} = \int\int_U \delta_{x_0y_0}\Phi_{mn}(x,y)\,dxdy = \Phi_{mn}(x_0,y_0).$$

Once the matrix $K$ and vector $\delta$ are determined, the remaining process is identical to the single dimension system; moreover, the recovery can be again be refined without recomputing $K$, as it's independent of the recovery point $z_0 = (x_0, y_0)$.

CHAPTER 12

# MULTIPLE COEFFICIENTS

A more theoretically challenging variation of the method constructed in Chapter 4 is the simultaneous identification of two unknown coefficients. This was first detailed in [**5**]; we modify it here for efficiency. Specifically, we recover the coefficients $s(x)$ and $a(x)$ in equation (1.1).

$$s(x)\,\partial_t u(x,t) + \nabla \cdot [a(x)\,\nabla u(x,t)] = f(x,t), \qquad (x,t) \in U_T$$

In the context of groundwater flow, we now have a variable storage coefficient $s(x)$, along with the coefficient for hydraulic conductivity, $a(x)$. We choose mixed conditions and denote the following initial and boundary conditions for the direct problem.

$$\begin{aligned} u(x,0) &= u_0 \text{ in } U \\ u(x,t) &= f_D(x,t) \text{ on } \Gamma_T^D \\ a(x)\,\nabla u(x,t) \cdot n &= f_N(x,t) \text{ on } \Gamma_T^N \end{aligned}$$

The comparison solution $v(x,t)$ again solves the direct problem, this time with constant coefficients $s_0$ and $a_0$. With

$$\begin{aligned} z(x,t) &= u(x,t) - v(x,t), \\ \tau(x) &= a(x) - a_0, \\ \sigma(x) &= s(x) - s_0, \end{aligned}$$

we have the following "difference problem".

$$s\partial_t u - (s-\sigma)\partial_t v + \nabla \cdot (a\nabla u) - \nabla \cdot [(a-\tau)\nabla v] = 0$$

$$\Rightarrow s\partial_t z + \nabla \cdot (a\nabla z) = -\nabla \cdot (\tau \nabla v) - \sigma \partial_t v \tag{12.1}$$

$$u(x,0) = 0 \text{ in } U$$

$$z = 0 \text{ on } \Gamma_T^D$$

$$a\nabla u \cdot n - a_0 \nabla v \cdot n = 0 \text{ on } \Gamma_T^N$$

Multiplying (12.1) by the smooth function $w(x,t)$ and integrating, we have the identity (12.2).

$$\int\int_{U_T} w\left[s\partial_t z + \nabla \cdot (a\nabla z)\right] dxdt = \int\int_{U_T} w\left[-\nabla \cdot (\tau \nabla v) - \sigma \partial_t v\right] dxdt \tag{12.2}$$

Integrating by parts, we note that

$$\int\int_{U_T} ws\partial_t z dxdt = -\int\int_{U_T} zs\partial_t w dxdt,$$

if we consider $w(x,T) = 0$. Similarly,

$$\begin{aligned} & \int\int_{U_T} w\nabla\cdot(a\nabla z)\,dxdt \\ = & \int\int_{\Gamma_T} wa\nabla z\cdot n ds dt - \int\int_{\Gamma_T^N} zaw\nabla\cdot n ds dt \\ & + \int\int_{U_T} z\nabla\cdot(a\nabla w)\,dxdt, \end{aligned}$$

and

$$\begin{aligned} & \int\int_{U_T} w\nabla\cdot(\tau\nabla v)\,dxdt \\ = & \int\int_{\Gamma_T} w\tau\nabla v\cdot n ds dt - \int\int_{U_T} \tau\nabla w\cdot\nabla v dxdt. \end{aligned}$$

Then (12.2) can be rewritten as

$$\begin{aligned} & -\int\int_{U_T} zs\partial_t w dxdt \\ & + \int\int_{\Gamma_T} wa\nabla z\cdot n ds dt - \int\int_{\Gamma_T^N} zaw\nabla\cdot n ds dt + \int\int_{U_T} z\nabla\cdot(a\nabla w)\,dxdt \\ = & -\int\int_{\Gamma_T} w\tau\nabla v\cdot n ds dt + \int\int_{U_T} \tau\nabla w\cdot\nabla v dxdt - \int\int_{U_T} w\sigma\partial_t v dxdt \end{aligned}$$

After another round in integration by parts, this simplifies to

$$\begin{aligned} & \int\int_{U_T} z\left[-s\partial_t v + \nabla\cdot(a\nabla w)\right] dxdt \\ & + \int\int_{\Gamma_T^D} w\left(a\nabla u - a_0\nabla v\right)\cdot n ds dt - \int\int_{\Gamma_T^N} za\nabla w\cdot n ds dt \\ = & \int\int_{U_T} \left[\tau\nabla w\cdot\nabla v - w\sigma\partial_t v\right] dxdt. \end{aligned} \tag{12.3}$$

We now require the function $w(x,t)$ to satisfy the following adjoint problem.

$$\begin{aligned} -s\partial_t v + \nabla\cdot(a\nabla w) &= F^*(x,t) \text{ on } U_T \\ w(x,T) &= 0 \text{ on } U \\ w(x,t) &= G^*(x,t) \text{ on } \Gamma_T^D \\ a\nabla w\cdot n &= H^*(x,t) \text{ on } \Gamma_T^N. \end{aligned}$$

Then (12.3) can be written as

$$\begin{aligned} & \int\int_{U_T} zF^* dxdt \\ & + \int\int_{\Gamma_T^D} G^*\left(a\nabla u - a_0\nabla v\right)\cdot n ds dt - \int\int_{\Gamma_T^N} zH^* ds dt \\ = & \int\int_{U_T} \left[\tau\nabla w\cdot\nabla v - w\sigma\partial_t v\right] dxdt. \end{aligned}$$

If we choose $F^*(x,t) = 0$, and choose two sets of $N$ linearly independent inputs $G_j^*$ and $H_j^*$, then (12.3) further simplifies to

$$\begin{aligned} & \int\int_{\Gamma_T^D} G_j^* \left(a\nabla u - a_0 \nabla v\right) \cdot n ds dt + \int\int_{\Gamma_T^N} \left(v-u\right) H_j^* ds dt \qquad (12.4) \\ = & \int\int_{U_T} \left[\tau \nabla w_j \cdot \nabla v - w_j \sigma \partial_t v\right] dx dt. \end{aligned}$$

Similar to our previous work, we denote

$$\begin{aligned} \mu_j &= \int\int_{\Gamma_T^D} G_j^* \left(a\nabla u - a_0 \nabla v\right) \cdot n ds dt + \int\int_{\Gamma_T^N} \left(v-u\right) H_j^* ds dt, \\ K_\sigma^j &= -\int_0^T w_j \partial_t v dt, \\ K_\tau^j &= \int_0^T \nabla w_j \cdot \nabla v dt. \end{aligned}$$

We then rewrite (12.4) as

$$\begin{aligned} \mu_j &= \int_U \left[\sigma(x)\left(-\int_0^T w_j \partial_t v dt\right) + \tau(x)\int_0^T \nabla w_j \cdot \nabla v dt\right] dx \\ &= \int_U \left(\sigma K_\sigma^j + \tau K_\tau^j\right) dx \\ &= \int_U \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} \sigma \\ \tau \end{bmatrix} dx. \end{aligned}$$

We employ the Backus-Gilbert ansatz, this time in vector form and with $x_0 \in U$:

$$\begin{aligned}
\begin{bmatrix} \sigma(x_0) \\ \tau(x_0) \end{bmatrix} &= \sum_{j=1}^{N} \mu_j \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \\
&= \sum_{j=1}^{N} \int_U \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} \sigma \\ \tau \end{bmatrix} dx \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \\
&= \int_U \left( \sum_{j=1}^{N} \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right) \begin{bmatrix} \sigma \\ \tau \end{bmatrix} dx
\end{aligned}$$

This implies that

$$\sum_{j=1}^{N} \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} = \delta(x - x_0). \tag{12.5}$$

Following the same template as the single coefficient recovery from Chapter 4, we consider the mapping $A : H^s(U) \to \{\mathbb{R}_2\}_N$, which takes the arbitrary element $x \in H^s(U)$ to a set of $N$ row vectors.

$$Ax = \left\{ \left[ \int_U x K_\sigma^j dx, \int_U x K_\tau^j dx \right] \right\}_{j=1}^{N}$$

We explicitly define the inner product of elements in $\{\mathbb{R}_2\}_N$ by

$$\begin{aligned}\{a_j\}\cdot\{b_j\} &= \{a_1,\ldots,a_n\}\cdot\{b_1,\ldots,b_n\}\\ &= \sum_{j=1}^{N} a_j\cdot b_j,\end{aligned}$$

where $a_j \cdot b_j$ is the standard $\mathbb{R}_2$ inner product.

We note that $A^T$ must map from $\{\mathbb{R}^2\}_N$ to $H^{-s}(U)$, where $\{\mathbb{R}^2\}_N$ denotes a space whose elements consist of $N$ column vectors. Taking the set of vectors $\{b\} \in \{\mathbb{R}^2\}_N$, we note the following.

$$\begin{aligned}&(Ax,\{b\})_{\{\mathbb{R}^2\}_N}\\ &= \left[b_1^1, b_2^1\right]\begin{bmatrix}\int_U xK_\sigma^1 dx\\ \int_U xK_\tau^1 dx\end{bmatrix} + \cdots + \left[b_1^N, b_2^N\right]\begin{bmatrix}\int_U xK_\sigma^N dx\\ \int_U xK_\tau^N dx\end{bmatrix}\end{aligned}$$

$$\begin{aligned}&= \int_U x\sum_{j=1}^{N}\left[K_\sigma^j, K_\tau^j\right]\begin{bmatrix}b_1^j\\ b_2^j\end{bmatrix} dx\\ &= \left\langle x, \sum_{j=1}^{N}\left[K_\sigma^j, K_\tau^j\right]\begin{bmatrix}b_1^j\\ b_2^j\end{bmatrix}\right\rangle_{H^s(U)\times H^{-s}(U)}\end{aligned}$$

Then

$$\begin{aligned} A^T &: \left\{\mathbb{R}^2\right\}_N \to H^{-s}(U), \\ A^T\{b\} &= \sum_{j=1}^{N} \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} b_1^j \\ b_2^j \end{bmatrix}. \end{aligned}$$

We now construct $\left(A^T\right)^* : H^{-s}(U) \to \left\{\mathbb{R}^2\right\}_N$, taking $y \in H^{-s}(U)$.

$$\begin{aligned} &\left(A^T\{b\}, y\right)_{-s} \\ &= \left( \sum_{j=1}^{N} \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} b_1^j \\ b_2^j \end{bmatrix}, y \right)_{-s} \\ &= \left(K_\sigma^1 b_1^1 + K_\tau^1 b_2^1 + \cdots + K_\sigma^N b_1^N + K_\tau^N b_2^N, y\right)_{-s} \end{aligned}$$

$$= \left[b_1^1, b_2^1\right] \begin{bmatrix} \left(K_\sigma^1, y\right)_{-s} \\ \left(K_\tau^1, y\right)_{-s} \end{bmatrix} + \cdots + \left[b_1^N, b_2^N\right] \begin{bmatrix} \left(K_\sigma^N, y\right)_{-s} \\ \left(K_\tau^N, y\right)_{-s} \end{bmatrix}$$

$$= \left( \{b\}, \left\{ \begin{bmatrix} \left(K_\sigma^j, y\right)_{-s} \\ \left(K_\tau^j, y\right)_{-s} \end{bmatrix} \right\} \right)_{\{\mathbb{R}^2\}_N}$$

This gives that

$$\begin{aligned} \left(A^T\right)^* &: H^{-s}(U) \to \left\{\mathbb{R}^2\right\}_N, \\ \left(A^T\right)^* y &= \left\{ \begin{bmatrix} (K_\sigma^j, y)_{-s} \\ (K_\tau^j, y)_{-s} \end{bmatrix} \right\}_{j=1}^N . \end{aligned}$$

With these constructed mappings, we can rewrite (12.5) as

$$A^T \left\{ \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right\} = \delta(x - x_0). \tag{12.6}$$

As with Equation (4.2), we know nothing about the solveability of (12.6). Accordingly, we apply $\left(A^T\right)^*$ to arrive at the normal equation

$$\left(A^T\right)^* A^T \left\{ \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right\} = \left(A^T\right)^* \delta(x - x_0). \tag{12.7}$$

For now, we work with the left side of (12.7). Note that

$$\left(A^T\right)^* A^T \left\{ \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right\} = \left(A^T\right)^* \left( \sum_{j=1}^N \left[K_\sigma^j, K_\tau^j\right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right)$$

$$= \left\{ \left[ \left( K_\sigma^i, \sum_{j=1}^N \left[ K_\sigma^j, K_\tau^j \right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right)_{-s} \right] \right\}_{i=1}^N .$$

We can rewrite each of these inner products:

$$\begin{aligned} & \left( K_\sigma^i, \sum_{j=1}^N \left[ K_\sigma^j, K_\tau^j \right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right)_{-s} \\ = & \sum_{j=1}^N \left[ \Lambda_\sigma^j(x_0) \left( K_\sigma^i, K_\sigma^j \right)_{-s} + \Lambda_\tau^j(x_0) \left( K_\sigma^i, K_\tau^j \right)_{-s} \right] \\ & \left( K_\tau^i, \sum_{j=1}^N \left[ K_\sigma^j, K_\tau^j \right] \begin{bmatrix} \Lambda_\sigma^j(x_0) \\ \Lambda_\tau^j(x_0) \end{bmatrix} \right)_{-s} \\ = & \sum_{j=1}^N \left[ \Lambda_\sigma^j(x_0) \left( K_\tau^i, K_\sigma^j \right)_{-s} + \Lambda_\tau^j(x_0) \left( K_\tau^i, K_\tau^j \right)_{-s} \right]. \end{aligned}$$

Then the left hand side Equation (12.7) can be written as

$$\left\{ \begin{bmatrix} \sum_{j=1}^N \left[ \Lambda_\sigma^j(x_0) \left( K_\sigma^i, K_\sigma^j \right)_{-s} + \Lambda_\tau^j(x_0) \left( K_\sigma^i, K_\tau^j \right)_{-s} \right] \\ \sum_{j=1}^N \left[ \Lambda_\sigma^j(x_0) \left( K_\tau^i, K_\sigma^j \right)_{-s} + \Lambda_\tau^j(x_0) \left( K_\tau^i, K_\tau^j \right)_{-s} \right] \end{bmatrix} \right\}_{i=1}^N ,$$

which can then be written in terms of inner products in $\mathbb{R}^N$, i.e.

$$\sum_{j=1}^{N} \Lambda_\sigma^j (x_0) \left(K_\sigma^i, K_\sigma^j\right)_{-s} = \left[\left(K_\sigma^i, K_\sigma^1\right)_{-s}, \cdots, \left(K_\sigma^i, K_\sigma^N\right)_{-s}\right] \begin{bmatrix} \Lambda_\sigma^1 \\ \vdots \\ \Lambda_\sigma^N \end{bmatrix}.$$

Given this, it is easy to show that the expression can be conveniently rewritten in the matrix vector form

$$\begin{bmatrix} K_{\sigma\sigma} & K_{\sigma\tau} \\ K_{\tau\sigma} & K_{\tau\tau} \end{bmatrix} \begin{bmatrix} \Lambda_\sigma \\ \Lambda_\tau \end{bmatrix},$$

where the square matrix is size $2N$ and has $N \times N$ blocks defined by the following.

$$K_{\sigma\sigma} = \begin{bmatrix} \left(K_\sigma^1, K_\sigma^1\right)_{-s} & \cdots & \left(K_\sigma^1, K_\sigma^N\right)_{-s} \\ \vdots & \ddots & \vdots \\ \left(K_\sigma^N, K_\sigma^1\right)_{-s} & \cdots & \left(K_\sigma^N, K_\sigma^N\right)_{-s} \end{bmatrix}$$

$$K_{\sigma\tau} = \begin{bmatrix} \left(K_\sigma^1, K_\tau^1\right)_{-s} & \cdots & \left(K_\sigma^1, K_\tau^N\right)_{-s} \\ \vdots & \ddots & \vdots \\ \left(K_\sigma^N, K_\tau^1\right)_{-s} & \cdots & \left(K_\sigma^N, K_\tau^N\right)_{-s} \end{bmatrix}$$

$$K_{\tau\sigma} = \begin{bmatrix} \left(K_\tau^1, K_\sigma^1\right)_{-s} & \cdots & \left(K_\tau^1, K_\sigma^N\right)_{-s} \\ \vdots & \ddots & \vdots \\ \left(K_\tau^N, K_\sigma^1\right)_{-s} & \cdots & \left(K_\tau^N, K_\sigma^N\right)_{-s} \end{bmatrix}$$

$$K_{\sigma\sigma} = \begin{bmatrix} \left(K_\tau^1, K_\tau^1\right)_{-s} & \cdots & \left(K_\tau^1, K_\tau^N\right)_{-s} \\ \vdots & \ddots & \vdots \\ \left(K_\tau^N, K_\tau^1\right)_{-s} & \cdots & \left(K_\tau^N, K_\tau^N\right)_{-s} \end{bmatrix}$$

The corresponding vector has size $2N$, with block vectors defined:

$$\Lambda_\sigma = \begin{bmatrix} \Lambda_\sigma^1 \\ \vdots \\ \Lambda_\sigma^N \end{bmatrix}, \quad \Lambda_\tau = \begin{bmatrix} \Lambda_\tau^1 \\ \vdots \\ \Lambda_\tau^N \end{bmatrix}.$$

Finally, we note that

$$\begin{aligned} &\left(A^T\right)^* \delta\left(x - x_0\right) \\ &= \left\{ \begin{bmatrix} \left(K_\sigma^i, \delta\left(x - x_0\right)\right)_{-s} \\ \left(K_\tau^i, \delta\left(x - x_0\right)\right)_{-s} \end{bmatrix} \right\}_{i=1}^{N}. \end{aligned}$$

Then we can write

$$\begin{bmatrix} K_{\sigma\sigma} & K_{\sigma\tau} \\ K_{\tau\sigma} & K_{\tau\tau} \end{bmatrix} \begin{bmatrix} \Lambda_\sigma \\ \Lambda_\tau \end{bmatrix} = \begin{bmatrix} \delta_\sigma \\ \delta_\tau \end{bmatrix},$$

where

$$\delta_\sigma = \begin{bmatrix} \left(K_\sigma^1, \delta(x-x_0)\right)_{-s} \\ \vdots \\ \left(K_\sigma^N, \delta(x-x_0)\right)_{-s} \end{bmatrix},$$

$$\delta_\tau = \begin{bmatrix} \left(K_\tau^1, \delta(x-x_0)\right)_{-s} \\ \vdots \\ \left(K_\tau^N, \delta(x-x_0)\right)_{-s} \end{bmatrix}.$$

In even more compact form, our estimate of the vector coefficients in the Backus-Gilbert ansatz is computed by solving the linear system

$$K\Lambda(x_0) = \delta(x_0),$$

where $K$ is again symmetric positive definite. As in the single coefficient recovery, the matrix $K$ is independent of the point of recovery, $x_0$, and thus need only be computed once and then used repeatedly in the refinement of the spatial recovery.

CHAPTER 13

# CONCLUSIONS AND FUTURE WORK

Approximating the solution $f$ to problem (1.3), $Af = g$, where $A$ is an integral operator of the first kind, and $g_i$ represent data measurements, is an important type of inverse problem. The extremely ill-posed nature of straightforward discretization of the integral equation necessitates some alternative solution approach. The Backus-Gilbert method is one such approach, where the localized $f(x_0)$ is assumed to be a linear combination of the measurements,

$$f(x_0) = \sum_{i=1}^{N} g_i \phi_i,$$

which leads to choosing the functions $\phi_i(x)$ to minimize

$$\int_U \phi_i(x)^2 (x - x_0)^2 \, dx$$

under some constraint such as $\int_U \phi_i(x)^2 \, dx = 1$ [**2, 18**].

In our modified approach presented in Chapter 4, we moved away from this specific minimization problem, and instead proposed the use of an adjoint technique in

an appropriate Hilbert space setting, which resulted in the normal equation (4.3)

$$\left(A^T\right)^* \left[A^T \bar{\Lambda}\left(z_0\right)\right] = \left(A^T\right)^* \left(\delta_{z_0}\right),$$

characterized as the linear system (4.4):

$$K\bar{\Lambda} = \left\{\left(K_i, \delta_{z_0}\right)_{-s}\right\}_{i=1}^{N}.$$

The vector $\bar{\Lambda}$ was then computed in this ill-posed portion of the algorithm, and used to locally construct the coefficent via the data measurements $\mu_j$:

$$\tau\left(z_0\right) = \sum_{j=1}^{N} \Lambda_j\left(z_0\right) \mu_j.$$

First and foremost, the goal of this research was simply to see if the method worked in any sense. To accomplish this, we initially tested (Chapter 5) the validity of direct identification of a constant coefficient – the only case possible – via just the integral identity. Despite foregoing the Backus-Gilbert method above, this determination held value to us, since even in this overly simplistic scenario the recovered coefficent $\tau$ was dependent upon observation data $(\Delta G_0, \Delta G_1)$ and

selected adjoint inputs $(B_0^*, B_1^*)$:

$$\tau = \frac{\int_0^T B_0^* \Delta G_0 dt + \int_0^T B_1^* \Delta G_1 dt}{\int \int_{U_T} \partial_x w \partial_x v dx dt}.$$

The success of our input choices and use of the "measured" data was clear when the unknown coefficient $w = w(a)$ was used in the identification. Using the coefficient to identify itself provided a convenient control to see if these data and inputs were of value computationally. Moreover, when using an approximate adjoint solution in the computation, $\hat{w} = \hat{w}(\hat{a})$, the results were consistent, albeit less accurate. This was our first indication of the effects of this approximation on the recovery process.

To allow for more complex coefficients, we then employed the Backus-Gilbert method to generate the preliminary results of Chapter 7. As with the earlier direct identification, these results clearly indicate the algorithm is theoretically sound, while artificially using the adjoint solution $w = w(a)$ constructed with the true coefficient. Of course, as before, having the unknown coefficient appear in the adjoint problem is the inherent flaw in this adjoint method. Again, however, these preliminary results were of value, as assurance that the local coefficient reconstruction could indeed be based on the data measurements, and to provide

a control while determining the effects of changing parameters (adjoint inputs, eigenfunctions, etc.) in the algorithm.

These preliminary results provided the motivation to implement a predictor-corrector system to deal with the adjoint approximation difficulty. The subsequent results in Chapter 8 show that the iterative process $R(\hat{w}_n) = \hat{w}_{n+1}$, with estimated adjoint solutions $\hat{w}_n = \hat{w}_n(\hat{a}_n)$, can lead to a satisfactory recovery using a proper subspace dimension. In doing so, we performed notably accurate recoveries of a variety of linear and piecewise constant functions, often despite seemingly inaccurate "initial" behavior. Additionally, we noted that the choice of boundary conditions (Dirichlet, Neumann, mixed) did not provide any consistent advantage, in contrast to the choice of subspace dimension.

Because the accurate choice of such a dimension depends on the nature of the coefficient (which is unknown), the experimental determination is a matter of noting and comparing the potential instabilities of the iterative computations among several choices. Reaching a maximum number of iterations in the algorithm is often accompanied by increased oscillatory behavior requiring coefficient truncation. When this occurs at dimension $N = N_0$, we take the recovery associated with dimension $N_0 - 1$ as the best coefficient approximation. This was detailed in Chapter 9, along with other experimental considerations:

- Experiments shorter in duration do not allow for coefficient differences to sufficiently propagate in order to provide adequate distinction upon recovery.
- Increasing the magnitudes of boundary conditions (if feasible) can be used to bring the observation data to a required sensitivity, while keeping the method results intact.
- The ill-posed nature of the inverse problem can lead to great sensitivity to unavoidable errors in data collection.

We reiterate here the main advantages of this method over others mentioned previously:

- The matrix $K$ in Equation (4.4) is independent of both the data and the spatial variable $z_0$, significantly reducing the computational workload involved, and allowing simple recovery refinement as desired.
- As shown in Chapter 7, the method is robust under changes in the selected adjoint data $g^*$ and $h^*$. Various choices made for each greatly changed the functions $K_j(x)$ used in the construction of (4.4), while minimally affecting the subsequent recoveries.

- A limited amount of data can be used in the recovery process: Measurements from a portion of the boundary was shown to be sufficient in certain cases.

We restate that this method is not limited to the single case studied in depth here. Rather, it can be extended to the recoveries in higher spatial dimensions, the recovery of different coefficients (such as that detailed in Chapter 10), as well as the simultaneous recoveries of multiple coefficients.

As with most research concerning new solution techniques, we leave important work for future endeavors, including:

- Controlling the effects of data measurement error: This is critical to the applied success of this method. Incorporating regularization techniques detailed in [**10, 25, 26**] may be necessary.
- Applying the technique to the recovery of coefficients in two spatial dimensions (Chapter 11).
- Applying the technique to the recovery of multiple coefficients in a single PDE (Chapter 12).
- Theoretically justifying the assumed iterative convergence associated with the results in Chapter 8.

# REFERENCES


[1] Avdonin, S. and Seidman, T.I.. Identification of $q(x)$ in $u_t = \Delta u - qu$ from boundary observations. *SIAM J. Control and Optimization.* Vol. 33, No. 4, pp. 1247-1255, July 1995.

[2] Backus, G. and Gilbert, F. The resolving power of gross earth data. *Geophys. J. R. Astron. Soc.* 16, pp. 169-205, 1968.

[3] Chavent, G. and Kunisch, K. The output least squares identifiability of the diffusion coefficient from an $h_1$-observation in a 2-d elliptic equation. Technical Report 4067, *Institut National de Recherche en Information et en Automatique (INRIA)*, 2000.

[4] Chavent, G. and Lemonneir, P. Identification de la non-linearité d'une équation parabolique quasilineaire. *Applied math. & opt.* 1(2):121-161, 1974.

[5] DuChateau, P. and Badran, A. Identification of spatially dependent coefficients in partial differential equations. *Dynamics of Continuous, Discrete and Impulsive Systems* Series A: Mathematical Analysis 10, 2002.

[6] DuChateau, P. and Zachmann, D. *Applied Partial Differential Equations.* Harper & Row, Publishers, Inc., 1989.

[7] Evans, L.C. *Partial Differential Equations.* American Mathematical Society, Providence, Rhode Island, 1991.

[8] Groetsch, C.W. *Inverse Problems in the Mathematical Sciences.* Vieweg, Braunschweig, Wiesbaden, 1993.

[9] Gustafson, K.E. *Introduction to Partial Differential Equations and Hilbert Space Methods.* 3rd Ed., revised, International Journal Services, Inc., Calcutta and Charleston, Ill., 1993.

[10] Hansen, P.C. *Rank-Deficient and Discrete Ill-Posed Problems, Numerical Aspects of Linear Inversion.* SIAM, 1998.

[11] Jafrré, J., Ameur, H.B. and Chavent, G. Refinement and coarsening of parameterization for the estimation of hydraulic transmissivity. *Proceeding of Inverse Problems in Engineering: Theory and Practice*; 3rd International Conference on Inverse Problems in Engineering, 1999.

[12] Kirsch, A., Schomburg, B. and Berendt, G. The Backus-Gilbert method. *Inverse Problems* 4, pp. 771-783, 1988.

[13] Klibanov, M.V. Carleman estimates and inverse problems in the last two decades. *Solution Methods for Inverse Problems,* Springer Verlag, Vienna/New York, 2000.

[14] Knowles, I. and Yan, A. The recovery of an anisotropic conductivity in groundwater modelling. *Applicable Analysis*, 81, pp. 1347–1365, 2002.

[15] Kreyszig, E. *Introductory Functional Analysis with Applications.* John Wiley & Sons, Inc., 1978.

[16] Lamm, P.K., and Scofield, T.L. Sequential predictor-corrector methods for the variable regularization of Volterra inverse problems. *Inverse Problems* 16, pp. 373-399, 2000.

[17] Louis, A.K., and Maab, P. Smoothed projection methods for the moment problem, *Numer. Math* 59, pp. 277-294, 1991.

[18] Maab, P. Generalized Backus-Gilbert methods. MA 6-1, Technische Unversität Berlin, Strabe des 17. Juni135, D-1000 Berlin 12, Germany, preprint.

[19] McCamy, R.C., and Mizel, V.J., Seidman, T.I. Approximate boundary controllability of the heat equation. *Journal of Mathematical Analysis and Applications* 23, pp. 699-703, 1968.

[20] McCamy, R.C., and Mizel, V.J., Seidman, T.I. Approximate boundary controllability of the heat equation II. *Journal of Mathematical Analysis and Applications* 28, pp. 482-492, 1969.

[21] Nabakov, R. An inverse problem for porous medium equation: Identification of the permeability. *Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology*, pp. 155-163, 1996.

[22] Ozisik, M.N., Jarny, Y. and Bardon, J.P. A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *Int. J. Heat Mass Transfer*, 34(11):2911-2918, 1991.

[23] Street, R.L. *The Analysis and Solution of Partial Differential Equations.* Wadsworth Publishing Company, 1973.

[24] Sylvester, J. and Uhlmann, G. The Dirichlet to Neumann map and applications. Inverse problems in partial differential equations. *SIAM* pp. 101–139, 1990

[25] Tikhonov, A.N., and Arsenin, V.Y. *Solutions of Ill-Posed Problems.* John Wiley & Sons, 1977.

[26] Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V., and Yagola, A.G. *Numerical Methods for the Solution of Ill-Posed Problems.* Kluwer Academic Publishers, Volume 328, 1995.

APPENDICES

# ADDITIONAL COMPUTATIONS AND CODE

## 1. Difference Problem Construction

### 1.1. The a(x) Problem

Given that $u(x,t)$ solves

$$\partial_t u = \nabla \cdot [a(x) \nabla u],$$

with given initial and boundary conditions, and $v(x,t)$ solves the same IVBP with $a(x) = a_0 =$ constant, we develop a difference problem with homogeneous conditions. Letting $z = u - v$ and $\tau = a - a_0$, we have that

$$\begin{aligned} \partial_t (u - v) &= \nabla \cdot (a \nabla u - a_0 \nabla v) \\ &= \nabla \cdot (a \nabla u - a \nabla v + a \nabla v - a_0 \nabla v) \\ &= \nabla \cdot [a \nabla (u - v) + (a - a_0) \nabla v]. \end{aligned}$$

Then

$$\partial_t z = \nabla \cdot (a \nabla z + \tau \nabla v) \Rightarrow \partial_t z - \nabla \cdot (a \nabla z) = \nabla \cdot (\tau \nabla v).$$

## 1.2. The q(x) Problem

Given that $u(x,t)$ solves

$$\partial_t u = \nabla^2 u + q(x)\, u,$$

with given initial and boundary conditions, and $v(x,t)$ solves the same IVBP with $q(x) = q_0 = \text{constant}$, we develop a difference problem with homogeneous conditions. Letting $z = u - v$ and $\tau = q - q_0$, we have that

$$\begin{aligned} \partial_t (u - v) &= \nabla^2 (u - v) + qu - q_0 v \\ &= \nabla^2 (u - v) + qu - qv + qv - q_0 v \\ &= \nabla^2 (u - v) + q(u - v) + (q - q_0) v. \end{aligned}$$

Then

$$\partial_t z = \nabla^2 z + qz + \tau v \Rightarrow \partial_t z - \nabla^2 z - qz = \tau v.$$

## 2. Adjoint Construction Computations

As discussed in Section 3.4:

$$\int\int_{U_T} w \partial_t z - \int\int_{U_T} w \partial_x (a \partial_x z) = \int\int_{U_T} w \partial_x (\tau \partial_x v)$$

Applying $w(x,T)=0$, we compute

$$\begin{aligned} & -\int\int_{U_T} z\partial_t w dx dt \qquad (.1) \\ & -\int_0^T \left[a(1)\, w(1,t)\, \partial_x z(1,t) - a(0)\, w(0,t)\, \partial_x z(0,t) - \int_U a\partial_x w \partial_x z dx\right] dt \\ = & \int_0^T \left[\tau(1)\, w(1,t)\, \partial_x v(1,t) - \tau(0)\, w(0,t)\, \partial_x v(0,t) - \int_U \tau\partial_x w \partial_x v dx\right] dt, \end{aligned}$$

and then apply various boundary conditions to construct specific adjoints.

## 2.1. Single Dimension, Mixed

Simplifying (.1) and integrating by parts again, we get

$$\begin{aligned} & -\int\int_{U_T} z\partial_t w dx dt - \int_0^T a(1)\, w(1,t) \left[\partial_x u(1,t) - \partial_x v(1,t)\right] dt \\ & + \int_0^T a(0)\, w(0,t) \left[\partial_x u(0,t) - \partial_x v(0,t)\right] dt \\ & + \int_0^T \left[a(1)\, z(1,t)\, \partial_x w(1,t) - a(0)\, z(0,t)\, \partial_x w(0,t) - \int_U z\partial_x \left(a\partial_x w\right) dx\right] dt \\ = & \int_0^T \left[a(1) - a_0\right] w(1,t)\, \partial_x v(1,t)\, dt - \int_0^T \left[a(0) - a_0\right] w(0,t)\, \partial_x v(0,t)\, dt \\ & - \int\int_{U_T} \tau\partial_x w \partial_x v dx dt. \end{aligned}$$

Regrouping, we arrive at

$$\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dxdt \\
&-\int_0^T w\left(1,t\right)\left[a\left(1\right)\partial_x u\left(1,t\right)-a\left(1\right)\partial_x v\left(1,t\right)\right]dt \\
&+\int_0^T w\left(0,t\right)\left[a\left(0\right)\partial_x u\left(0,t\right)-a\left(0\right)\partial_x v\left(0,t\right)\right]dt \\
&+\int_0^T a\left(1\right)\partial_x w(1,t)\left[u\left(1,t\right)-v\left(1,t\right)\right]dt \\
=\ &\int_0^T\left[a\left(1\right)-a_0\right]w\left(1,t\right)\partial_x v\left(1,t\right)dt-\int_0^T\left[a\left(0\right)-a_0\right]w\left(0,t\right)\partial_x v\left(0,t\right)dt \\
&-\int\int_{U_T}\tau\partial_x w\partial_x v dxdt.
\end{aligned}$$

Rearranging and cancelling terms, we have

$$\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dxdt \\
&+\int_0^T w\left(1,t\right)\left[a_0\partial_x v\left(1,t\right)-a\left(1\right)\partial_x u\left(1,t\right)\right]dt \\
&+\int_0^T w\left(0,t\right)\left[a\left(0\right)\partial_x u\left(0,t\right)-a_0\partial_x v\left(0,t\right)\right]dt \\
&+\int_0^T a\left(1\right)\partial_x w(1,t)\left[u\left(1,t\right)-v\left(1,t\right)\right]dt \\
=\ &-\int\int_{U_T}\tau\partial_x w\partial_x v dxdt.
\end{aligned}$$

Noting the second integral vanishes per the difference conditions, we arrive at the following useful integral identity.

$$\begin{aligned} & \int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dxdt \\ & +\int_0^T w\left(0,t\right)\left[a\left(0\right)\partial_x u\left(0,t\right)-a_0\partial_x v\left(0,t\right)\right]dt \\ & +\int_0^T a\left(1\right)\partial_x w(1,t)\left[u\left(1,t\right)-v\left(1,t\right)\right]dt \\ = & -\int\int_{U_T}\tau\partial_x w\partial_x v dxdt. \end{aligned}$$

Then we use the associated adjoint equation

$$-\partial_t w-\partial_x\left(a\partial_x w\right)=0,$$

with a homogeneous final condition and inputs $w\left(0,t\right)=g^*\left(t\right)$ and $a\left(1\right)\partial_x w\left(1,t\right)=h^*\left(t\right)$.

## 2.2. Single Dimension, Dirichlet

Referring to (.1), a second round of integration by parts gives us

$$-\int\int_{U_T} z\partial_t w dxdt$$

$$
\begin{aligned}
&-\int_0^T \left[a(1)\, w(1,t)\, \partial_x z(1,t) - a(0)\, w(0,t)\, \partial_x z(0,t)\right] dt \\
&+\int_U \left[a(1)\, z(1,t)\, \partial_x w(1,t) - a(0)\, z(0,t)\, \partial_x w(0,t) - \int z \partial_x \left(a \partial_x w\right) dx\right] dt \\
=\ &\int_0^T \left[\tau(1)\, w(1,t)\, \partial_x v(1,t) - \tau(0)\, w(0,t)\, \partial_x v(0,t) - \int_U \tau \partial_x w \partial_x v dx\right] dt.
\end{aligned}
$$

Simplifying, we have that

$$
\begin{aligned}
&-\int\int_{U_T} z \partial_t w dx dt - \int\int_{U_T} z \partial_x \left(a \partial_x w\right) dx dt \\
&-\int_0^T \left[a(1)\, w(1,t)\, \partial_x z(1,t) - a(0)\, w(0,t)\, \partial_x z(0,t)\right] dt \\
=\ &\int_0^T \left[\tau(1)\, w(1,t)\, \partial_x v(1,t) - \tau(0)\, w(0,t)\, \partial_x v(0,t) - \int_U \tau \partial_x w \partial_x v dx\right] dt.
\end{aligned}
$$

Regrouping and further simplifying, we arrive at following useful integral identity.

$$
\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w - \partial_x \left(a \partial_x w\right)\right] dx dt \\
&+\int_0^T w(0,t)\left[a(0)\, \partial_x u(0,t) - a_0 \partial_x v(0,t)\right] dt \\
&+\int_0^T w(1,t)\left[a_0 \partial_x v(1,t) - a(1)\, \partial_x u(1,t)\right] dt \\
=\ &-\int\int_{U_T} \tau \partial_x w \partial_x v dx dt.
\end{aligned}
$$

Then we use the associated adjoint equation

$$-\partial_t w - \partial_x \left(a \partial_x w\right) = 0,$$

with a homogeneous final condition and inputs $w\left(0,t\right) = g^*\left(t\right)$ and $w\left(1,t\right) = h^*\left(t\right)$.

### 2.3. Single Dimension, Neumann

This time, the second integration by parts result is

$$\begin{aligned}
&- \int\int_{U_T} z \partial_t w dx dt - \int\int_{U_T} z \partial_x \left(a \partial_x w\right) dx dt \\
&- \int_0^T \left[a\left(1\right) w\left(1,t\right) \partial_x z\left(1,t\right) - a\left(0\right) w\left(0,t\right) \partial_x z\left(0,t\right)\right] dt \\
&+ \int_0^T \left[a\left(1\right) z\left(1,t\right) \partial_x w\left(1,t\right) - a\left(0\right) z\left(0,t\right) \partial_x w\left(0,t\right)\right] dt \\
= &\int_0^T \left[\tau\left(1\right) w\left(1,t\right) \partial_x v\left(1,t\right) - \tau\left(0\right) w\left(0,t\right) \partial_x v\left(0,t\right)\right] dt \\
&- \int\int_{U_T} \tau \partial_x w \partial_x v dx dt.
\end{aligned}$$

Rearranging and simplifying, we have that

$$\begin{aligned}
&\int\int_{U_T} z \left[-\partial_t w - \partial_x \left(a \partial_x w\right)\right] dx dt \\
&+ \int_0^T w\left(0,t\right) \left[a\left(0\right) \partial_x u\left(0,t\right) - a_0 \partial_x v\left(0,t\right)\right] dt
\end{aligned}$$

$$
\begin{aligned}
&+\int_0^T w(1,t)\left[a_0\partial_x v(1,t)-a(1)\,\partial_x u(1,t)\right]dt \\
&+\int_0^T a(0)\,\partial_x w(0,t)\left[v(0,t)-u(0,t)\right]dt \\
&+\int_0^T a(1)\,\partial_x w(1,t)\left[u(1,t)-v(1,t)\right]dt \\
=\;& -\int\int_{U_T}\tau\partial_x w\partial_x v\,dx\,dt.
\end{aligned}
$$

Noting that two of the integrals vanish, we are left with the following useful integral identity.

$$
\begin{aligned}
&\int\int_{U_T} z\left[-\partial_t w-\partial_x\left(a\partial_x w\right)\right]dx\,dt \\
&+\int_0^T a(0)\,\partial_x w(0,t)\left[v(0,t)-u(0,t)\right]dt \\
&+\int_0^T a(1)\,\partial_x w(1,t)\left[u(1,t)-v(1,t)\right]dt \\
=\;& -\int\int_{U_T}\tau\partial_x w\partial_x v\,dx\,dt.
\end{aligned}
$$

Then we use the associated adjoint equation

$$
-\partial_t w-\partial_x\left(a\partial_x w\right)=0,
$$

with a homogeneous final condition and inputs

$$a(0)\,\partial_x w(0,t) = g^*(t),\ a(1)\,\partial_x w(1,t) = h^*(t).$$

### 3. Lemma integration

We multiply the differential equation (3.12) by $\partial_x \phi(x,t)$, where $\phi = \phi(x,t)$ is a smooth function, and integrate over the domain.

$$\int\int_{U_T} \{\partial_t u(x,t) - \partial_x [a(x)\,\partial_x u(x,t)]\}\,\partial_x \phi(x,t)\,dxdt = 0$$

Integrating by parts, we have that

$$\int_u \left[u\partial_x\phi|_0^T - \int_0^T u\partial_t(\partial_x\phi)\,dt\right]dx - \int_0^T \left[a\partial_x u\partial_x\phi|_0^1 - \int_U a\partial_x u\partial_{xx}\phi dx\right]dt = 0.$$

Now, since $\partial_t(\partial_x\phi) = \partial_x(\partial_t\phi)$,

$$\begin{aligned}\int\int_{U_T} u\partial_t(\partial_x\phi)\,dxdt &= \int\int_{U_T} u\partial_x(\partial_t\phi)\,dxdt \\ &= \int_0^T \left[u\partial_t\phi|_0^1 - \int_U \partial_x u\partial_t\phi dx\right]dt.\end{aligned}$$

Then

$$\int_U u\partial_x\phi|_0^T dx - \int_0^T \left[u\partial_t\phi|_0^1 - \int_U \partial_x u\partial_t\phi dx\right] dt$$
$$- \int_0^T a\partial_x u\partial_x\phi|_0^1 dt + \int\int_{U_T} a\partial_x u\partial_{xx}\phi dx dt = 0,$$

and so

$$\int\int_{U_T} (\partial_t\phi + a\partial_{xx}\phi)\,\partial_x u dx dt + \int_u u\partial_x\phi|_0^T dx$$
$$- \int_0^T u\partial_t\phi|_0^1 dt - \int_0^T a\partial_x u\partial_x\phi|_0^1 dt = 0.$$

## 4. Generating Orthonormal Bases

### 4.1. Single Spatial Dimension

In all cases, we consider $0 < x < 1$.

#### 4.1.1. Dirichlet Conditions.

$$\begin{aligned} -u''(x) &= \lambda u(x) \qquad (.2) \\ u(0) &= u(1) = 0 \end{aligned}$$

It can be shown that the case $\lambda > 0$ leads to the only nontrivial solution to this initial value problem. Since the roots of the auxiliary equation are $r = \pm i\sqrt{\lambda}$, the

real solution is

$$u(x) = c_1 \cos\left(\sqrt{\lambda}x\right) + c_2 \sin\left(\sqrt{\lambda}x\right).$$

Now, $u(0) = 1 \Rightarrow c_1 = 0$. Then

$$\begin{aligned} u(x) &= c_2 \sin\left(\sqrt{\lambda}x\right) \quad \text{and} \\ u(1) &= 0 \Rightarrow c_2 \sin\left(\sqrt{\lambda}\right) = 0 \Rightarrow \\ \sin\left(\sqrt{\lambda}\right) &= 0 \Rightarrow \lambda_n = (n\pi)^2, \quad n \in \mathbb{N}. \end{aligned}$$

Then, choosing $c_2 = \sqrt{2}$, we have

$$u_n(x) = \sqrt{2}\sin(n\pi x), \quad n \in \mathbb{N}.$$

Then the orthonormal set of eigenfunctions, with associated eigenvalues are:

$$\begin{aligned} u_n(x) &= \sqrt{2}\sin(n\pi x) \quad n \in \mathbb{N}, \\ \lambda_n &= (n\pi)^2, \quad n \in \mathbb{N}. \end{aligned} \tag{.3}$$

### 4.1.2. Neumann Conditions.

$$-u''(x) = \lambda u(x)$$

$$u'(0) = u'(1) = 0$$

This time, we can purposely evaluate the case $\lambda = 0$. Clearly,

$$u(x) = c_1 x + c_2 \Rightarrow u'(x) = c_1.$$

Then the initial conditions lead to $c_1 = 0$. Setting $c_2 = 1$, we have that $u_0 = 1$. Now, similar to the previous example, we can show the case $\lambda > 0$ leads to

$$\begin{aligned} \lambda_n &= (n\pi)^2, \quad n \in \mathbb{N} \text{ and} \\ u_n(x) &= \sqrt{2}\cos(n\pi x) \quad n \in \mathbb{N}. \end{aligned} \tag{.4}$$

The orthonormal set of eigenfunctions, with associated eigenvalues are:

$$\begin{aligned} u_0(x) &= 1, \quad u_n(x) = \sqrt{2}\cos(n\pi x) \quad n \in \mathbb{N}, \\ \lambda_n &= (n\pi)^2, \quad n \in \mathbb{N} \cup \{0\}. \end{aligned}$$

### 4.1.3. Mixed, Part 1.

$$-u''(x) = \lambda u(x)$$

$$u(0) = u'(1) = 0$$

Proceeding in similar fashion as in previous two examples, we find that the orthonormal set of eigenfunctions, with associated eigenvalues are

$$\begin{aligned} u_n(x) &= \sqrt{2}\sin\left[\left(n-\frac{1}{2}\right)\pi x\right] \quad n\in\mathbb{N}, \\ \lambda_n &= \left[\left(n-\frac{1}{2}\right)\pi\right]^2, \quad n\in\mathbb{N}. \end{aligned} \tag{.5}$$

### 4.1.4. Mixed, Part 2.

$$\begin{aligned} -u''(x) &= \lambda u(x) \\ u'(0) &= u(1) = 0 \end{aligned}$$

Proceeding in similar fashion as above, we find that the orthonormal set of eigenfunctions, with associated eigenvalues are

$$u_n(x) = \sqrt{2}\cos\left[\left(n-\frac{1}{2}\right)\pi x\right], \quad n\in\mathbb{N}, \tag{.6}$$

$$\lambda_n = \left[\left(n-\frac{1}{2}\right)\pi\right]^2, \quad n\in\mathbb{N}.$$

## 4.2. Two Spatial Dimensions

We now work in 2 dimensions, with the region $U$ a subset of $\mathbb{R}^2$, namely the unit square defined by all $(x, y)$ such that $0 < x < 1$ and $0 < y < 1$. We vary the boundary conditions in each example.

### 4.2.1. Dirichlet Conditions.

$$u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0.$$

With $u_i$ defined in (.3), we define

$$\begin{aligned} \Phi_{mn}(x,y) &= u_m(x)\, u_n(y) \qquad (.7) \\ &= 2\sin(m\pi x)\sin(n\pi y), \quad n, m \in \mathbb{N}. \end{aligned}$$

We see that

$$-\nabla^2 \Phi_{mn}(x,y) = -\left[u_m''(x)\, u_n(y) + u_m(x)\, u_n''(y)\right].$$

Now, from (.2), we have that

$$-u_m''(x) = \lambda_m u_m(x) \quad \text{and} \quad -u_n''(y) = \lambda_n u_n(y).$$

So,

$$\begin{aligned} -\nabla^2\Phi_{mn}(x,y) &= [-u_m''(x)]\,u_n(y) + u_m(x)\,[-u_n''(y)] \\ &= \lambda_m u_m(x)\,u_n(y) + u_m(x)\,\lambda_n u_n(y) \\ &= (\lambda_m + \lambda_n)\,u_m(x)\,u_n(y) \\ &= \lambda_{mn}\Phi_{mn}(x,y)\,. \end{aligned}$$

Thus the orthonormal set of eigenfunctions, with associated eigenvalues are:

$$\begin{aligned} \Phi_{mn}(x,y) &= 2\sin(m\pi x)\sin(n\pi y)\,, \quad m,n\in\mathbb{N}, \\ \lambda_{mn} &= \left(m^2+n^2\right)\pi^2, \quad m,n\in\mathbb{N}. \end{aligned}$$

### 4.2.2. Neumann Conditions.

$$u_y(x,0) = u_y(x,1) = u_x(0,y) = u_x(1,y) = 0.$$

With $u_i$ defined in (.4), we again define

$$\Phi_{mn}(x,y) = u_m(x)\,u_n(y)\,, \quad m,n\in\mathbb{N}\cup\{0\}\,.$$

We obtain orthonormal eigenfunctions

$$\begin{aligned}
\Phi_{00}(x,y) &= 1, \\
\Phi_{0n}(x,y) &= u_n(y) = \sqrt{2}\cos(n\pi y), \quad n \in \mathbb{N}, \\
\Phi_{m0}(x,y) &= u_m(x) = \sqrt{2}\cos(m\pi x), \quad m \in \mathbb{N}, \\
\Phi_{mn}(x,y) &= 2\cos(m\pi x)\cos(n\pi y), \quad m,n \in \mathbb{N},
\end{aligned}$$

with associated eigenvalues

$$\lambda_{mn} = \left(m^2 + n^2\right)\pi^2, \quad m,n \in \mathbb{N} \cup \{0\}.$$

### 4.2.3. Mixed Conditions, Part 1.

$$u_y(x,0) = u_y(x,1) = u(0,y) = u(1,y) = 0.$$

Neumann conditions with the variable $x$ fixed arbitrarily at $x_0$ necessitate using the eigenfunctions $u_n(y)$ with $n \in \mathbb{N} \cup \{0\}$ as defined in (.4). The Dirichlet conditions with $y$ arbitrarily fixed at $y_0$ require using $u_m(x)$ with $n \in \mathbb{N}$, as defined in (.3).

Then with $\Phi_{mn}$ defined generally as in (.7), we obtain orthonormal eigenfunctions

$$\begin{aligned}\Phi_{m0}(x,y) &= u_m(x) = \sqrt{2}\sin(m\pi x), \quad m \in \mathbb{N}, \\ \Phi_{mn}(x,y) &= 2\sin(m\pi x)\cos(n\pi y), \quad m,n \in \mathbb{N},\end{aligned}$$

with associated eigenvalues

$$\lambda_{mn} = \left(m^2+n^2\right)\pi^2, \quad m \in \mathbb{N}, \quad n \in \mathbb{N} \cup \{0\}.$$

### 4.2.4. Mixed Conditions, Part 2.

$$u_y(x,0) = u(x,1) = u(0,y) = u_x(1,y) = 0.$$

Similar to the previous case, the mixed conditions $u_y(x,0) = u(x,1) = 0$ require the use of the eigenfunctions $u_n(y)$ with $n \in \mathbb{N}$ as defined in (.6). The mixed conditions $u(0,y) = u_x(1,y) = 0$ require using the $u_m(x)$ with $m \in \mathbb{N} \cup \{0\}$, as defined in (.5). Then with $\Phi_{mn}$ defined generally as in (.7), we obtain orthonormal eigenfunctions

$$\Phi_{mn}(x,y) = 2\sin\left[\left(m-\frac{1}{2}\right)\pi x\right]\cos\left[\left(n-\frac{1}{2}\right)\pi y\right], \quad m,n \in \mathbb{N},$$

with associated eigenvalues

$$\lambda_{mn} = \left[\left(m - \frac{1}{2}\right)^2 + \left(n - \frac{1}{2}\right)^2\right]\pi^2, \quad m, n \in \mathbb{N}.$$

## 5. Analytical Solutions

### 5.1. Comparison Problem

We analytically solve IBVP (6.1). We define $\hat{v}(x,t) = v(x,t) - f(t)$, and require $f(0) = 0$. Then

$$\begin{aligned}
\hat{v}_t &= v_t - f'(t) = a_0 v_{xx} - f'(t), \\
\hat{v}(x,0) &= v(x,0) - f(0) = 0, \\
\hat{v}(0,t) &= v(0,t) - f(t) = 0, \\
\hat{v}_x(1,t) &= v_x(1,t) = 0.
\end{aligned}$$

Then the associated IBVP is

$$\begin{aligned}
\hat{v}_t &= a_0 \hat{v}_{xx} - f'(t), \qquad (.8) \\
\hat{v}(x,0) &= \hat{v}_x(1,t) = \hat{v}(0,t) = 0.
\end{aligned}$$

We generate eigenpairs from the S-L problem

$$\begin{aligned} \phi_n''(x) + \lambda_n \phi_n(x) &= 0, \\ \phi_n(0) &= \phi_n'(1) = 0, \end{aligned}$$

which for $n \in \mathbb{N}$ implies

$$\begin{aligned} \lambda_n &= \left(n - \frac{1}{2}\right)^2 \pi^2, \\ \phi_n(x) &= \sqrt{2} \sin\left[\left(n - \frac{1}{2}\right) \pi x\right]. \end{aligned}$$

Then using eigenfunction expansion, we have that

$$\begin{aligned} \hat{v}(x,t) &= \sum_{n=1}^{\infty} \hat{v}_n(t)\, \phi_n(x), \\ \hat{v}_t &= \sum_{n=1}^{\infty} \hat{v}_n'(t)\, \phi_n(x), \\ a_0 \hat{v}_{xx} &= a_0 \sum_{n=1}^{\infty} \hat{v}_n(t)\, \phi_n''(x) = a_0 \sum_{n=1}^{\infty} \hat{v}_n(t) \left[-\lambda_n \phi_n(x)\right], \\ f'(t) &= f'(t)(1) = f'(t) \sum_{n=1}^{\infty} (1, \phi_n)_0\, \phi_n(x) = f'(t) \sum_{n=1}^{\infty} c_n \phi_n(x). \end{aligned}$$

Here we note that

$$\sum_{n=1}^{\infty} c_n \phi_n(x) = \begin{cases} 0, & x = 0 \\ 1, & x \neq 0 \end{cases}.$$

We then substitute these expansions in (.8):

$$\sum_{n=1}^{\infty} \hat{v}_n'(t)\,\phi_n(x) = a_0 \sum_{n=1}^{\infty} \hat{v}_n(t)\left[-\lambda_n \phi_n(x)\right] - f'(t) \sum_{n=1}^{\infty} c_n \phi_n(x)$$

Then for $n \in \mathbb{N}$,

$$\hat{v}_n'(t) + a_0 \lambda_n \hat{v}_n(t) = -f'(t)\, c_n. \tag{.9}$$

We solve (.9) with variation of parameters:

$$\begin{aligned}
(\hat{v}_n)_h(t) &= c e^{-a_0 \lambda_n t} \\
(\hat{v}_n)_p(t) &= c(t)\, e^{-a_0 \lambda_n t} \Rightarrow \\
c'(t) &= -e^{a_0 \lambda_n t} f'(t)\, c_n \Rightarrow \\
c(t) &= -c_n \int_0^t e^{a_0 \lambda_n s} f'(s)\, ds \Rightarrow \\
(\hat{v}_n)_p(t) &= -c_n \int_0^t e^{a_0 \lambda_n s} f'(s)\, ds e^{-a_0 \lambda_n t} \\
&= -c_n \int_0^t e^{-a_0 \lambda_n (t-s)} f'(s)\, ds.
\end{aligned}$$

So,

$$\hat{v}_n(t) = ce^{-a_0\lambda_n t} - c_n \int_0^t e^{-a_0\lambda_n(t-s)} f'(s)\, ds.$$

Solving for $c$,

$$\begin{aligned} \hat{v}(x,0) &= \sum_{n=1}^{\infty} \hat{v}_n(0)\,\phi_n(x) = 0 \\ &\Rightarrow \hat{v}_n(0) = 0 \\ &\Rightarrow c = 0, \end{aligned}$$

we have that

$$\begin{aligned} \hat{v}_n(t) &= -c_n \int_0^t e^{-a_0\lambda_n(t-s)} f'(s)\, ds \qquad and \\ \hat{v}(x,t) &= -\sum_{n=1}^{\infty} c_n \int_0^t e^{-a_0\lambda_n(t-s)} f'(s)\, ds\phi_n(x) \qquad and \\ v(x,t) &= f(t) - \sum_{n=1}^{\infty} c_n \int_0^t e^{-a_0\lambda_n(t-s)} f'(s)\, ds\phi_n(x). \end{aligned} \tag{.10}$$

We derive an equivalent expression using integration by parts.

$$\begin{aligned} &\int_0^t e^{-a_0\lambda_n(t-s)} f'(s)\, ds \\ &= e^{-a_0\lambda_n(t-s)} f(s) \,|_0^t - a_0\lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \end{aligned}$$

$$= f(t) - a_0\lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds$$

Then (.10) is rewritten as

$$\begin{aligned} v(x,t) &= f(t) - \sum_{n=1}^{\infty} c_n \left[ f(t) - a_0\lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \right] \phi_n(x) \\ &= f(t) - f(t) \sum_{n=1}^{\infty} c_n \phi_n(x) + \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n(x) \\ &= f(t) \left( 1 - \sum_{n=1}^{\infty} c_n \phi_n(x) \right) + \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n(x) \end{aligned}$$

$$= \begin{cases} f(t) + \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n(0), & x = 0 \\ \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n(x), & x \neq 0 \end{cases}$$

$$= \begin{cases} f(t), & x = 0 \\ \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n(x), & x \neq 0 \end{cases}.$$

We see the initial and boundary conditions are met:

$$\begin{aligned} v(x,0) &= \begin{cases} f(0) = 0, & x = 0 \\ 0, & x \neq 0 \end{cases} \\ \frac{\partial v}{\partial x}(1,t) &= \sum_{n=1}^{\infty} c_n a_0 \lambda_n \int_0^t e^{-a_0\lambda_n(t-s)} f(s)\, ds \phi_n'(1) = 0 \\ v(0,t) &= f(t). \end{aligned}$$

## 5.2. Adjoint Problem

We analytically solve the final BVP (6.3). We define $\hat{w}(x,t) = w(x,t) - g(t)$, and require $g(T) = 0$. Then

$$\begin{aligned} \hat{w}_t &= w_t - g'(t) = -a_0 w_{xx} - g'(t), \\ \hat{w}(x,T) &= w(x,T) - g(T) = 0, \\ \hat{w}(0,t) &= w(0,t) - g(t) = 0, \end{aligned}$$

$$\hat{w}_x(1,t) = w_x(1,t) = 0.$$

Then the associated IBVP is

$$\begin{aligned} \hat{w}_t &= -a_0 w_{xx} - g'(t), \qquad (.11) \\ \hat{w}(x,T) &= \hat{w}_x(1,t) = \hat{w}(0,t) = 0. \end{aligned}$$

We generate eigenpairs from the S-L problem

$$\phi_n''(x) + \lambda_n \phi_n(x) = 0$$

$$\phi_n(0) = \phi_n'(1) = 0,$$

which for $n \in \mathbb{N}$ implies

$$\begin{aligned} \lambda_n &= \left(n - \frac{1}{2}\right)^2 \pi^2, \\ \phi_n(x) &= \sqrt{2} \sin\left[\left(n - \frac{1}{2}\right) \pi x\right]. \end{aligned}$$

Then using eigenfunction expansion, we have that

$$\begin{aligned} \hat{w}(x,t) &= \sum_{n=1}^{\infty} \hat{w}_n(t)\, \phi_n(x), \\ \hat{w}_t &= \sum_{n=1}^{\infty} \hat{w}_n'(t)\, \phi_n(x), \end{aligned}$$

$$\begin{aligned} a_0 \hat{w}_{xx} &= a_0 \sum_{n=1}^{\infty} \hat{w}_n(t)\, \phi_n''(x) = a_0 \sum_{n=1}^{\infty} \hat{w}_n(t) \left[-\lambda_n \phi_n(x)\right], \\ g'(t) &= g'(t)(1) = g'(t) \sum_{n=1}^{\infty} (1, \phi_n)_0\, \phi_n(x) = g'(t) \sum_{n=1}^{\infty} c_n \phi_n(x). \end{aligned}$$

Here we note that

$$\sum_{n=1}^{\infty} c_n \phi_n(x) = \begin{cases} 0, & x = 0 \\ 1, & x \neq 0 \end{cases}.$$

We then substitute these expansions in (.11):

$$\sum_{n=1}^{\infty} \hat{w}_n'(t)\, \phi_n(x) = a_0 \sum_{n=1}^{\infty} \hat{w}_n(t)\, \lambda_n \phi_n(x) - g'(t) \sum_{n=1}^{\infty} c_n \phi_n(x).$$

Then for $n \in \mathbb{N}$

$$\hat{w}_n'(t) + a_0 \lambda \hat{w}_n(t) = -g'(t) c_n. \tag{.12}$$

We solve (.12) with variation of parameters:

$$\begin{aligned}
(\hat{w}_n)_h(t) &= ce^{a_0 \lambda_n t} \\
(\hat{w}_n)_p(t) &= c(t) e^{a_0 \lambda_n t} \Rightarrow \\
c'(t) &= -e^{-a_0 \lambda_n t} g'(t) c_n \Rightarrow \\
c(t) &= -c_n \int_T^t e^{-a_0 \lambda_n s} g'(s)\, ds \Rightarrow
\end{aligned}$$

$$\begin{aligned}
(\hat{w}_n)_p(t) &= -c_n \int_T^t e^{-a_0 \lambda_n s} g'(s)\, ds e^{a_0 \lambda_n t} \\
&= c_n \int_t^T e^{-a_0 \lambda_n (s-t)} g'(s)\, ds.
\end{aligned}$$

So,

$$\hat{w}_n(t) = ce^{a_0 \lambda_n t} + c_n \int_t^T e^{-a_0 \lambda_n (s-t)} g'(s)\, ds.$$

Solving for $c$,

$$\begin{aligned}
\hat{w}(x,T) &= \sum_{n=1}^{\infty} \hat{w}_n(T) \phi_n(x) = 0 \\
&\Rightarrow \hat{w}_n(T) = 0 \Rightarrow c = 0,
\end{aligned}$$

we have that

$$\begin{aligned}
\hat{w}_n(t) &= c_n \int_t^T e^{-a_0\lambda_n(s-t)} g'(s)\, ds \qquad and \\
\hat{w}(x,t) &= \sum_{n=1}^{\infty} c_n \int_t^T e^{-a_0\lambda_n(s-t)} g'(s)\, ds \phi_n(x) \qquad and \\
w(x,t) &= g(t) + \sum_{n=1}^{\infty} c_n \int_t^T e^{-a_0\lambda_n(s-t)} g'(s)\, ds \phi_n(x). \qquad (.13)
\end{aligned}$$

We derive an equivalent expression using integration by parts.

$$\begin{aligned}
&\int_t^T e^{-a_0\lambda_n(s-t)} g'(s)\, ds \\
&= e^{-a_0\lambda_n(s-t)} g(s) \,|_t^T + a_0\lambda_n \int_t^T e^{-a_0\lambda_n(s-t)} g(s)\, ds \\
&= -g(t) + a_0\lambda_n \int_t^T e^{-a_0\lambda_n(s-t)} g(s)\, ds.
\end{aligned}$$

Then (.13) is rewritten as

$$\begin{aligned}
w(x,t) &= g(t)+\sum_{n=1}^{\infty}c_n\left[-g(t)+a_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\right]\phi_n(x).\\
&= g(t)-g(t)\sum_{n=1}^{\infty}c_n\phi_n(x)+\sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n(x)\\
&= g(t)\left(1-\sum_{n=1}^{\infty}c_n\phi_n(x)\right)+\sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n(x)\\
&= \begin{cases} g(t)+\sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n(0), & x=0\\ \sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n(x), & x\neq 0\end{cases}\\
&= \begin{cases} g(t), & x=0\\ \sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n(x), & x\neq 0\end{cases}.
\end{aligned}$$

We see the initial and boundary conditions are met:

$$\begin{aligned}
w(x,T) &= \begin{cases} g(T)=0, & x=0\\ 0, & x\neq 0\end{cases}\\
\frac{\partial w}{\partial x}(1,t) &= \sum_{n=1}^{\infty}c_na_0\lambda_n\int_t^T e^{-a_0\lambda_n(s-t)}g(s)\,ds\phi_n'(1)=0\\
w(0,t) &= g(t).
\end{aligned}$$

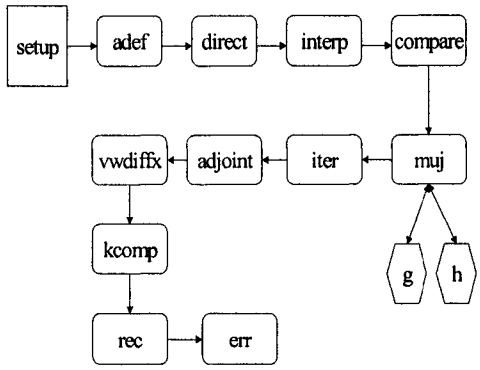


Figure .1. Matlab code schematic

## 6. Matlab Code

The basic schematic for the Matlab code used to generate the results of Chapter 8 is presented in Figure .1, where each of the subfigures represents a computational file. Selected details of the associated m-files are included below; they are not complete.

### 6.1. setup.m

This m-file is used to define the subspace dimension, boundary conditions, forcing terms, grid sizes, and calls most subsequent subfiles. The iterative process contains a while loop which measures the difference (adiff) between results of successive iterations (i3), as well as the number of iterations:

```
adef
direct
interp
compare
muj
while (adiff(i3)>.01) & (i3<102)
iter
adjoint
vwdiffx
kcomp
rec
end
err
```

### 6.2. adef.m

This m-file is used to define the discretized spatial coefficient to be recovered, and is used to produce the "measured" data. This file is used to set the appropriate time steps for the explicit finite differences scheme, as detailed in Chapter 6.

```
if bc==1
x1=linspace(0,1+dx1,2*Mx1+3);
elseif bc==2
x1=linspace(0,1,2*Mx1+1);
elseif bc==3
x1=linspace(-dx1,1+dx1,2*Mx1+5);
else
x1=linspace(-dx1,1,2*Mx1+3);
end
a1=(1)*((.1).*(x1<=0.5)+(.3).*(x1>.5));
A=max(ceil(a1));
dt1=(dx1^2)/(2*A);
Mt1=round(1/dt1);
dt=(dx^2)/(2*A);
```

```
Mt=round(1/dt);
```

### 6.3. direct.m

This m-file uses the explicit finite difference scheme detailed in Chapter 6 to solve the direct problem.

```
u=zeros(Mt1+1,Mx1+2);
for m=1:Mt1+1
u(m,1)=F0(m);
end
u(1,Mx1+2)=u(1,Mx1)+(2/a1(2*Mx1+1))*dx1*F1(1);
for m=1:Mt1
for n=2:Mx1+1
u(m+1,n)=dt1/(dx1^2)*(a1(2*n)*(u(m,n+1)-u(m,n))...
-a1(2*n-2)*(u(m,n)-u(m,n-1)))+u(m,n);
end
u(m+1,Mx1+2)=u(m+1,Mx1)+(2/a1(2*Mx1+1))*dx1*F1(m+1);
end
for m=1:Mt1+1
if bc==1
ux0(m)=(1/dx1)*(u(m,2)-u(m,1));
u1(m)=u(m,Mx1+1);
elseif bc==2
ux0(m)=(1/dx1)*(u(m,2)-u(m,1));
ux1(m)=(1/dx1)*(u(m,Mx1+1)-u(m,Mx1));
else
u0(m)=u(m,2);
u1(m)=u(m,Mx1+2);
end
end
```

### 6.4. interp.m

This m-file takes the data measurements from the file direct.m and either interpolates or extrapolates this data to the recovery grid.

```
if Mt1<Mt
Ux0=interp1(t1,ux0,t);
U1=interp1(t1,u1,t);
else
for m=1:Mt+1
Ux0(m)=interp1(t1,ux0,((Mt1+1)*(m-1))/((Mt1+1)*(Mt+1)));
U1(m)=interp1(t1,u1,((Mt1+1)*(m-1))/((Mt1+1)*(Mt+1)));
end
end
if Mx1<Mx
x=linspace(0,1+dx,2*Mx+3);
a=interp1(x1,a1,x);
else
for m=1:2*Mx+3
a(m)=interp1(x1,a1,((2*Mx1+3)*(m-1))/((2*Mx1+3)*(2*Mx+3)));
end
end
```

### 6.5. compare.m

This m-file uses the explicit finite difference scheme detailed in Chapter 6 to solve the comparison problem.

### 6.6. muj.m

This m-file is used to compute the scalars $\mu_j$ from the data produced in direct.m as well as the adjoint data defined in the files g.m and h.m.

```
g
for m=1:Mt+1
G0d(m)=G0a0(m)-G0a(m);
end
h
for m=1:Mt+1
G1d(m)=G1a0(m)-G1a(m);
end
for j=1:N
for m=1:Mt
H1(m,j)=(dt/2)*(G1(m,j)*G1d(m)+G1(m+1,j)*G1d(m+1));
end
end
for j=1:N
mu(j)=sum(H0(:,j))+sum(H1(:,j));
end
```

**6.6.1. g.m.** This m-file is used to construct the specific observation data and define the adjoint inputs at $x = 0$.

```
for m=1:Mt+1
Vx0(m)=(1/dx)*(V(m,2)-V(m,1));
end
for m=1:Mt+1
G0a(m)=a(1)*Ux0(m);
G0a0(m)=a0*Vx0(m);
end
t=linspace(0,1,Mt+1);
```

```
for j=1:N
gj=1.*(((1).*(t>=0 & t<=(j-1)/N)+(N/(j-N-1)
*(t-1)).*(t>(j-1)/N & t<=1)));
G0(:,j)=gj';
end
```

**6.6.2. h.m.** This m-file is used to construct the specific observation data and define the adjoint inputs at $x = 1$.

```
for m=1:Mt+1
Vx1(m)=(1/dx)*(V(m,Mx+1)-V(m,Mx));
end
for m=1:Mt+1
G1a(m)=U1(m);
G1a0(m)=V(m,Mx+1);
end
t=linspace(0,1,Mt+1);
for j=1:N
hj=1.*sin(j*pi*t);
G1(:,j)=hj';
end
```

### 6.7. iter.m

This m-file is used to iterate recovery process.

```
if i3==2
for n=1:2*Mx+3
ait(i3,n)=a0;
end
else
for n=1:2*Mx+3
if (tan(n)>=0)&(tan(n)<=A)
```

```
ait(i3,n)=tan(n);
elseif tan(n)<0
ait(i3,n)=.001;
else
ait(i3,n)=A;
end
end
end
ad1=ait(i3,:)-ait(i3-1,:);
for n=1:2*Mx
H9(n)=(1/2)*((1/2)*dx)*(ad1(n)^2+ad1(n+1)^2);
end
adiff(i3)=sqrt(sum(H9))/a0;
```

### 6.8. adjoint.m

This m-file uses the explicit finite difference scheme detailed in Chapter 6 to solve the adjoint problem.

```
for j=1:N
for m=1:Mt+1
W(m,1,j)=G0(m,j);
end
W(Mt+1,Mx+2,j)=W(Mt+1,Mx,j)+(2/ait(i3,2*Mx+1))*dx*G1(Mt+1,j);
for m=2:Mt+1
for n=2:Mx+1
W(Mt+2-m,n,j)=W(Mt+3-m,n,j)+dt/(dx^2)*(ait(i3,2*n)...
*(W(Mt+3-m,n+1,j)-W(Mt+3-m,n,j))...
-ait(i3,2*n-2)*(W(Mt+3-m,n,j)-W(Mt+3-m,n-1,j)));
end
W(Mt+2-m,Mx+2,j)=W(Mt+2-m,Mx,j)...
+(2/ait(i3,2*Mx+1))*dx*G1(Mt+2-m,j);
end
end
```

### 6.9. vwdiffx.m

This m-file computes the spatial derivatives $\partial_x v$ and $\partial_x w_j$ needed in the computation of the functions $K_j(x) = \int_0^T \partial_x v \partial_x w_j dt$.

```
for m=1:Mt+1
Vx(m,1)=(1/dx)*(V(m,2)-V(m,1));
for n=2:Mx+1
Vx(m,n)=(1/2)*(1/dx)*(V(m,n+1)-V(m,n-1));
end
end
for j=1:N
for m=1:Mt+1
Wx(m,1,j)=(1/dx)*(W(m,2,j)-W(m,1,j));
for n=2:Mx+1
Wx(m,n,j)=(1/2)*(1/dx)*(W(m,n+1,j)-W(m,n-1,j));
end
end
end
```

### 6.10. kcomp.m

This m-file computes the matrix $K$ for Equation (4.4).

```
for j=1:N
for m=1:Mt+1
for n=1:Mx+1
P(m,n,j)=Wx(m,n,j)*Vx(m,n);
end
end
end
for j=1:N
for n=1:Mx+1
```

```
for m=1:Mt
H2(m,n,j)=(dt/2)*(P(m,n,j)+P(m+1,n,j));
end
Kj(n,j)=sum(H2(:,n,j));
end
end
x=linspace(0,1,Mx+1);
for k=1:d
phi=sqrt(2)*sin((k-1/2)*pi*x);
Phi(:,k)=phi';
end
for j=1:N
for k=1:d
for n=1:Mx
H3(n)=(dx/2)*(Kj(n,j)*Phi(n,k)+Kj(n+1,j)*Phi(n+1,k));
end
c(k,j)=sum(H3);
end
end
for k=1:d
lambda(k)=(k-1/2)^2*pi^2;
end
for i=1:N
for j=1:N
for k=1:d
z2(k)=lambda(k)^(-s)*c(k,i)*c(k,j);
end
K(i,j)=sum(z2);
end
end
```

### 6.11. rec.m

This m-file solves (4.4) and completes the coefficient recovery.

```
for i=1:N
```

```
for j=1:Mx+1
for k=1:d
z1(k)=lambda(k)^(-s)*c(k,i)*Phi(j,k);
end
Delta(i,j)=sum(z1);
end
end
[VK,EK]=eig(K);
for i=1:Mx+1
for j=1:N
CK(j,i)=(transpose(VK(:,j))*Delta(:,i))/(EK(j,j)...
*transpose(VK(:,j))*VK(:,j));
DK(:,j,i)=CK(j,i)*VK(:,j);
end
end
for i=1:Mx+1
Lambda(:,i)=DK(:,1,i);
for j=2:N
Lambda(:,i)=Lambda(:,i)+DK(:,j,i);
end
end
tau=zeros(Mx+1,1);
for i=1:Mx+1
tau(i)=dot(Lambda(:,i),mu);
end
for i=1:Mx+1
adjust(i)=a0;
end
ta=tau+adjust';
```

### 6.12. err.m

This m-file is used to introduce error in the data measurements, and then perform the subsequent recovery for comparison against the error free recovery. This file calls certain subfiles, including interpe.m, muje.m, itere.m, adjointe.m and rece.m,

which performs the same functions as interp.m, muj.m, iter.m, adjoint.m, and rec.m under error.

```
er=10;
for m=1:Mt1+1
ux0e(m)=ux0(m)*(1+(-1)^(round(rand))*(er/100)*rand);
u1e(m)=u1(m)*(1+(-1)^(round(rand))*(er/100)*rand);
end
interpe
muje
while (adiffe(i4)>.01) & (i4<102)
itere
adjointe
vwdiffx
kcomp
rece
end
```