DISSERTATION

ADJOINT APPROACH TO PARAMETER IDENTIFICATION WITH APPLICATION TO THE RICHARDS EQUATION

Submitted by
Roger Thelwell
Department of Mathematics

In partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Colorado State University
Fort Collins, Colorado
Summer 2004

COLORADO STATE UNIVERSITY

May 12, 2004

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY ROGER THELWELL ENTITLED "ADJOINT APPROACH TO PARAMETER IDENTIFICATION WITH APPLICATION TO THE RICHARDS EQUATION" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Dr. Greg Butters

Dr. Simon Tavener

Dr. David Zachmann

Adviser: Dr. Paul DuChateau

Department Head: Dr. Simon Tavener

ABSTRACT OF DISSERTATION

ADJOINT APPROACH TO PARAMETER IDENTIFICATION WITH APPLICATION TO THE RICHARDS EQUATION

The inverse problem for the unknown coefficient ingredient of a class of quasilinear parabolic partial differential equations is considered. An approach based on utilizing adjoint versions of the direct problems to derive integral equations explicitly relating changes in inputs (coefficients) to changes in outputs (measured data) is presented. Using the integral equations it is possible to demonstrate properties of these maps. In the first problem, we show that the coefficient to data mappings are continuous, strictly monotone and injective. In the second, the mapping is shown to be explicitly invertible. The equations are further exploited to construct an approximate solution to the inverse problem. In the first problem, these equations are also used to analyze the error in the approximation. These equations are then used to construct a numerical recovery algorithm, which we call the integral identity method. Finally, numerical experiments are presented which explore the recovery process.

Roger Thelwell
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523
Summer 2004

ACKNOWLEDGEMENTS

I would like to thank my friend and advisor, Paul DuChateau. This work is testimony to his vision, patience and encouragement. I would also like to thank my committee members; Dr. Greg Butters, Dr. Simon Tavener and Dr. Dave Zachmann. Their advice and suggestions made this a much better paper. I would also like to thank my parents and family for giving me a love of learning. Lastly, and most importantly, my thanks to Liz. Although I think that she could have done without me at times, I know that I couldn't have done this without her.

TABLE OF CONTENTS

1 A Brief History **1**

1.1 Darcy, Darcy's Law and the Richards equation 1

1.2 Parameter Estimation 3

1.3 Why? 5

2 One parameter Identification **7**

2.1 Analysis of the Direct and Inverse Problems 8

2.2 The Approximate Solution of the Inverse Problem 20

3 One Parameter numerical experiments **39**

3.1 One parameter problem 39

3.2 Numerical methodology 40

3.3 Recovery algorithm 42

3.4 Numerical Code 45

3.4.1 Direct algorithm implementation 46

3.4.2 Pseudo Code for the Direct Problem 47

3.4.3 Experiment Algorithm implementation 48

3.4.4 Recovery pseudo code 50

3.5 Adaptive control 51

3.6 Experiments utilizing only the direct algorithm 53

3.6.1 Coefficient taken from a Sine family 54

3.6.2 Coefficients taken from an Arctan family 57

3.6.3 Coefficients taken from a Piecewise Linear family . . 59

3.7 Experiments Utilizing Full Recovery Algorithm 65

3.7.1 Data selection and weighting 65

3.7.2 Dual data selection 66

3.7.3 Dimension of uniform nodal basis 71

3.8 Local Nodal Refinement 75

3.9 Minimum Resolution 76

3.10 Width of Data strips 78

3.11 Iteration 79

3.12 Noisy data 84

4 Two parameter identification 86

4.1 Phase 1 problem 87

4.2 Phase 2 problem 105

5 Two parameter numerical experiments 113

5.1 Numerical methodology 115

5.2 Recovery Algorithm 115

5.3 Numerical Code 119

5.3.1 Direct algorithm implementation 120

5.3.2 Direct Problem 120

5.4 Phase 2 121

5.5 Comparison of Phase 1 and Phase 2 Experiments 121

5.6 Experiments utilizing only the forward solution 122

5.6.1 Allowing C(h) to vary 123

5.6.2 Allowing K(h) to vary 126

5.7 Experiments requiring full Recovery algorithm 128

5.7.1 Iteration 129

5.7.2 Dimension of Nodal Basis 132

5.7.3 Boundary Condition 134

5.7.4 Scaling of the Inversion 139

5.7.5 Scaling of the C coefficient 142

5.8 Recovery from Matlab generated data 143

5.9 Van Genuchten Family Recovery 147

5.10 Noisy data 148

6 Conclusions 153

A Existence Uniqueness for One parameter 158

B Data Generation 165

C Matlab Pseudo Code 168

D Maple code 171

E van Genuchten Parameters 176

LIST OF FIGURES

2.1 Isoclines 24

3.1 Coefficient and corresponding boundary data 54
3.2 Correlation between input D and output g 56
3.3 Data from Arctan with $\alpha = 11$ 57
3.4 Data from Arctan with $\alpha = 21$ 58
3.5 Data from Arctan with $\alpha = 41$ 59
3.6 Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 0.25$ 60
3.7 Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 0.5$ 60
3.8 Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 0.75$ 61
3.9 Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 1$ 61
3.10 Data from piecewise linear family with $c = 0.5, \beta = 1$ and $\alpha = 0.25$ 62
3.11 Data from piecewise linear family with $c = 0.5, \beta = 1$ and $\alpha = 0.5$ 63
3.12 Data from piecewise linear family with $c = 0.5, \beta = 1$ and $\alpha = 0.75$ 63
3.13 Data from piecewise linear family with $c = 0.5, \beta = 1$ and $\alpha = 1$ 64
3.14 Convergence comparison of $h(t)$ data 64
3.15 Error in λ parameter space for $H^*(t)$ data 67
3.16 Error in γ_2 parameter space 69
3.17 Error in λ parameter space for $G^*(t)$ data 70
3.18 Error in γ_1 parameter space 71
3.19 Recovery of a linear coefficient - g data only 73
3.20 Uniform Nodal Basis with g+h data 74
3.21 Local Grid refinement 76
3.22 Minimum Resolution 78
3.23 Diagonal elements 80
3.24 Recovery with iteration of function 3.8 80
3.25 Recovery of $D(u) = 1 + \frac{1}{2}\sin(2\pi u)$ 82
3.26 Recovery of $D(u) = 1.1 + \sin(10u)$ 83

3.27 Recovery from data with 10% noise 84
3.28 Flux data used by 3.27 84
3.29 Recovery from data with 15% noise 85

5.1 Data and coefficients with $\alpha = 1/3$ and $\beta = 2$ 124
5.2 Data and coefficients with $\alpha = 1$ and $\beta = 2$ 124
5.3 Data and coefficients with $\alpha = 3$ and $\beta = 2$ 125
5.4 Data and coefficients with $\alpha = 1$ and $\beta = 1$ 126
5.5 Data and coefficients with $\alpha = 1$ and $\beta = 2$ 127
5.6 Data and coefficients with $\alpha = 1$ and $\beta = 4$ 127
5.7 Iteration, $C(h) = 1 - (1/2)h$ and $K(h) = 2 + (1/2)h$ 131
5.8 Iteration, $C(h) = 2 + (1/2)h$ and $K(h) = 1 - (1/2)h$ 131
5.9 Error for various uniform nodal bases 132
5.10 Error summary for uniform nodal bases 133
5.11 Boundary Conditions 135
5.12 Simulations with boundary condition and C up K up 136
5.13 Simulations with boundary condition and C down K up 136
5.14 Simulations with boundary condition and C up K down 137
5.15 Simulations with boundary condition and C down K down . . . 137
5.16 Error over Linear family 138
5.17 Error over Power family 139
5.18 Scaling of M_{11} entry 140
5.19 Scaling of M_{12} entry 141
5.20 Scaling of M_{21} entry 141
5.21 C scale const 143
5.22 C scale linear 144
5.23 A Simple Recovery 145
5.24 A Slightly Harder example 145
5.25 Sample recovery 1 146
5.26 Sample recovery 2 146
5.27 Sample recovery 3 147
5.28 Recovery of coefficients associated with Si C L 149
5.29 Recovery of coefficients associated with C Loam 149
5.30 Recovery of coefficients associated with Sand 150
5.31 Recovery of coefficients associated with Silt 150
5.32 Recovery with 10% uniform noise 152
5.33 Data with 10% uniform noise 152

LIST OF TABLES

3.1 Breakthrough times for various coefficients 56

5.1 Data crossing times and values 125

5.2 Suction functions used in boundary experiments 134

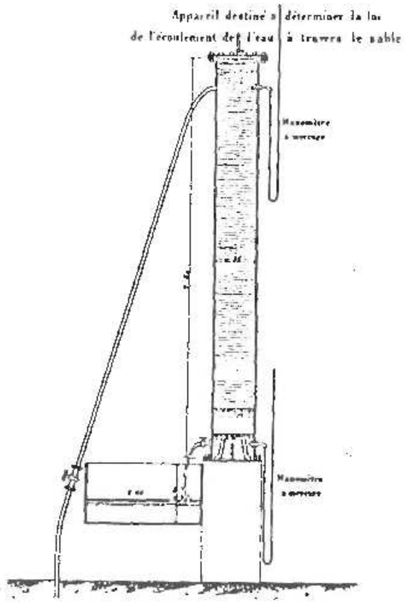
Chapter 1

A BRIEF HISTORY

1.1 Darcy, Darcy's Law and the Richards equation

In 1835, M. Henri Darcy was commissioned to enlarge and modernize the town water works of Dijon, France. Suitable filters were required for the expanding system. Unfortunately, the information necessary to determine the size of these filters was unavailable. Darcy designed an experiment that could provide the needed information.

A large column was filled with sand, with a water supply at the top, and a discharge at the bottom into a measuring tank. Valves controlled both the input and output rate, while manometers measured pressure. Experiments were conducted over a range of constant flow rates and the pressure was recorded at each rate. This data made apparent that the flow rate of water (q) through a sand filled vertical column is closely approximated by the relation


Appareil destiné à déterminer la loi
de l'écoulement de l'eau à travers le sable
Manomètre à mercure
Manomètre à mercure

$$q = -\kappa \frac{(h_1 - h_0)}{z_1 - z_0}$$

where q is the volume of water per unit area, h_1 and h_0 are the heights above a reference water level, $z_1 - z_0$ the depth of the sand, and κ a constant of proportionality. This constitutive relationship quickly became known as *Darcy's Law*, and has been extended to cover more general settings, including various soils and three dimensional non-vertical flow. In general differential form Darcy's law is

$$q = -\kappa \partial_z (h - z \ \sin(\vartheta)),$$

taking $h_0 = 0$. Here h denotes pressure head and $z \sin(\vartheta)$ gravity head, with ϑ the angle from vertical in the usual coordinate reference.

The conservation of mass statement for water content $\Theta(h)$ is given by

$$\partial_t \Theta + \partial_z q = 0.$$

Applying Darcy's law, yields

$$\partial_t(\Theta(h)) - \partial_z(\kappa(\partial_z h - \sin(\vartheta))) = 0.$$

Making the simplifying assumptions that the soil is homogeneous, unsaturated, and neglecting hysteresis, κ and Θ can be considered single valued functions of h. One can then write

$$C(h)\partial_t h = \partial_z(K(h)(\partial_z h - \sin(\vartheta))), \tag{1.1}$$

where $C(h) = d\Theta/dh$ represents soil capacity and $K(h)$ hydraulic conductivity. Equation (1.1) is known as the Capacity/Conductivity form of the one-dimensional Richards equation. In vertical flow situations, $\sin(\vartheta)$ is taken to be one, and thus the equation becomes

$$C(h)\partial_t h = \partial_z(K(h)(\partial_z h - 1)).$$

1.2 Parameter Estimation

Using partial differential equations to model physical systems is one of the oldest activities in applied mathematics. A complete model requires certain state inputs in the form of initial and/or boundary data together with what might be called structure inputs such as coefficients or source terms which are related to the physical properties of the system. Obtaining a unique solution for the associated well posed problem constitutes what we will call solving the direct problem. Solving the direct problem permits the computation of various system outputs of physical interest. On the other hand, when some of the required inputs are not available we may instead be able to determine the missing inputs from outputs that are measured rather than computed by formulating and solving an appropriate inverse problem. In particular, when the missing inputs are one or more unknown coefficients in the partial differential equation, the problem is called a coefficient identification problem.

The most common technique for identifying an unknown coefficient from some measured output is the method of output least squares [15, 8, 24, 20, 21]. Here the unknown coefficient, χ, is chosen from an appropriate space X and the output, $\Upsilon[\chi]$, is computed by solving the direct problem. One defines an error functional, $E[\chi] = ||\Upsilon[\chi] - y||_Y^2$, comparing the computed output to the measured value, y, in the norm of the output space, Y, and seeks to minimize E over X. Output least squares (OLS) methods are very general and can be efficiently programmed for computer implementation. Typically there are problems with lack of uniqueness, convergence to false minima, and instability under parameter mesh refinement, although a skillful user may be able to incorporate *a priori* information about the solution into the parametric description of the unknown coefficient in order to lessen some of these difficulties [15, 20]. General information about an input to output mapping is not readily available from OLS methods, since the connection between the inputs and outputs is expressed only indirectly through the solver.

An alternative to coefficient identification by output least squares is the so called equation error method [2, 16, 17, 9, 23]. Here the measured over specification is used as input to the differential equation in the direct problem which is viewed then as an equation for the unknown coefficient. This equation expresses a direct relationship between the unknown coefficient values and the measured data values. Since the relationship is frequently quite complicated, it is not easy to discern from equation error methods the properties of an input to output mapping. Additionally, these methods are quite problem dependent and produce varying degrees of success.

The Integral Identity method described in this paper is based on integral expressions relating changes in the unknown coefficients to corresponding changes in the measured output. These integral equations are derived by exploiting problems which are adjoint to the direct problem, an idea close to the techniques often used to estimate sensitivity in the OLS approach [24, 20]. The integral identities provide a means to study the input/output map without constructing an error functional, as is required with the OLS method. Integral identities are developed for two quasilinear parabolic equations. The first problem considers the recovery of a single parameter while the second extends the technique to recover two independent parameters.

1.3 Why?

The Richards Equation is perhaps the most widely applied model in porous media flow. Numerical solutions to Richards equation play a significant role in ground-water simulation and contaminant transport models. The validity of these depends on the accuracy of several soil parameters. Typically, Richards equation is posed using water content Θ and soil conductivity K as parameters. These parameters are calculated directly, via experiments relatively unchanged since the 1850's, or indirectly, through the formulation of a suitable mathematical inverse problem. The direct experiments are often tedious, and the goal of the inverse problem is to provide a structure in which the experiments become easier to conduct, while still providing accurate results. In this paper, the alternative Capacity Conductivity formulation of the Richards equation is used, since this is more amenable than the water content form to the inverse approach employed here. Water capacity $C(h)$, the derivative of water content $\Theta(h)$, describes how the soil holds water. Hydraulic conductivity $K(h)$ is a measure of soil water flow in response to a hydraulic pressure gradient.

Output Least Squares is a well known and effective method for coefficient inversion. While often the first tool used to approach many inverse problems, it provides almost no explanation in cases of failure. Various techniques have been made rigorous that strengthen the method. Little progress has been made in explaining instances where recovery fails; whether the OLS machinery is to blame, or rather some inherent ill-posedness of the physical system. This research seeks to more completely understand the underlying physical system. The Integral Identity method is shown here to allow a more complete picture of the recovery process.

Integral identities are developed for two quasilinear parabolic equations. The first problem considers the recovery of a single parameter while the second extends the technique to recover two independent parameters. The preliminary work focuses on identification of the diffusion coefficient in a quasilinear conduction diffusion equation of the form

$$\partial_t u(x,t) = \partial_x (D(u(x,t))\partial_x u(x,t)).$$

While other methods can effectively recover this unknown ingredient, we choose to explore integral identity methods. Many features of the Richards Equation are present in this simpler setting, and this preliminary study provided essential insight. Once the machinery was developed to treat the one parameter case, work began on the simultaneous identification of the water capacity, $C(h)$, and hydraulic conductivity, $K(h)$, functions of the Richards equation,

$$C(h(z,t))\partial_t h(z,t) = \partial_z (K(h(z,t))(\partial_z h(z,t) - 1)).$$

Here we have written the one dimensional form with vertical downward flow.

It is hoped that this research proves useful as both a tool to understand parameter estimation in the porous media setting, and perhaps contribute to a foundation for future work in adjoint approaches. Explicit numerical methods are presented here, which allow rapid determination of parameter information, and suggest a means to evaluate and adaptively control flow experiments.

Chapter 2

ONE PARAMETER IDENTIFICATION

In this chapter we analyze the one parameter identification problem. The method described in this chapter is based on an integral equation relating changes in the unknown coefficient to corresponding changes in the measured output. The integral equation is derived by exploiting a problem which is adjoint to the direct problem. The integral equation provides direct information about the input/output mapping. It is possible then to prove that the input to output map is continuous, monotone and injective. Moreover, it is shown that the input-output map is explicitly invertible when restricted to a finite dimensional space of polygonal coefficients. This observation provides the basis for a method for numerically approximating the unknown coefficient. It is shown that a unique polygonal approximation to the unknown coefficient is obtained by solving a triangular system of linear algebraic equations. Error estimates show that the accuracy of the approximation is limited by the precision of the data measurements so that there is an optimal attainable accuracy but exact determination of the coefficient is never possible.

In the next sections, we develop the theoretic framework which is fundamental to the recovery algorithm. We begin with an analysis of the direct problem.

2.1 Analysis of the Direct and Inverse Problems

Consider the following IBVP for a quasilinear conduction diffusion equation on the domain $Q_T = \{0 < x < 1,\, 0 < t < T\}$,

$$\begin{aligned} \partial_t u(x,t) = \partial_x(D(u)\partial_x u(x,t)) = \partial_{xx}B(u(x,t)) & \qquad \text{on } Q_T \\ u(x,0) = f(0) & \qquad 0 < x < 1 \\ u(0,t) = f(t); \quad \partial_x u(1,t) = 0, & \qquad 0 < t < T. \end{aligned} \tag{2.1}$$

Here $\quad B(u) = \displaystyle\int_{f(0)}^{u} D(s)\, ds \quad$ and we suppose

$$f \in \mathcal{C}^1[0,T] \text{ and } f'(t) > 0 \text{ for } t > 0 \tag{2.2}$$

For f satisfying (2.2), we let $J = [f(0), f(T)]$, and then suppose for positive constants, $D^\flat \leq D^\sharp$ and K,

$$\left.\begin{aligned} D^\flat \leq D(u) \ \leq D^\sharp, & \qquad \text{for } u \in J \qquad (a) \\ |D(\mu_2) - D(\mu_1)| \ \leq K|\mu_2 - \mu_1| & \qquad \forall \mu_1, \mu_2 \in J. \qquad (b) \end{aligned}\right\} \tag{2.3}$$

Note that any polygonal function (i.e., a continuous and piecewise linear function) satisfies (2.3a) and that the difference of two functions satisfying both conditions of (2.3) is bounded away from zero and has at most finitely many zeroes on J.

Using standard techniques, it has been shown that conditions 2.2 and 2.3 allow the initial value problem (2.1) to have a unique weak solution, denoted by u, with the properties

$$\begin{aligned} & u(x,t) \in L^2[0,T : H^1(0,1)] \cap \mathcal{C}[0,T : L^2(0,1)] \text{ and} \\ & \partial_t u(x,t) \in L^2[0,T : H^{-1}(0,1)]. \end{aligned}$$

The details of this argument may be found in appendix A.

We now consider the inverse problem in which the coefficient $D = D(u)$ is to be identified from measured output data. There are a variety of output measurements that are experimentally feasible in any given physical setting; we are going to base our identification on one or the other of the following observations at the boundary,

$$\begin{aligned} &g(t) = -D(u(0,t))\partial_x u(0,t) \quad \text{or} \\ &h(t) = u(1,t), \quad 0 < t < T \end{aligned}$$

If we denote the class of uniformly positive, Lipschitz coefficients D satisfying (2.3) by $W(J)$, then for a fixed f satisfying (2.2), we can define mappings

$$\begin{aligned} &\Phi \text{ and } \Psi : W(J) \longrightarrow L^2[0,T] \\ &\Phi[f,D] = g \\ &\Psi[f,D] = h, \end{aligned}$$

which assign to a coefficient D from $W(J)$, the flux data g or the function value data h, obtained by solving the direct problem (2.1) with inputs f and D. Then solving the inverse problem will amount to inverting these mappings.

We begin with a result about the IBVP (2.1).

Lemma 2.1.1. *Suppose f and D satisfy (2.2) and (2.3) and let $u = u(x,t)$ denote the corresponding solution of (2.1). Then*

a) for each $t \in (0,T)$, $\quad f(0) \leq u(x,t) \leq f(t), \quad 0 \leq x \leq 1$

b) $\partial_x u(x,t) < 0 \quad$ almost everywhere on Q_T

Proof. It follows from (2.1) that

$$\begin{aligned}\partial_t[f(t)-u(x,t)]-\partial_{xx}[B(f(t))-B(u(x,t))]=&f'(t) && \text{on } Q_T\\ f(0)-u(x,0)=&0 && 0<x<1\\ f(t)-u(0,t)=0,\quad \partial_x[f(t)-u(1,t)]=&0, && 0<t<T.\end{aligned}$$

Then we multiply the equation by an arbitrary test function, $\psi(x,t)$, and integrate by parts,

$$\begin{aligned}&-\int\int_{Q_T}[(f-u)\partial_t\psi+(B(f)-B(u))\partial_x x\psi]\ dx\ dt\\ &\quad+\int_0^1(f-u)\psi\Big|_{t=0}^{t=T}\ dx\\ &\quad-\int_0^T[\psi\partial_x(B(f)-B(u))-(B(f)-B(u))\partial_x\psi]\Big|_{x=0}^{x=1}\ dt\\ &\qquad=\int\int_{Q_T}f'(t)\psi\ dx\ dt.\end{aligned}$$

Note that

$$B(f(t))-B(u(x,t))=k(x,t)(f-u),$$

where we define $k(x,t)=D(\mu(x,t))$ for $\mu(x,t)$ between $f(t)$ and $u(x,t)$.

Next we require $\psi(x,t)$ to solve the adjoint problem,

$$\begin{aligned}\partial_t\psi(x,t)+k(x,t)\partial_{xx}\psi(x,t)=&F(x,t) && \text{in } Q_T,\\ \psi(x,T)=&0, && 0<x<1,\\ \psi(0,t)=&0,\quad \partial_x\psi(1,t)=0, && 0<t<T,\end{aligned}$$

for a smooth function $F(x,t)$. Then the integral expression above reduces to

$$-\int\int_{Q_T}(f-u)F(x,t)\ dxdt=\int\int_{Q_T}f'(t)\psi(x,t)\ dxdt \tag{2.4}$$

The smoothness of $k(x,t)$ and $F(x,t)$ imply that the strong maximum principle can be applied to the adjoint problem to conclude that if the function $F(x,t)$ is positive in Q_T, then $\psi(x,t) < 0 \in Q_T$. Since f satisfies (2.2), it follows that for every function $F(x,t)$ which is positive in Q_T, the right side of (2.4) is negative. That is, for every $F(x,t)$, smooth and positive in Q_T,

$$\int\int_{Q_T} (f-u)\ F(x,t)\ dx\ dt > 0.$$

But this is just the assertion that $f(t) - u(x,t)$ is positive in the sense of distributions on Q_T. Given the smoothness of the solution $u(x,t)$ this means $f(t) > u(x,t)$ almost everywhere on Q_T. Applying the same reasoning to $u(x,t) - f(0)$, we arrive at the expression

$$-\int\int_{Q_T} (u(x,t) - f(0))\ F(x,t)\ dx\ dt = \int_0^T B(f(t))\partial_x\psi(0,t)\ dt.$$

where we again use that $\psi(x,t) < 0$ in Q_T if the function $F(x,t)$ is positive in Q_T. Now this fact, together with the adjoint boundary conditions imply that $\partial_x\psi(0,t) < 0$, for $0 < t < T$. Then the conclusion follows as before. This completes the proof of (a).

To prove (b), multiply both sides of (2.1) by $\partial_x\phi(x,t)$ for an arbitrary test function $\phi(x,t)$ and use integration by parts to arrive at

$$\begin{aligned} 0 = \int\int_{Q_T} \partial_x u[\partial_t\phi + D(u)\partial_{xx}\phi]\ dx\ dt + \int_0^T \phi\partial_t u\Big|_{x=0}^{x=1}\ dt \\ - \int_0^1 \phi\partial_x u\Big|_{t=0}^{t=T}\ dx - \int_0^T \partial_x\phi\partial_x B(u)\Big|_{x=0}^{x=1}\ dt. \end{aligned}$$

Now require that $\phi(x,t)$ satisfies the adjoint problem

$$\begin{aligned} \partial_t\phi(x,t) + D(u(x,t))\partial_{xx}\phi(x,t) = F(x,t) \qquad & \text{in } Q_T, \\ \phi(x,T) = 0, \qquad & 0 < x < 1, \\ \partial_x\phi(0,t) = 0, \quad \phi(1,t) = 0, \qquad & 0 < t < T. \end{aligned}$$

Then the preceding integral expression reduces to

$$\int\int_{Q_T} \partial_x u(x,t) F(x,t)dxdt = \int_0^T \phi(0,t)f'(t)dt$$

The maximum principle can be applied to the adjoint problem to conclude that $\phi(x,t) < 0$ in Q_T if the continuous function $F(x,t)$ is positive in Q_T. In particular, $\phi(0,t) < 0$ for $0 < t < T$. Since f satisfies (2.2), it follows that for every function $F(x,t)$ which is positive in Q_T the right side of the expression is negative. Then it follows as in the proof of part a) that $\partial_x u(x,t) < 0$ almost everywhere in Q_T. □

The results of this lemma are crucial to the proof of,

Lemma 2.1.2. *Suppose f satisfies (2.2) and D_1, D_2 both satisfy (2.3). Then if $D_1(u) > D_2(u)$ for $u \in J = [f(0), f(T)]$ it follows that,*

a) $\Phi[f, D_1](t) > \Phi[f, D_2](t), \quad 0 < t < T,$

b) $\Psi[f, D_1](t) < \Psi[f, D_2](t), \quad 0 < t < T.$

Proof. For w in J, let $B_j'(w) = D_j(w)$ for $j = 1, 2$. Also, let u_1, u_2 denote the solutions for the direct problem with coefficients D_1, D_2, respectively. Then

$$\partial_t(u_1 - u_2) - \partial_{xx}(B_1(u_1) - B_2(u_2)) = 0 \quad \text{or}$$
$$\partial_t(u_1 - u_2) - \partial_{xx}(B_1(u_1) - B_1(u_2)) = \partial_{xx}(B_1(u_2) - B_2(u_2))$$

and, for an arbitrary test functions $\phi = \phi(x,t)$ and arbitrary $\tau,\ 0 < \tau \leq T$,

$$\int_0^\tau \int_0^1 [\partial_t(u_1 - u_2) - \partial_{xx}(B_1(u_1) - B_1(u_2))]\, \phi\, dx\, dt$$
$$= \int_0^\tau \int_0^1 \phi\, \partial_{xx}(B_1(u_2) - B_2(u_2))\, dx\, dt.$$

Apply integration by parts on the left side of this equation,

$$\begin{aligned}
&\int_0^\tau \int_0^1 \left[\partial_t(u_1 - u_2) - \partial_{xx}(B_1(u_1) - B_1(u_2))\right] \phi \, dx \; dt \\
&\quad = -\int_0^\tau \int_0^1 (u_1 - u_2) \left\{\partial_t \phi + D_1(\mu(x,t))\partial_{xx}\phi\right\} \, dx \, dt \\
&\qquad + \int_0^1 (u_1 - u_2)\phi\Big|_{t=0}^{t=\tau} dx \\
&\qquad - \int_0^\tau \left[\phi \, \partial_x(B_1(u_1) - B_1(u_2)) - \partial_x \phi \, (B_1(u_1) - B_1(u_2))\right]\Big|_{x=0}^{x=1} dt,
\end{aligned}$$

and on the right side as well,

$$\begin{aligned}
&\int_0^\tau \int_0^1 \phi \left\{\partial_{xx}(B_1(u_2) - B_2(u_2))\right\} \; dx \; dt \\
&\quad = \int_0^\tau \left[\phi \; \partial_x(B_1(u_2) - B_2(u_2))\right]\Big|_{x=0}^{x=1} dt \\
&\qquad - \int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2)) \; \partial_x \phi \; \partial_x u_2 \; dx \; dt
\end{aligned}$$

where for all $(x,t) \in Q_\tau$, $\mu(x,t)$ lies between $u_1(x,t)$ and $u_2(x,t)$ such that for $(x,t) \in Q_\tau$

$$B_1(u_1(x,t)) - B_1(u_2(x,t)) = D_1(\mu(x,t))[u_1(x,t) - u_2(x,t)].$$

We then obtain the integral expression,

$$\begin{aligned}
&-\int_0^\tau \int_0^1 (u_1 - u_2) \left\{\partial_t \phi + D_1(\mu(x,t))\partial_{xx}\phi\right\} \, dx \, dt \\
&\quad + \int_0^1 (u_1 - u_2)\phi \,\Big|_{t=0}^{t=\tau} dx \\
&\quad - \int_0^\tau \left[\phi \, \partial_x(B_1(u_1) - B_2(u_2)) - \partial_x \phi \, (B_1(u_1) - B_1(u_2))\right]\Big|_{x=0}^{x=1} dt \\
&\qquad\qquad = -\int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2))\partial_x \phi \, \partial_x u_2 \, dx \, dt.
\end{aligned}$$

The boundary and initial conditions of the direct problem cause this expression to reduce to,

$$-\int_0^\tau \int_0^1 (u_1 - u_2)\left\{\partial_t\phi + D_1(\mu(x,t))\partial_{xx}\phi\right\} dx\, dt$$
$$+ \int_0^1 (u_1 - u_2)(x,\tau)\phi(x,\tau)\, dx$$
$$+ \int_0^\tau \phi(0,t)\partial_x(B_1(u_1) - B_2(u_2))dt$$
$$- \int_0^\tau \partial_x\ \phi(1,t)\ [B_1(u_1) - B_1(u_2)]\ dt$$
$$= -\int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2))\partial_x\phi\partial_x\ u_2\ dx\ dt \tag{2.5}$$

Now require the arbitrary function $\phi(x,t)$ to solve the so-called g-adjoint problem,

$$\begin{aligned} \partial_t\phi + D_1(\mu(x,t))\partial_{xx}\phi &= 0 && \text{in } Q_\tau && (2.6)\\ \phi(x,\tau) &= 0 && 0 < x < 1, \\ \phi(0,t) = \theta(t), \quad \partial_x\phi(1,t) &= 0, && 0 < t < \tau, \end{aligned}$$

where $\theta(t) = F(\tau - t)$ and F is any function satisfying (2.2). Then (2.5) reduces to

$$\int_0^\tau \theta(t)[g_1(t) - g_2(t)]\, dt = \int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2))\partial_x\phi\, \partial_x u_2\, dx\, dt. \tag{2.7}$$

An argument similar to the one used in the proof of lemma (2.1.2), applied to (2.6), shows that the assumption on the adjoint input, θ, implies $\partial_x\phi(x,t) < 0$ on Q_τ. Since $\partial_x u_2 < 0$ on Q_T and $D_1(u_2) > D_2(u_2)$ it follows that the right side of the last expression is positive. Since (2.7) holds for all $\theta(t) = F(\tau - t)$, such that F satisfies (2.2), it follows that

$$g_1(t) - g_2(t) > 0 \quad \text{for} \quad 0 < t < T$$
$$i.e. \qquad g_1(t) = \Phi[f, D_1](t) > \Phi[f, D_2](t) = g_2(t).$$

To see that this is true, note first that if $D_1(u) > D_2(u)$ for $u \in J$, then existence of an interval $(0, t_1)$ with $g_1(t) < g_2(t)$ for $0 < t < t_1$ is precluded by (2.7) simply by choosing $\tau = t_1$. Suppose then that there exists $t_2 > t_1 > 0$ such that $g_1(t) \geq g_2(t)$ for $0 < t \leq t_1$ and $g_1(t) < g_2(t)$ for $t_1 < t < t_2$. Then choosing $\tau = t_2$ in (2.7) implies that for any admissible $\theta(t)$,

$$\begin{aligned}\int_{t_1}^{t_2} \theta(t)[g_1(t) - g_2(t)]\,dt &= \int_{t_1}^{t_2}\int_0^1 (D_1(u_2) - D_2(u_2))\partial_x\phi\,\partial_x u_2\,dx\,dt \\ &\quad + \int_0^{t_1}\int_0^1 (D_1(u_2) - D_2(u_2))\partial_x\phi\,\partial_x u_2\,dx\,dt \\ &\quad - \int_0^{t_1} \theta(t)[g_1(t) - g_2(t)]\,dt.\end{aligned}$$

By applying equality (2.7) with $\tau = t_1$, the last two terms of the previous equation vanish. By assumption, the right side of the resulting expression is strictly positive, while a suitable choice of $\theta(t)$ makes the left side negative. This contradicts (2.7).

Suppose now that we choose ϕ in (2.5) to solve a problem different from (2.6). This problem will be called the h-adjoint problem,

$$\begin{aligned}\partial_t\phi + D_1(\mu(x,t))\partial_{xx}\phi = 0 \qquad & \text{in } Q_\tau, \qquad (2.8)\\ \phi(x,\tau) = 0, \qquad & 0 < x < 1,\\ \phi(0,t) = 0, \quad D(\mu(1,t))\partial_x\phi(1,t) = \beta(t), \qquad & 0 < t < \tau.\end{aligned}$$

Here, choose $\beta(t) = F(\tau - t)$ where F is any function satisfying (2.2). Then (2.5) reduces to

$$\begin{aligned}&\int_0^\tau D(\mu(1,t))\partial_x\phi(1,t)(u_1(1,t) - u_2(1,t))\;dt \\ &\qquad = \int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2))\;\partial_x\phi\;\partial_x u_2\;dx\;dt,\end{aligned}$$

or

$$\int_0^\tau \beta(t)\left[h_1(t) - h_2(t)\right] dt = \int_0^\tau \int_0^1 (D_1(u_2) - D_2(u_2))\partial_x\phi\partial_x u_2 dxdt. \quad (2.9)$$

In this case, the hypotheses on $\beta(t)$ imply that $\partial_x\phi(x,t) > 0$ on Q_τ and since $\partial_x u_2 < 0$ and $D_1(u_2) > D_2(u_2)$ for all u_2 in J, it then follows that the right side of (2.9) is negative. Since this holds with $\beta(t) = F(\tau - t)$ for any F satisfying (2.2), it follows that

$$\Psi\left[f, D_1\right](t) = u_1(1,t) < u_2(1,t) = \Psi\left[f, D_2\right](t) \qquad for \qquad 0 < t < \tau.$$

Finishing the argument as in the previous case, we see that this holds for $\tau \leq T$. □

The conclusions of lemma 2.1.2 assert that input to output mappings Φ and Ψ are monotone mappings. More precisely, the mapping Φ is *isotone* while the mapping Ψ is an *antitone* mapping.

Now suppose $D_1(u_1)$ and $D_2(u_1)$ are any two coefficients, both satisfying (2.3). Let $u_1(x,t), u_2(x,t)$ denote the solutions of (2.1) when the coefficient is, respectively, $D_1(u)$ and $D_2(u)$, and for $i = 1, 2$, let

$$g_i(t) = \Phi[f, D_i] \qquad \text{and}$$
$$h_i(t) = \Psi[f, D_i], \qquad 0 < t < T.$$

Now choose the data in the adjoint problems (2.6) and (2.8) as,

$$\phi(0,t) = \theta(t) = \frac{g_1(t) - g_2(t)}{||g_1 - g_2||_{L^2[0,T]}}, \quad \text{in} \quad (2.6)$$

and

$$D_1(\mu(1,t))\partial_x\psi(1,t) = \beta(t) = \frac{h_1(t) - h_2(t)}{||h_1 - h_2||_{L^2[0,T]}} \quad \text{in} \quad (2.8).$$

It follows at once from (2.7) that

$$\begin{aligned} ||\, g_1 - g_2||_{L^2[0,T]} &\leq \left| \int_0^T \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \phi \, dx\, dt \right| \\ &\leq C\ ||D_1 - D_2||_\infty \end{aligned}$$

and from (2.9) that

$$\begin{aligned} ||\, h_1 - h_2||_{L^2[0,T]} &\leq \left| \int_0^T \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \psi \, dx\, dt \right| \\ &\leq C\ ||D_1 - D_2||_\infty \end{aligned}$$

Evidently, this is just the assertion that Φ and Ψ are continuous as a function of D from $W(J)$ into $L^2[0,T]$; i.e.,

$$||g_1 - g_2||_{L^2[0,T]} = ||\Phi(f, D_1) - \Phi(f, D_2)||_{L^2[0,T]} \leq C||D_1 - D_2||_\infty$$

$$||h_1 - h_2||_{L^2[0,T]} = ||\Psi(f, D_1) - \Psi(f, D_2)||_{L^2[0,T]} \leq C||D_1 - D_2||_\infty.$$

Having shown that Φ and Ψ are continuous and strictly monotone, one is encouraged to believe that this inverse problem is not so badly ill posed and that Φ and Ψ might be continuously invertible. Such a strong result seems to be unlikely without a simple ordering on the domain and range of these maps but it is at least true that the input/output maps Φ and Ψ are injective as the following lemma shows.

Lemma 2.1.3. *For a fixed f satisfying (2.2) and coefficients $D_1, D_2 \in W(J)$ let $g_k(t) = \Phi[f, D_k]$ and $h_k(t) = \Psi[f, D_k]$, for $k = 1, 2$.*

Then

a) $\Phi[f, D_1] = \Phi[f, D_2], 0 < t < T,$ implies $D_1(u) = D_2(u)$ for $u \in J$.

b) $\Psi[f, D_1] = \Psi[f, D_2], 0 < t < T,$ implies $D_1(u) = D_2(u)$ for $u \in J$.

Proof. Suppose first that $D_1(f(0)) = D_2(f(0))$. Now, since D_1 and D_2 both satisfy (2.3), their difference satisfies (2.3) and if these functions are not identical on J then there exists a positive time t_1, $0 < t_1 \leq T$, where the difference, $D_1(f(t)) - D_2(f(t))$ is of one sign on $[0, t_1]$. Then lemma 2.1.1(a) implies $D_1(u_2(x,t)) - D_2(u_2(x,t))$ is of one sign on $(0,1) \times (0, t_1)$. Using the identity (2.7), we have

$$\int_0^{t_1} \int_0^1 (D_1(u_2(x,t)) - D_2(u_2(x,t)))\, \partial_x u_2 \, \partial_x \phi \, dx \, dt$$
$$= \int_0^{t_1} (g_1(t) - g_2(t))\, \theta(t)\, dt,$$

where ϕ solves (2.6) with $\tau = t_1$. Then the hypotheses imply the right side of this equation vanishes; i.e.,

$$\int_0^{t_1} \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2 \, \partial_x \phi \, dx \, dt = 0$$

and this holds independently of the data $\theta(t)$ chosen as input to the adjoint problem. It is clearly possible to choose $\theta(t)$ so that $\partial_x \phi < 0$ on $(0,1) \times (0, t_1)$ and in view of lemma 2.1.1(b) it is also the case that, $\partial_x u_2 < 0$ on $(0,1) \times (0, t_1)$. Then the vanishing integral above has an integrand which is of one sign over the domain of integration and vanishes on no positive measure subset of the domain. This contradiction is in opposition to the assumption that D_1 and D_2 are not identical.

If we suppose $D_1(f(0)) \neq D_2(f(0))$ then it follows that either there is a smallest time $t_1, 0 < t_1 < T$, where the difference $D_1(f(t)) - D_2(f(t))$ is zero, or else $t_1 = T$ and the difference is of one sign on $[0, T]$. In either case, it is evident that $D_1(f(t)) - D_2(f(t))$ is of one sign on $[0, t_1]$, $0 < t_1 \leq T$, and the argument can be completed as before. A similar argument, using the identity in (2.9), establishes conclusion (b). □

Formally, we can write,

$$\begin{aligned}
(\Phi\left[f, D_1\right] - \Phi\left[f, D_2\right], \theta)_{L^2} &\overset{def}{=} (\delta\Phi\left[D_1, D_2\right]\Delta D, \theta)_{L^2} \\
&= \left\langle \Delta D,^t \delta\Phi\left[D_1, D_2\right]\, \theta \right\rangle_{W(J)\times W(J)^*}.
\end{aligned}$$

In view of (2.7),

$$\begin{aligned}
(\Phi\left[f, D_1\right] - \Phi\left[f, D_2\right], \theta)_{L^2} &= \int_0^T (g_1(t) - g_2(t))\theta(t)\, dt \\
&= \int_0^T \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \phi\, dx\, dt, \\
&= \left\langle \Delta D,^t \delta\Phi\left[D_1, D_2\right]\, \theta \right\rangle_{W(J)\times W(J)^*}
\end{aligned}$$

Similarly,

$$\begin{aligned}
(\Psi\left[f, D_1\right] - \Psi\left[f, D_2\right], \beta)_{L^2} &\overset{def}{=} (\delta\Psi\left[D_1, D_2\right]\Delta D, \beta)_{L^2} \\
&= \left\langle \Delta D,^t \delta\Psi\left[D_1, D_2\right]\, \beta \right\rangle_{W(J)\times W(J)^*},
\end{aligned}$$

and, referring to (2.9),

$$\begin{aligned}
(\Psi\left[f, D_1\right] - \Psi\left[f, D_2\right], \beta)_{L^2} &= \int_0^T (h_1(t) - h_2(t))\beta(t)\, dt \\
&= \int_0^T \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \psi\, dx\, dt \\
&= \left\langle \Delta D,^t \delta\Psi\left[D_1, D_2\right]\, \beta \right\rangle_{W(J)\times W(J)^*}
\end{aligned}$$

Evidently, (2.7),(2.9) provide realizations for ${}^t\delta\Phi[D_1, D_2]$ and ${}^t\delta\Psi[D_1, D_2]$, the Gateaux derivatives with respect to D of the mappings Φ and Ψ. It will be shown in the next section that ${}^t\delta\Phi[D_1, D_2]$ and ${}^t\delta\Psi[D_1, D_2]$ are invertible in an approximate sense. More precisely we will devise a restriction of the coefficient to data maps that induces a mapping from $\mathbb{R}$ into $\mathbb{R}$. The restriction inherits the strict monotonicity and continuity from the coefficient to data map hence the restriction defines a homeomorphism from its domain onto its range. Inversion of this mapping leads to an approximate inverse for the coefficient to data map.

2.2 The Approximate Solution of the Inverse Problem

We consider the inverse problem in which the coefficient $D = D(u)$ is to be identified from data which is assumed to be recorded at fixed nodes $0 = t_0 < t_1 < \cdots < t_N = T$ in the interval $[0, T]$:

$$\text{data}(f, g) \left\{ \begin{array}{lcl} f(t_k) & = & \mu_k \\ g(t_k) & = & -D(\mu_k)\,\partial_x u_1(0, t_k) = \gamma_k, \qquad k = 0, 1, ..., N \end{array} \right.$$

We are also interested in the identification of $D_1 = D(u_1)$ based on the alternative data,

$$\text{data}(f, h) \left\{ \begin{array}{lcl} f(t_k) & = & \mu_k \\ h(t_k) & = & u_1(1, t_k) = \eta_k, \qquad\qquad\qquad k = 0, 1, ..., N \end{array} \right.$$

More precisely, we are going use one or the other of these data sets to construct a polygonal (i.e. piecewise linear and continuous) approximation to the unknown coefficient $D(u)$. The data set, $f_k = f(t_k),\ k = 0, 1, \ldots, N$, is assumed to be given at fixed nodes which define a partition, $\{0 = t_0 < t_1 < \ldots < t_N = T\}$, of the interval $I = [0, T]$. This partition of I will be called the *inner mesh.* We then define an associated (but coarser) partition of $J = [f(0), f(T)]$, the domain of the coefficient D. This partition will be called the *outer mesh* and is given by $f(0) = \mu_0 < \mu_1 < \cdots < \mu_M = f(T)$, ;i.e., $\mu_0 = f_0$, and $\mu_M = f_N$ and for each $j = 1, ..., M < N$ we have $\mu_j = f_k$ for some $k \geq j$.

It is necessary for the outer mesh to be coarser than the inner mesh since on each subinterval in the outer mesh, we will need to compute interior values of the solution $u(x, t)$ for the direct problem in order to be able to evaluate the integrals which appear in the identities used in the identification. Between two outer mesh knots $\mu_j = f(t_k)$ and μ_{j+1}, there must occur several inner mesh knots and this fact prevents the outer mesh from being made arbitrarily fine in order to improve the accuracy of the identification.

We can now consider a family of polygonal functions, $\hat{D}$, associated with the partition of J. Each member of the family is characterized by its values at the nodes μ_k; i.e. ,for $d_k = \hat{D}(\mu_k)$. More precisely, we define

$$\hat{D}(u) = \sum_{k=1}^{M} d_k \lambda_k(u) \tag{2.10}$$

where

$$\lambda_k(u) = \begin{cases} 0, & \text{if } u < \mu_{k-1} \\ \dfrac{u - \mu_{k-1}}{\mu_k - \mu_{k-1}} & \text{if } \mu_{k-1} \le u \le \mu_k \\ 1, & \text{otherwise.} \end{cases} \tag{2.11}$$

Equivalently, we could write, for $1 \le k \le M$,

$$\hat{D}(u) = d_{k-1} + (d_k - d_{k-1})\lambda_k(u) \quad \text{for} \quad \mu_{k-1} \le u \le \mu_k \tag{2.12}$$

We will introduce several notations:

- $\hat{D}(u) = P_M[d_0, d_1, \dots , d_M]$ denotes the polygonal coefficient given by (2.10) based on nodal values $[d_0, d_1, \dots , d_M]$.
- $u(x, t; D, f)$ denotes the solution of the direct problem (2.1) with coefficient D and data, f.
- $\phi(x, t; D, \theta)$ denotes the solution of the adjoint problem (2.6) with coefficient $D(x, t) \overset{def}{=} D(\mu(x, t))$ and data, $\theta(t)$.
- $\psi(x, t; D, \beta)$ denotes the solution of the adjoint problem (2.8) with coefficient $D(x, t) \overset{def}{=} D(\mu(x, t))$ and data, $\beta(t)$

For a given $f(t)$ satisfying (2.2), an unknown coefficient $D = D(u)$ satisfying (2.3) and measured flux data $g(t) = \Phi[f, D]$, we assume there is a fixed outer partition, $\Pi = \{0 = \mu_0 < \mu_1 < \cdots < \mu_M = f(T)\}$ of J. Then we will define a polygonal coefficient approximation to D by the following recursive algorithm based on the data pair $\{f(t), g(t)\}$:

- d_0 is assumed to be given
- for $k = 1, 2, ...$ d_k is determined from $d_0, d_1, ..., d_{k-1}$ by

$$(d_k - d_{k-1}) \int_{T_{k-1}}^{T_k} \lambda_k(u_2)\ \partial_x u_2\ \partial_x \phi\ dx\ dt$$
$$= - \int_{T_{k-1}}^{T_k} (g(t) - g_2(t))\ \theta(t)\ dt, \quad (2.13)$$

where

$$D_1(u) = P_M\left[d_0, d_1, \ldots, d_{k-1}, d_k\right]$$
$$D_2(u) = P_M\left[d_0, d_1, \ldots, d_{k-1}, d_{k-1}\right]$$
$$u_2(x,t) = u(x,t; D_2, f),$$
$$g_2(t) = -D_2(f(t))\partial_x u_2(0,t)\ 0 \le x \le 1,\ 0 \le t \le T_k,$$
$$\phi(x,t) = \phi(x,t; D_1, f(T-t)),\ for\ 0 \le x \le 1,\ 0 \le t \le T_k.$$

The approximation of $D(u)$ based on data pair $(f, h), \{f(t), h(t)\}$, is analogous. We can show then,

Lemma 2.2.1. *For $f(t)$ satisfying (2.2), for coefficient D satisfying (2.3) and for a fixed partition, $\Pi = 0 = \mu_0 < \mu_1 < \cdots < \mu_M = f(T)$ of J, let the nodal values $[d_0, d_1, \ldots, d_M]$ be determined by the algorithm (2.13). Then for $k = 1, 2, ..., M$,*

$$|D(\mu_k) - d_k| \le C\,|\mu_k - \mu_{k-1}| \quad (2.14)$$

Proof. We are going to assume that the initial nodal value, $D(\mu_0) = D(f(0)) = d_0$, is known and that the remaining values $d_1, \ldots, d_M$ are determined by the algorithm (2.13). Consider first, the value d_1. If we apply the identity (2.7) with $\tau = T_1$, and

- on $J_1 = [\mu_0, \mu_1]$, $D_1(u) = P_M[d_0, d_1]$, and $D_2(u) = P_M[d_0, d_0]$,
- on $Q_1 = (0,1)\times(0,T_1)$ $u_1(x,t) = u(x,t; D_1, f)$ and $u_2(x,t) = u(x,t; D_2, f)$

then we have

$$\int_0^{T_1}\int_0^1 (D_1(u_2) - D_2(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt = \int_0^{T_1}(g(t) - g_2(t))\theta(t)\,dt.$$

Here $g(t)$ is the measured flux data and $g_2(t)$ is the output generated by solving (2.1) with the coefficient $D(u) = D_2(u)$; i.e., $g_2 = \Phi[f, D_2]$. The functions $\theta(t)$ and $\phi(x,t)$ denote the data and solution respectively for the g-adjoint problem. Since the function $f(t)$ in the direct problem satisfies (2.2), it follows from Lemma (2.1.1)(a) that u_2 satisfies

$$f(0) = \mu_0 \leq u_2(x,t) \leq \mu_1 = f(T_1) \text{ for } (x,t) \in (0,1)\times(0,T_1).$$

Then according to (2.12), for $u \in J_1 = \mu_0 \leq u \leq \mu_1$,

$$\begin{aligned} D_1(u) &= d_0 + (d_0 - d_0)\lambda_1 \\ D_2(u) &= d_0 + (d_1 - d_0)\lambda_1 \\ \text{and so} \qquad D_1(u_2) - D_2(u_2) &= (d_1 - d_0)\lambda_1(u_2). \end{aligned}$$

Note that for each nodal value, μ_k, $0 \leq k \leq M$, we have $u_2(x_k(t), t) = \mu_k$ along some curve $x = x_k(t)$, with $x_k(0) = \mu_k$ and $x_k(\tau_k) = 1$ for some $\tau_k > \tau_{k-1} > \cdots > \tau_1 > 0$. Examples of such curves are shown in figure 2.1.

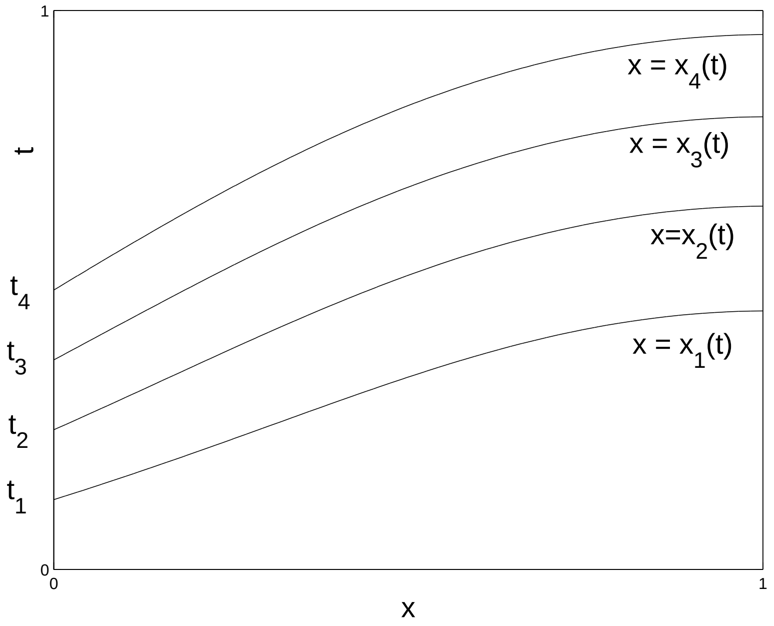


Figure 2.1: Isoclines

Then we have $u_2(x(t), t) = \mu_0$ along a curve $x = x_0(t)$, with $x_0(0) = 0$ and $x_0(\tau_1) = 1$ for some $\tau_1 > 0$. We suppose further that T_1 is sufficiently small that $0 < x_0(T_1) < 1$. Then

$$\lambda_1(u_2(x,t)) = \begin{cases} \dfrac{u_2(x,t) - \mu_0}{\mu_1 - \mu_0} & \text{if } 0 \le x \le x_0(t),\ 0 \le t \le T_1 \\ 0 & \text{if } x > x_0(t)\ 0 \le t \le T_1 \end{cases}$$

and the integral identity reduces to

$$(d_1 - d_0) \int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2)\, \partial_x u_2\, \partial_x \phi\, dx\, dt = \int_0^{T_1} (g(t) - g_2(t))\theta(t)\, dt;$$

i.e.,

$$d_1 = d_0 + \frac{\displaystyle\int_0^{T_1} (g(t) - g_2(t))\theta(t)\, dt}{\displaystyle\int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2)\ \partial_x\ u_2\ \partial_x \phi\ dx\ dt}.$$

This equation defines the first unknown nodal value d_1. Now we will establish the relationship between d_1 and $D(\mu_1)$. It follows from (2.7) that

$$\begin{aligned}\int_0^{T_1}\int_0^{x_0(t)} & (D(u_2) - D_2(u_2))\ \partial_x u_2\ \partial_x \phi\ dx\ dt \\ &= \int_0^{T_1} (g(t) - g_2(t))\theta(t)\ dt \\ &= (d_1 - d_0)\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\ \partial_x u_2\ \partial_x \phi\ dx\ dt.\end{aligned}$$

Let $\hat{D}_M(u)$ denote the polygonal coefficient on the partition Π which satisfies $\hat{D}_M(\mu_k) = D(\mu_k)$ for all k. Note that this coefficient does not, in general, generate the given measured data, $g(t)$, and is not then the polygonal coefficient with nodal values $\{d_k\}$ generated by the algorithm. However, these coefficients are related as follows,

$$\begin{aligned}\int_0^{T_1}\int_0^{x_0(t)} & (D(u_2) - D_2(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &= \int_0^{T_1}\int_0^{x_0(t)} (D(u_2) - \hat{D}_M(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &+ \int_0^{T_1}\int_0^{x_0(t)} (\hat{D}_M(u_2) - D_2(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &= \int_0^{T_1}\int_0^{x_0(t)} (D(u_2) - \hat{D}_M(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &+ (D(\mu_1) - d_0)\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \phi\,dx\,dt.\end{aligned}$$

By combining these two expressions it follows that

$$\begin{aligned}(d_1 - D(\mu_1)) & \int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &= \int_0^{T_1}\int_0^{x_0(t)} (D(u_2) - \hat{D}_M(u_2))\,\partial_x u_2\,\partial_x \phi\,dx\,dt \\ &\leq \max_{J_1} \left|D - \hat{D}_M\right| \left|\int_0^{T_1}\int_0^{x_0(t)} \partial_x u_2\,\partial_x \phi\,dx\,dt\right|.\end{aligned}$$

Now

$$\max_{J_1} \left| D - \hat{D}_M \right| = \left| D(\mu_*) - \hat{D}_M(\mu_*) \right| \text{for some } \mu_* \in J_1.$$

But

$$\begin{aligned} \left| D(\mu_*) - \hat{D}_M(\mu_*) \right| &\leq \left| D(\mu_*) - D(\mu_0) \right| + \left| D(\mu_0) - \hat{D}_M(\mu_*) \right| \\ &\leq K \left| \mu_* - \mu_0 \right| + \left| \hat{D}_M(\mu_0) - \hat{D}_M(\mu_*) \right|. \end{aligned}$$

In addition, $\left| \hat{D}_M(\mu_0) - \hat{D}_M(\mu_*) \right| \leq K \left| \mu_* - \mu_0 \right|$, and

$$\left| D(\mu_*) - \hat{D}_M(\mu_*) \right| \leq 2K \left| \mu_* - \mu_0 \right|.$$

Then

$$|d_1 - D(\mu_1)| \leq 2K \frac{\left| \int_0^{T_1} \int_0^{x_0(t)} \partial_x u_2 \, \partial_x \phi \, dx \, dt \right|}{\left| \int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2) \, \partial_x u_2 \, \partial_x \phi \, dx \, dt \right|} \left| \mu_* - \mu_0 \right|.$$

Since it is clear that for some $\lambda_1^*, 0 < \lambda_1^* < 1$,

$$\int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2) \, \partial_x u_2 \, \partial_x \phi \, dx \, dt = \lambda_1^* \int_0^{T_1} \int_0^{x_0(t)} \partial_x u_2 \, \partial_x \phi \, dx \, dt$$

we find

$$1 \leq \frac{\left| \int_0^{T_1} \int_0^{x_0(t)} \partial_x u_2 \, \partial_x \phi \, dx \, dt \right|}{\left| \int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2) \, \partial_x u_2 \, \partial_x \phi \, dx \, dt \right|} \leq \frac{1}{\lambda_1^*} < \infty.$$

Then,

$$| \, d_1 - D(\mu_1)| \leq \frac{2K}{\lambda_1^*} \left| \mu_* - \mu_0 \right| \leq C_1 \left| \mu_1 - \mu_0 \right|.$$

This is the result (2.14) for $k = 1$.

In determining the succeeding values d_k, we assume $d_0, d_1, ..., d_{k-1}$ are known and we let,

- on $[\mu_0, \mu_k]$, $D_1(u) = P_M[d_0, d_1, \dots , d_{k-1}, d_k]$,

  and $D_2(u) = P_M[d_0, d_1, \dots , d_{k-1}, d_{k-1}]$,

- on $Q_k = (0, 1) \times (0, T_k)$, $u_1(x, t) = u(x, t; D_1, f)$,

  and $u_2(x, t) = u(x, t; D_2, f)$

Then $D_1(u)$ and $D_2(u)$ are identical on $[\mu_0, \mu_{k-1}]$ and only differ on $J_k = [\mu_{k-1}, \mu_k]$ where we have

$$\begin{aligned} D_1(u) &= d_{k-1} + (d_k - d_{k-1})\lambda_k && \text{for } \mu_{k-1} \le u \le \mu_k, \\ D_2(u) &= d_{k-1} && \text{for } \mu_{k-1} \le u \le \mu_k, \end{aligned}$$

Then

$$\begin{aligned} &\int_0^{T_k} \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \phi \, dx\, dt \\ &\qquad = \int_{T_{k-1}}^{T_k} \int_0^1 (D_1(u_2) - D_2(u_2))\, \partial_x u_2\, \partial_x \phi \, dx\, dt \\ &\qquad =(d_k - d_{k-1}) \int_{T_{k-1}}^{T_k} \int_0^{x_{k-1}(t)} \lambda_k(u_2)\, \partial_x u_2\, \partial_x \phi \, dx\, dt. \end{aligned}$$

and we have

$$\begin{aligned} (d_k - d_{k-1}) \int_{T_{k-1}}^{T_k} \int_0^{x_{k-1}(t)} &\lambda_k(u_2)\, \partial_x u_2\, \partial_x \phi \, dx\, dt \\ &= \int_{T_{k-1}}^{T_k} (g(t) - g_2(t))\theta(t)\, dt, \end{aligned}$$

as prescribed by (2.13). Now we proceed as in the first part of the proof to show that

$$|d_k - D(\mu_k)| \le C|\mu_k - \mu_{k-1}|.$$

The proof of the analogous result based on the data $\{f(t_k), h(t_k)\}$ proceeds similarly. □

For d_0 fixed and $d_1 > 0$, let $P_1(d_1)(u) = d_0\,\rho_0(u) + d_1\,\lambda_1(u)$ for $u \in J_1$. Then P_1 is a mapping from $[0, \infty]$ into a one dimensional subspace of $W(J_1)$. It follows from (2.13) in the case $k = 1$ that

$$\begin{aligned}
&\left\langle \Delta D(u_2),^t \delta\Phi\left[P_1(d_1), P_1(d_0)\right](\theta)\right\rangle \\
&\qquad = \left\langle (d_1 - d_0)\lambda_1(u_2),^t \delta\Phi\left[P_1(d_1), P_1(d_0)\right](\theta)\right\rangle \\
&\qquad = (d_1 - d_0)\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x\phi\,dx\,dt
\end{aligned}$$

This means that the double integral in the expression above is a representation for the derivative with respect to the parameter d, of the coefficient-to-data mapping, Φ, restricted to the one dimensional subspace of $W(J_1)$. Since the double integral can be shown to be nonzero, it follows that the restricted input/output mapping is locally approximately invertible. Lemma (2.2.2) asserts that, if we are given the data, $\{f(t_k), g(t_k)\}$ or $\{f(t_k), h(t_k)\}$, then we can compute the nodal values $\{d_k\}$ which reproduce the measured data in the sense of (2.13) and that these nodal values approach the nodal values of the "true coefficient" $D(u_1)$, as the mesh size of the outer mesh decreases. However, this conclusion ignores certain difficulties:

- it is not possible to know the coefficient $D_1(\mu(x,t))$ in the adjoint problems since D_1 is the coefficient we wish to identify and μ is an indeterminate point between u_1 *and* u_2. This means we can only approximate the solution to the adjoint problem and this will have an influence on the conclusions of lemma (2.2.2).

- the integrals in the identity can only be approximated by numerical integrations for which there is only a limited degree of refinement possible. This may further interfere with the agreement between d_k and $D(\mu_k)$.

We will consider both of these effects, starting with the effect of the approximate adjoint solution.

Note first, that the algorithm (2.13) asserts that in determining the nodal value μ_k, it is necessary to solve the adjoint problem only on the strip $S_k = \{(0,1) \times (T_{k-1}, T_k)\}$. Let $\hat{\phi}(x,t)$ denote the adjoint solution we compute using a convenient approximation for the unknown coefficient $D_1(\mu(x,t))$ on this strip. For example, suppose the coefficient in the g-adjoint problem is chosen to have the known constant value, d_{k-1}; i.e.,

$$D_1(\mu(x,t)) = d_{k-1} \qquad \mu(x,t) \in J_k = [\mu_{k-1}, \mu_k] \,.$$

Then if we replace ϕ in (2.13) by $\hat{\phi}(x,t)$, we can denote the resulting computed nodal value by $\hat{d}_k$. Note that with this choice for the coefficient, there is now no difficulty in solving the adjoint problem (2.6) for $\hat{\phi}$ on the strip, $(0,1) \times [T_{k-1}, T_k]$ and proceeding to compute $\hat{d}_k$ using (2.13). It remains to be seen how the values $\hat{d}_k$ compare to the values d_k. We begin with a lemma.

Lemma 2.2.2. *Let $f(t)$ satisfy (2.2), let coefficient D satisfy (2.3) and let* Π *denote a fixed partition,* $\Pi = \{\mu_k = f(T_k): \ k = 0, 1, ..., M\}$ *of J. For k between 1 and M consider the following adjoint problem,*

$$\begin{aligned}
\partial_t \phi(x,t) + c\ \partial_{xx}\phi(x,t) &= 0, && \in S_k \\
\phi(x, T_k) &= 0, && x \in (0,1) \\
\phi(0,t) &= f(T_k - t) && t \in (T_{k-1}, T_k) \\
\partial_x \phi(1,t) &= 0, && t \in (T_{k-1}, T_k)
\end{aligned}$$

Suppose $\{\phi_i, c_i\},\ i = 1,2$ *denote two solutions to the adjoint problem corresponding to distinct choices of the coefficient c. In particular, suppose* $\phi_1 = \phi(x,t,c_1,\theta)$ *for the constant* $c_1 = d_{k-1}$, *while* $\phi_2 = \phi(x,t,c_2,\theta)$ *corresponding to the choice,* $c_2(x,t) = D(\mu(x,t))$, *where* $\mu(x,t)$ *denotes a function that is continuous on the strip* $S_k = (0,1) \times (T_{k-1}, T_k)$ *with values in* $J_k = [\mu_{k-1}, \mu_k]$. *Then*

$$||\,\partial_x(\phi_1 - \phi_2)||_{L^2(S_k)} \le C\ |\mu_k - \mu_{k-1}|$$

Proof. Begin by noting that $\Delta\phi = \phi_1 - \phi_2$ satisfies,

$$\begin{aligned} \partial_t(\Delta\phi) + c_1\ \partial_{xx}(\Delta\phi) &= (c_2 - c_1)\,\partial_{xx}\phi_2, && (x,t) \in S_k \\ \Delta\phi(x,T_k) &= 0, && x \in (0,1) \\ \Delta\phi(0,t) &= 0, && \in (T_{k-1}, T_k), \\ \partial_x(\Delta\phi)(1,t) &= && 0, t \in (T_{k-1}, T_k), \end{aligned}$$

and if ψ denotes an arbitrary test function, then

$$\iint\limits_{S_k} \{\partial_t(\Delta\phi) + c_1\ \partial_{xx}(\Delta\phi)\}\,\partial_x\psi\,dxdt = \iint\limits_{S_k} \{-\Delta c\ \partial_{xx}\phi_2\}\,\partial_x\psi\,dxdt.$$

Integration by parts yields,

$$\begin{aligned} \iint\limits_{S_k} \partial_t(\Delta\phi)\partial_x\psi\,dxdt = &\iint\limits_{S_k} \partial_x(\Delta\phi)\partial_t\psi\,dxdt \\ &+ \int_0^1 \Delta\phi\,\partial_x\psi|_{t=0}^{t=T}dx \\ &- \int_{T_{k-1}}^{T_k} \Delta\phi\,\partial_x\psi|_{x=0}^{x=1}dt, \end{aligned}$$

and

$$\begin{aligned} \iint\limits_{S_k} \partial_{xx}(\Delta\phi)\,\partial_x\psi\,dxdt = &-\iint\limits_{S_k} \partial_x(\Delta\phi)\partial_{xx}\psi\,dxdt \\ &+ \int_{T_{k-1}}^{T_k} \partial_x(\Delta\phi)\,\partial_x\psi|_{x=0}^{x=1}dt, \end{aligned}$$

s0

$$\iint\limits_{S_k} \partial_x(\Delta\phi)[\partial_t\psi - c_1\partial_{xx}\psi]\,dxdt + \int_0^1 \Delta\phi\,\partial_x\psi|_{t=T_{k-1}}^{t=T_k}dx$$

$$- \int_{T_{k-1}}^{T_k} \Delta\phi\,\partial_x\psi|_{x=0}^{x=1}dt + c_1\int_{T_{k-1}}^{T_k} \partial_x(\Delta\phi)\,\partial_x\psi|_{x=0}^{x=1}dt$$

$$= \iint\limits_{S_k} \{-\Delta c\ \partial_{xx}\phi_2\}\,\partial_x\psi\,dxdt$$

Now choose the test function ψ to satisfy

$$\partial_t\psi - c_1\partial_{xx}\psi = \partial_x(\Delta\phi), \qquad (x,t)\in S_k$$

$$\psi(x,T_{k-1}) = 0, \qquad x\in(0,1)$$

$$\partial_x\psi(0,t) = 0, \quad \psi(1,t) = 0, \qquad t\in(T_{k-1},T_k)$$

Then the previous integral identity reduces to

$$\iint\limits_{S_k}[\partial_x(\Delta\phi)]^2dxdt = \iint\limits_{S_k}(c_2-c_1)\,\partial_{xx}\phi_2\,\partial_x\psi\,dxdt.$$

Now, ψ is the solution to a linear problem with constant coefficients so it can be expressed in terms of a Green's function, $\Gamma(x,t)$,

$$\psi(x,t) = \int_{T_{k-1}}^{t}\int_0^1 \Gamma(x-y,t-\tau)\,\partial_x(\Delta\phi)(y,\tau)\,dyd\tau,\ (x,t)\in S_k,$$

and

$$\partial_x\psi(x,t) = \int_{T_{k-1}}^{t}\int_0^1 \partial_x\Gamma(x-y,t-\tau)\,\partial_x(\Delta\phi)(y,\tau)\,dyd\tau.$$

Then for all $(x,t)\in S_k$,

$$\begin{aligned}|\,\partial_x\psi(x,t)| &\le \int_{T_{k-1}}^{t}\int_0^1 |\partial_x\Gamma(x-y,t-\tau)\,\partial_x(\Delta\phi)(y,\tau)|\,dyd\tau,\\ &\le \left(\int_{T_{k-1}}^{T_k}\int_0^1 |\partial_x\Gamma(x-y,t-\tau)|^2dyd\tau\right)^{1/2}\\ &\quad\times\left(\int_{T_{k-1}}^{T_k}\int_0^1 |\partial_x(\Delta\phi)(y,\tau)|^2\,dyd\tau\right)^{1/2}\end{aligned}$$

and

$$\max_{(x,t)\in S_k} |\,\partial_x \psi(x,t)| \leq C \;\; ||\,\partial_x(\Delta\phi)||_{L^2(S_k)}\,.$$

Then it follows that

$$\begin{aligned}\iint\limits_{S_k} [\partial_x(\Delta\phi)]^2 \; dx \; dt &= \left| \iint\limits_{S_k} (c_2 - c_1)\,\partial_{xx}\phi_2\,\partial_x\psi\,dxdt \right| \\ &\leq \max_{S_k} |\Delta c(x,t)| \iint\limits_{S_k} \; |\partial_{xx}\phi_2\,\partial_x\psi| \; dxdt \\ &\leq \max_{S_k} |\Delta c(x,t)| \; ||\,\partial_{xx}\phi_2\,||_{L^1} \; ||\,\partial_x\psi||_\infty\end{aligned}$$

and

$$||\,\partial_x(\Delta\phi)||_{L^2(S_k)} \leq C \;\; \max_{S_k} |\Delta c(x,t)|\,.$$

Also

$$\begin{aligned}\max_{S_k} |\Delta c(x,t)| = \;& \max_{S_k} |d_{k-1} - D(\mu(x,t))| \\ &\leq |d_{k-1} - D(\mu_{k-1})| + \max_{S_k} |D(\mu_{k-1}) - D(\mu(x,t))| \\ &\leq 2K\,|\,\mu_k - \mu_{k-1}|\,.\end{aligned}$$

Then, it follows that,

$$||\,\partial_x(\Delta\phi)||_{L^2(S_k)} \leq C \; |\mu_k - \mu_{k-1}|\,.$$

□

Now we will use this estimate in considering the effect of using the approximate adjoint solution in the determination of the first nodal value, d_1. It follows from (2.13) that the difference between the value, d_1, computed using the correct but unknown adjoint solution and the value, $\hat{d}_1$, computed using an incorrect but computable adjoint solution is given by,

$$
\begin{aligned}
\hat{d}_1 - d_1 =& \frac{\displaystyle\int_0^{T_1} (g(t) - g_2(t))\theta(t)\,dt}{\displaystyle\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \hat{\phi}\,dx\,dt} \\
&- \frac{\displaystyle\int_0^{T_1} (g(t) - g_2(t))\theta(t)\,dt}{\displaystyle\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \phi\,dx\,dt} \\
=& \frac{(g - g_2, \theta)}{II(\hat{\phi})} - \frac{(g - g_2, \theta)}{II(\phi)} \\
=& (g - g_2, \theta)\left\{\frac{1}{II(\hat{\phi})} - \frac{1}{II(\phi)}\right\} \hat{d}_1 - d_1 = \quad (d_1 - d_0)\left\{\frac{II(\phi) - II(\hat{\phi})}{II(\hat{\phi})}\right\}
\end{aligned}
$$

Here

$$
II(\hat{\phi}) = \int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \hat{\phi}\,dx\,dt.
$$

We wish to show that as the outer mesh is refined, the discrepancy $II(\phi) - II(\hat{\phi})$ that is due to solving the adjoint problem with the wrong coefficient decreases to zero. On the other hand, $II(\hat{\phi})$ also decreases toward zero as the mesh is refined. To see whether $II(\hat{\phi})$ decreases more or less rapidly than $II(\phi) - II(\hat{\phi})$, it is necessary to examine the asymptotic behavior of $II(\hat{\phi})$. We assume that $x_0(T_1) < 1$ since if this is not the case, we can always refine the outer partition to shrink the width of the strip S_1 so as to make it true. Then the domain of integration for $II(\hat{\phi})$ is the approximately triangular region $\{0 \leq x \leq x_0(t), 0 \leq t \leq T_1\}$. An exact analysis of the asymptotic rate of convergence of $II(\hat{\phi})$ as T_1 tends to zero is difficult, but if we assume that $f(t) = At$ for a positive constant

A, then it is possible to solve explicitly for $u_2(x,t)$ and $\hat{\phi}(x,t)$. Using arguments like those in [1], one finds that $g(t) = -D(u_1(0,t))\,\partial_x u_1(0,t)$ and $g_2(t) = -d_k\partial_x u_2(0,t)$ behave asymptotically like $\sqrt{t}$.

This leads to

$$\begin{aligned}\int_0^{T_1} (g(t) - g_2(t))\theta(t)\ dt &= \int_0^{T_1} (g(t) - g_2(t))\,A(T_1 - t)\ dt \\ &\approx C\,T_1^{5/2}\end{aligned}$$

A similar crude estimate for $\partial_x u_2\,\partial_x\hat{\phi}$ on $0 \le x \le 1,\ 0 \le t \le T_1$, is the following

$$\partial_x u_2\,\partial_x\hat{\phi}(x,t) \approx \sqrt{t}\,m(x)\,\sqrt{T_1 - t}\,m(x)$$

where $m(x)$ denotes a decreasing function with $m(0) = 1$ and $m(1) = 0$. In addition, for T_1 small, one can suppose $x_0(t) \approx at$ for a positive constant a, and this leads to

$$\begin{aligned}II(\hat{\phi}) &= \int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x\hat{\phi}\ dx\ dt \\ &\approx \int_0^{T_1}\int_0^{at} \frac{u_2(x,t)}{AT_1}\sqrt{t}\,m(x)\,\sqrt{T_1 - t}\,m(x)\,dx\,dt \\ \textit{i.e.}\quad II(\hat{\phi}) &\approx CT_1^{5/2}. \qquad (2.15)\end{aligned}$$

Since this estimate (2.15) is rather rough, the quantity $II(\hat{\phi})$ was computed numerically for a sequence of values for T_1 decreasing to zero. The result of this numerical asymptotic estimate supported the estimate (2.15) which asserts that $II(\hat{\phi})$ decreases like the $\frac{5}{2}$ power of T_1 as T_1 tends to zero.

Now

$$\hat{d}_1 - d_1 = (d_1 - d_0)\left\{\frac{II(\phi) - II(\hat{\phi})}{II(\hat{\phi})}\right\}$$

and,

$$\begin{aligned}\left| II(\phi) - II(\hat{\phi}) \right| &= \left| \int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2)\, \partial_x u_2 \, [\partial_x \phi - \partial_x \hat{\phi}] \, dx \, dt \right| \\ &\le C(T_1) \, || \, \partial_x(\Delta\phi) ||_{L^2(S_1)} \le C(T_1) \; |\mu_1 - \mu_0| \, .\end{aligned}$$

Also,

$$| \, d_1 - d_0| = | \, D(\mu_1) - D(\mu_0)| \le K \, |\mu_1 - \mu_0| \, ,$$

and hence

$$\left| \hat{d}_1 - d_1 \right| \le |d_1 - d_0| \left| \frac{II(\phi) - II(\hat{\phi})}{II(\hat{\phi})} \right| \le \frac{K \, C(T_1)}{II(\hat{\phi})} \; |\mu_1 - \mu_0|^2 \, .$$

Then for T_1 sufficiently small,

$$\begin{aligned}\left| \hat{d}_1 - d_1 \right| &\le \frac{K \, C(T_1)}{C T_1^{5/2}} \; |\mu_1 - \mu_0|^2 \\ &\le \frac{K \, f'(\tau)^2}{C} T_1^{-1/2} \text{ for some } \tau > 0.\end{aligned}$$

In general, we have

Lemma 2.2.3. *For $f(t) = At, \; A > 0$, for coefficient D satisfying (2.3) and for a fixed partition, $\Pi = \{\mu_k = AT_k : k = 0, 1, ..., M\}$ of J, fix k between 1 and M. Let $\hat{\phi} = \phi(x, t, d_{k-1}, A(T_k - t))$ and $\phi = \phi(x, t, c, A(T_k - t))$ corresponding to the coefficients, d_{k-1} and $c(x, t) = D(\mu(x, t))$, respectively. Finally, let $\hat{d}_k$ and d_k denote the nodal values determined from (2.13) using the values $[d_0, d_1, ..., d_{k-1}]$ and the adjoint solutions $\hat{\phi}$ and ϕ, respectively. Then*

$$\left| \hat{d}_k - d_k \right| \le \frac{K}{II(\hat{\phi})} \; |\mu_k - \mu_{k-1}|^2 \le \frac{K \, f\,'(\tau)^2}{C} |T_k - T_{k-1}|^{-1/2}.$$

This lemma implies that the error introduced into the identification by solving the adjoint problem with an approximate coefficient has an increasing effect as the outer mesh is refined. As the mesh is refined, the discrepancy $II(\phi) - II(\hat{\phi})$ does tend to zero like the square of the mesh size. However, as the mesh size tends to zero, we find also that $II(\hat{\phi})$, which can be viewed as an approximation to the Gateaux derivative of the mapping Φ restricted to a one dimensional subspace of $W(J_k)$, tends to zero even faster, (like the $\frac{5}{2}$ power of the mesh size). It is likely that the means of approximating the adjoint solution could be improved so that $II(\phi) - II(\hat{\phi})$ would approach zero sufficiently rapidly that $\left|\hat{d}_k - d_k\right|$ would tend to zero as the mesh size goes to zero. However, the next result will show that such an improvement does not improve the convergence of the approximate solution.

We wish finally to consider the effect of numerical integration errors on the calculation of $\hat{d}_k$. We begin by considering $k = 1$. We have,

$$\hat{d}_1 = d_0 + \frac{\int_0^{T_1}(g(t) - g_2(t))\theta(t)\,dt.}{\int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x\hat{\phi}\,dx\,dt} = d_0 + \frac{I(g - g_2)}{II(\hat{\phi})}$$

and

$$\hat{d}_1^* = d_0 + \frac{I^*(g - g_2)}{II^*(\hat{\phi})}$$

where $I^*(g - g_2)$ and $II^*(\hat{\phi})$ denote, respectively, the computed results using the inner mesh to numerically approximate the corresponding exact single and double integrals. Then

$$\begin{aligned}\hat{d}_1^* &= d_0 + \frac{I^*(g - g_2) - I(g - g_2) + I(g - g_2)}{II^*(\hat{\phi}) - II(\hat{\phi}) + II(\hat{\phi})} \\ &= d_0 + \frac{I(g - g_2)}{II(\hat{\phi})}\frac{1 + \varepsilon_1}{1 + \varepsilon_2}\end{aligned}$$

where

$$\varepsilon_1 = |\frac{I-I^*}{I}| \text{ and } \varepsilon_2 = |\frac{II-II^*}{II}|.$$

Now $$\frac{1+\varepsilon_1}{1+\varepsilon_2} \approx 1+\varepsilon_1+\varepsilon_2$$

so

$$\hat{d}_1^* = d_0 + \frac{I(g-g_2)}{II(\hat{\phi})}\frac{1+\varepsilon_1}{1+\varepsilon_2} \approx d_0 + \frac{I(g-g_2)}{II(\hat{\phi})}(1+\varepsilon_1+\varepsilon_2),$$

and

$$\left|\hat{d}_1 - \hat{d}_1^*\right| \leq \left|\frac{I(g-g_2)}{II(\hat{\phi})}\right|(\varepsilon_1+\varepsilon_2) = \left|\hat{d}_1 - d_0\right|(\varepsilon_1+\varepsilon_2).$$

The numerical integration errors are estimated by terms of the form,

$$|I-I^*| \leq C(\Delta t)^2 \qquad \text{for } \Delta t = \text{inner mesh size}$$

and $$|II-II^*| \leq C(\Delta x\,\Delta t) = C(\Delta t)^2$$

Use of higher order integration schemes is limited by the fact that reducing the mesh size of the outer or $J-mesh$ in order to achieve identification accuracy absorbs $I-mesh$ node points into the $J-mesh$ leaving only enough points in the inner mesh to perform low order numerical integrations.

It follows from (2.2) and (2.15) that

$$I = \int_0^{T_1}(g(t)-g_2(t))\,A(T_1-t)dt \approx T_1^{5/2},$$
$$II = \int_0^{T_1}\int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x\phi\,dx\,dt \approx T_1^{5/2}$$

Then, since $T_1 = k\Delta t$, we find

$$\left|\hat{d}_1 - \hat{d}_1^*\right| \leq \left|\frac{I(g-p_M)}{II(\hat{\phi})}\right|(\varepsilon_1+\varepsilon_2)$$
$$\leq \left|\hat{d}_1 - d_0\right|\frac{C_1(\Delta t)^2}{C_2\,(k\Delta t)^{5/2}} \leq C(\Delta t)^{-1/2}.$$

More generally, we have

Lemma 2.2.4. Under the conditions of (2.2.3), let $\hat{d}_k^*$ reflect the error induced in $\hat{d}_k$ by numerically approximating the integrals needed for (2.13).

Then, as the (inner and outer) mesh size tends to zero,

$$\left|\hat{d}_k - \hat{d}_k^*\right| \leq C(\Delta t)^{-1/2}$$

This estimate suggests that as the outer mesh is refined in order to improve the accuracy of the identification of the nodal values of $D(u_1)$, more and more node points of the inner mesh are absorbed into the outer mesh, resulting in numerical integration errors, $|I - I^*|$ and $|II - II^*|$, that are of order Δt^2. At the same time, the approximate Gateaux derivative $II(\hat{\phi})$ tends to zero like $\Delta t^{5/2}$ so the effect of approximating the integrals becomes magnified at Δt tends to zero. Evidently, at some point the values of the integrals used to compute d_k become of the same order of magnitude as the numerical integration errors and the computation then no longer contains information. Further decreasing the mesh size then only increases the error.

Finally, we can combine lemmas 2.2.1, 2.2.3 and 2.2.4 to write

$$\begin{aligned}\left|D(\mu_k) - \hat{d}_k^*\right| &= \left|D(\mu_k) - d_k + d_k - \hat{d}_k + \hat{d}_k - \hat{d}_k^*\right| \\ &\leq |D(\mu_k) - d_k| + \left|d_k - \hat{d}_k\right| + \left|\hat{d}_k - \hat{d}_k^*\right|\end{aligned}$$

and,

$$\left|D(\mu_k) - \hat{d}_k^*\right| \leq C_1\Delta t + C_2(\Delta t)^{-1/2}. \tag{2.16}$$

Evidently the error in identifying d_k does not tend to zero as Δt tends to zero but is minimized by an optimal Δt different from zero.

Chapter 3

ONE PARAMETER NUMERICAL EXPERIMENTS

3.1 One parameter problem

In this section, we present a numerical implementation of a recovery algorithm and analyze this process via a series of numerical experiments. We consider several numerical experiments designed to gain insight into the recovery of the unknown coefficient $D(u)$ in the one parameter quasilinear conduction diffusion equation given by

$$\partial_t u(x,t) = \partial_x (D(u) \partial_x u(x,t)) \text{ on } 0 < x < L, 0 < t < T. \tag{3.1}$$

We choose to interpret this model as the heat equation, and as such will refer to the coefficient $D(u)$ as the conductivity coefficient. The method presented here is based on the integral identities

$$\int_0^\tau G^*(t)[g(t) - g_2(t)]\, dt = \int_0^\tau \int_0^L (D(u_2) - D_2(u_2)) \partial_x \phi \, \partial_x u_2 \, dx \, dt, \tag{3.2}$$

which we refer to as the g-integral identity, and

$$\int_0^\tau H^*(t)[h(t) - h_2(t)]\, dt = \int_0^\tau \int_0^L (D(u_2) - D_2(u_2)) \partial_x \phi \, \partial_x u_2 \, dx \, dt, \tag{3.3}$$

which we call the h-integral identity.

The algorithm constructed in this chapter creates a linear polygonal approximation to the unknown coefficient $D(u)$ utilizing observations of the system. Although it might be possible to place thermocouples over the length of the media and record this interior information over time, instead we restrict the observed measurement to take place on the boundary of the media, at $x = 0$ and $x = L$. While not the only observable boundary measurements, the flux $g(t) = D(u(0,t)\partial_x u(0,t)$ and the state $h(t) = u(L,t)$ are easily obtained. Both maps $\Phi[f, D] \to g$ and $\Psi[f, D] \to h$ have been shown to be continuous and invertible under monotone forcing via the integral identities.

We begin with a description of the numerical details.

3.2 Numerical methodology

The nonlinear PDE (3.1) was discretized on a non-uniform space grid. The resulting system of ODEs was then submitted to a implicit time integration scheme. This use of robust and sophisticated implicit schemes allowed control of many aspects of the numerics, such as relative and absolute error and the use of backward differentiation formulas. The Matlab ODE suite of solvers were used. The piecewise linear coefficient was passed as a call-out table in the state variable, which was then evaluated (via linear interpolation) by Matlab in each evaluation needed for the time integration. In addition, the numerical solution was returned in a ‘structure’ format, which allowed high order numerical interpolation schemes to be used to evaluate this solution between computed nodes. This allowed the solution to be projected onto a wide range of time nodes easily.

The standard finite difference scheme was used. Using a space discretization over the grid $\{0 = x_1, x_2, ..., x_{n-1}, x_n = L\}$, and the convention that $u(x_i, t) = u_i^t$, the scheme can be written

$$\dot{u}_i^t = \frac{\Big(D(u_{i+1/2}^t)\left(u_{i+1}^t - u_i^t\right)\Big) - \Big(D(u_{i-1/2}^t)\left(u_i^t - u_{i-1}^t\right)\Big)}{(\Delta x)^2},$$

although here it was implemented for use on a possibly non uniform grid, and was written

$$\dot{u}_i^t = \frac{\Big(D(u_{i+1/2}^t)\frac{u_{i+1}^t - u_i^t}{\Delta x_i}\Big) - \Big(D(u_{i-1/2}^t)\frac{u_i^t - u_{i-1}^t}{\Delta x_{i-1}}\Big)}{\Delta x_{i-1/2}}.$$

Non-uniform grids were occasionally used in an attempt to more accurately represent dynamics near the boundaries. Several numerical tests indicated that that modest sized uniform grids would provide sufficient accuracy in a reasonable compute time.

The scheme above produces $\dot{u}_i^t$ values on the interior nodes designated by i from 2 to $k-1$. The boundary conditions are applied via ghost nodes, in which u_1^t is set equal to $f(t)$ to enforce the Dirichlet condition, and u_k^t is set equal to u_{k-1}^t to impose the homogeneous Neumann condition. These assignments are made prior to the evaluation of $\dot{u}_i^t$ on each time level. Linear interpolation was used to calculate the required values on half nodes. This discretization, which transformed the PDE in a system of ODEs, was then passed to a Matlab ODE initial valued problem (IVP) integrator. This FD / IVP method had several benefits - although the most significant might be the ease with which this methods was implemented using Matlab. The Matlab suite of ODE solvers include a wide selection of well tested integrators, as well as the ability to control may aspects of the integration.

We note that the values returned from the time integrator represent only those values on interior nodes, and therefore need to be augmented via the boundary rules after computation to generate boundary observations.

In addition to the initial values solvers, Matlab is also able to solve boundary value problems (BVPs). The use of the `bvp4c` would eliminate the need for a fixed space discretization, and capture behavior near the boundary well. However, a toy implementation proved to increase computation time excessively, and the BVP / IVP method was therefore not applied to the full problem. Finite Element methods (FEM) could also have been implemented, but the additional flexibility offered by these methods didn't appear to justify their use.

3.3 Recovery algorithm

The integral identities (3.2,3.3) allow us to explore an approximation of the input to output map. A restriction of $f(t)$, which is a controlled quantity in the direct experiment, to a monotone function allows us to apply a weak version of a maximum-minimum principle. Assuming for the sake of discussion that f is monotone increasing, then

$$f(0) = u_0 \leq u(x,t) \leq f(T) = u(0,T)$$

for all (x,t) in the time space domain $U_T = (0,1)\times(0,T)$. We now recognize a chain of implication. By lemma 4.1.1, the time discretization $\{0 = t_0 < t_1 < \ldots < t_k\}$ leads to corresponding discretization of the range of f, $\{f_0 = f(t_0) < f(t_1) < \ldots < f(t_k) = f_k\}$. The range of f is also the domain of $D(u)$, by the maximum principle. Therefore we can parameterize a piecewise linear approximation of the coefficient $D(u)$ by

$$D(u) \approx \tilde{D}(u) = \sum_{k=0}^{M} \delta_k \lambda_k,$$

where the λ_k's are defined to be

$$\lambda_k(u) = \begin{cases} 1, & \text{if } u > \mu_k \\ \dfrac{u-\mu_{k-1}}{\mu_k - \mu_{k-1}} & \text{if } \mu_{k-1} \le u \le \mu_k \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

With is in mind, we now state our goal: *We seek the set of values $\{d_k\}$ for $k = 1..N$ so that $\tilde{D}$ approximates the true coefficient D.* We note that $\tilde{D}$ is a polygonal approximation to D and is therefore in $Lip(\mathbb{R})$. It should then be possible to recover $\tilde{D}$ from suitable experimental data. In numerical implementation, we make the further approximation

$$\tilde{D}(u) \approx \hat{D}(u) = \sum_{k=0}^{N} d_k \lambda_k,$$

an approximation of $\tilde{D}(u)$ over some $\{f_j\}_N$ nodes, for which the previous statements still hold.

The Algorithm

To begin, we apply the g-integral identity (2.7) on $Q_1 = [0,1] \times [0,T_1]$. Since the solution of the direct problem satisfies lemma 2.1.1(a), we have $\mu_0 \le u_1(x,t) \le \mu_1$ for $(x,t) \in Q_1$. Then only the known nodal value d_0 and the unknown nodal value d_1 are active on this strip. We are going to compute the unknown nodal values iteratively and we denote the $i-th$ iteration for d_k by $d_k^{(i)}$. We set $d_1^{(0)} = d_0$.

We apply the integral identity (2.7) on Q_1 with,

$$D_1 = P_1\left[d_0, d_1^{(1)}\right] \qquad \text{and} \qquad D_2 = P_1\left[d_0, d_1^{(0)}\right]$$

$$u_2(x,t) = u(x,t;D_2,At) \qquad \textit{and} \qquad g_2(t) = \Phi\left[f, D_2\right]$$

$$\hat{\phi}(x,t) = \phi(x,t;D_2,A(T_1-t))$$

We compute $\qquad A_{11} = \int_0^{T_1} \int_0^{x_0(t)} \lambda_1(u_2)\,\partial_x u_2\,\partial_x \hat{\phi}\,dx\,dt,$

$$b_1 = \int_0^{T_1} (g(t) - g_2(t))A(T_1 - t)\, dt$$

and solve

$$A_{11}(d_1^{(1)} - d_0) = b_1.$$

Note that A_{11} and b_1 are computed from $u_2,\ \hat{\phi},\ g_2$ which are all based on the known coefficient D_2.

To continue, we apply the g-integral identity (2.7) first on Q_1, where only d_0, d_1 are active, and then we apply the g-integral identity (2.7) again, but now on Q_2 where d_0, d_1, d_2 are active. That is,

$$\begin{array}{lll} \text{on } Q_1 & D_1 = P_1\left[d_0, d_1^{(2)}\right] & d_1^{(2)} \text{ is unknown,} \\ \text{and} & D_2 = P_1\left[d_0, d_1^{(1)}\right] & d_1^{(1)} \text{ is known,} \end{array}$$

and we compute $\quad A_{11}$ and b_1 as before.

Note that $u_2,\ \hat{\phi},\ g_2$ are based on the updated coefficient D_2 so that, in general, $d_1^{(2)}$ will not be the same as $d_1^{(1)}$.

$$\begin{array}{lll} \text{On } Q_2 & D_1 = P_2\left[d_0, d_1^{(2)}, d_2^{(1)}\right] & \\ \text{and} & D_2 = P_2\left[d_0, d_1^{(1)}, d_2^{(0)}\right] & \textit{note}:\ d_2^{(0)} = d_1^{(1)} \end{array}$$

we compute

$$\begin{aligned} A_{2,1} &= \int\int_{Q_{21}} \lambda_1(u_2)\partial_x u_2 \partial_x \hat{\phi}\ dx\ dt \\ Q_{21} &= \{\mu_0 \le u_2(x,t) \le \mu_1, 0 \le t \le T_2\} \\ A_{2,1} &= \int_{T_1}^{T_2} \int_0^{x_1(t)} \lambda_2(u_2)\partial_x u_2 \partial_x \hat{\phi}\ dx\ dt \\ b_2 &= \int_0^{T_2} (g(t) - g_2(t))A(T_2 - t)\ dt \end{aligned}$$

and we solve

$$\begin{bmatrix} A_{11} & 0 \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} d_1^{(2)} - d_1^{(1)} \\ d_2^{(1)} - d_2^{(0)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

We proceed in this way, where at the k-th stage we apply the integral identity k times, once on each of the strips Q_1 to Q_k. Of course this produces k equations, one for each strip. On each strip, Q_j there are only j unknown active node values $d_1^{(p)}, ..., d_j^{(q)}$, at various stages of iteration, hence the j-th equation contains only the first j unknowns. This leads to a k by k lower triangular system for the differences $d_j^{(p)} - d_j^{(p-1)}$. At the k-th stage of the algorithm we are solving for the first iterate for d_k, for the second iterate of d_{k-1}, etc. This algorithm, which we will call the iterative algorithm, differs from the non-iterative algorithm described in the preceding section. The non-iterative algorithm amounts to suppressing the iterative feature so that for each k, the nodal value d_k is obtained by solving just a single equation, $A_{kk}(d_k - d_{k-1}) = b_k$.

3.4 Numerical Code

The numerical code consists of two main parts. The first, the experiment algorithm, allowed numerical simulation the solution in the entire domain, and the generation of boundary data $D(u(0,t)\partial_x u(0,t) = g(t)$ and $u(L,t) = h(t)$. The ability to quickly simulate a wide range of experiments was extremely useful. The second, the recovery algorithm, was implementation based on the integral identity method. This algorithm utilized the boundary data observations in company with the initial and boundary conditions used to generate this data. The recovery code sought to identify the unknown diffusion coefficient when supplied with experimentally observed data.

All numerical methods were coded for the Matlab 6.1 environment. Preliminary work was performed under Matlab v5.3, although the expanded functionality of the v6.1 was quickly realized to provide more flexibility in implementation. In particular, the improvements made in the ODE suite package of v6.1 proved useful. The call to the ode solver was simplified, and the solution could be returned as a Matlab structure, which allowed the solution to be evaluated post computation on any time level in an appropriate interval. This greatly simplified experiments requiring time scale refinement.

The PDE was discretized using Finite Difference (FD) methods in space to produce a system of ODEs. Non-uniform grids were occasionally used in an attempt to more accurately represent dynamics near the boundaries. Several numerical tests indicated that that modest sized uniform grids would provide the sufficient accuracy in a reasonable compute time. Finite Element methods (FEM) could also have been implemented, but the additional flexibility, such as grid refinement, offered by these methods didn't appear to justify their use. The values of the unknown variable were recorded on whole knots on the spatial grid.

3.4.1 Direct algorithm implementation

In this section we discuss the code used to generate a numerical solution to the direct problem. Subsequent work required that the code be fairly efficient and flexible. The nonlinear term of the equation was managed at each time level evaluation as a linear interpolation of a passed call-out table. The resulting system of ODEs were submitted to Matlab's time integration methods, as chosen by the user. We now present a template for the code used to generate the so called direct solution.

3.4.2 Pseudo Code for the Direct Problem

A pseudo-code to generate the direct solution is as follows:

```
function dudt = udot(t,u,D,F);

% function dudt = udot(t,u,D,F);
% computes
% dudt = ddx(D(u) dudx)
% Boundary conditions are hard coded
% with u(0,t) = f(t) and dudx(1,t) = 0

global XXX

dx = diff(x);

% set bc's
u = [NaN ; u ; NaN];
u(1) = feval(F,t);        % make u(0,t) = f(t)
u(end) = u(end-1);        % make dudx(end,t) = 0

% find D on nodes
Du =  evaluate(D,u);

%compute udot
dudt = diff(Du .* diff(u)./ dx) ./ dx;
```

Matlab ODE solvers require a 'dot' function which takes as input a scaler t and vector valued (column) u, and returns the time derivative as a column vector. For this problem, we require the additional information F and D, both of which are names of user defined Matlab function files. Dirichlet boundary conditions are enforced on the first node and homogeneous Neumann on the last node. This is done by injecting artificial 'ghost' nodes. The Dirichlet condition is easily enforced, and the no flux condition is implemented by the replication of the original last value of u, which makes the right sided difference naturally zero. Notice that the time derivative that is returned is only given on the interior mesh nodes. The boundary values are not explicitly computed in this dot file, but instead are assigned at a later point in time.

This code was used in several ways in the resulting experiments. It was used to generate numerical experimental data, and was also called in by the recovery scheme. In an attempt to isolate the recovery process from the generation of the numerical data, the direct solvers, time and space grid were chosen independently.

The actual Matlab code used was ‘vectorized’- i.e. written so as to accept matrix valued input for u. This resulted in a large computational improvement. The complete code may be found in the appendix.

3.4.3 Experiment Algorithm implementation

In order to generate sufficient data, a numerical algorithm was written to construct a numerical solution, and use this numerical solution to create simulated boundary data. A pseudo code for this algorithm is as follows:

```
function [t,g,h,u,f] = experiment(Solver,Udot,tspan,u0,F,D);

global XXX

u0_0 = evaluate(F,min(tspan));
u0 = [u0_0 , 0*XXX(1:end-1)]';

[t,u] =  feval(Solver,Udot,tspan,u0,[],D,F);

f = evaluate(F,t);

[t,g,h,f] = get_bcs(t,u,f,D);

%-------------------------
```

```
function [t,g,h,f] = get_bcs(t,u,F,D);

global XXX

% this routine gets boundary data
% g = -D(u( 0.5 ,t)) * dudx( 0.5 ,t)
% h = u(1,t)

u = [NaN;u;NaN];
u(:,1) = feval(F,t) + 0*u(:,2);   % Fix Dirichlet BCs
u(:,end+1) = u(:,end);            % Fix Neumann BCs

dx = diff(XXX(1:2));
Du = evaluate(D,u(:,1));
dudx = ( u(:,2)-u(:,1) ) / dx ;

f = F;
g = -1 * Du .* dudx;
h = u(:,end);
```

The function `experiment` requires several inputs. A string specifying which solver to use, the name of the file which computes the time derivative, a time span for integration, and names of the coefficient function D and the nonhomogenous Dirichlet condition function F are necessary. This code first generates the initial profile. Then a call to a user specified ODE suite solver is made with the `feval` command, taking as inputs the user defined coefficients D and boundary condition F, which are external Matlab functions. The resulting time solution, here produced on a time grid provided by the user, is passed to the local sub-function `get_bcs`. This sub-function first fixes the boundary values and then returns the appropriate boundary measurements. Notice that the 'missing' boundary information is reconstructed in the `get_bcs` code and is not computed explicitly by Matlab's time integration.

3.4.4 Recovery pseudo code

In this section, we present pseudo code that utilizes the $g(t)$ data and the corresponding integral identity, 3.2. Code that incorporates $g(t)$ and/or $h(t)$ data is easily constructed with only slight modification.

```
function D = Coeff_Inverse();

% Try to recover the coefficient D(u)
% in the model u_t = (D(u)u_x)_x
% given experimental output

[T,F,G,D0] = load Data.file;

%  T = time               F = forcing f(t)
%  G = -D(u)u_x @   x=0 ; ie flux  at x=0
% D0 = D(u(0,0))        ; initial coefficient

% Begin Recovery

D = D0;       % Initial approximation
level = 1;    % Initial time level

while (max(t) < Tmax)

   % Initialize and set problem specs
   % includes problem, ic's, bc's, methods
   forward.info = ...

   for strip = 1:level
      [t,f,u,u_x,g] = solve_forward(forward.info,D);
   end

   deltag = G-g;
   theta = ???;      % assign theta (dual data)
   DualD = ???;      % assign approx dual D operator

   % Set problem specs
   % includes problem, ic's, bc's, methods
   dual.info = ...
   for strip = 1:level
      [phi_x] = solve_dual(dual.info,DualD);
   end

   for strip = 1:level
      for region = 1:strip
         % compute 2D region and basis element
         [Omega,lambda] = Active_region(u,f);
         % compute the integral over Omega
         A(strip,region) = Int(u_x.*phi_x*lambda, Omega);
      end
      b(strip) = Int(deltag.*theta, t );
   end

   deltaD = A \ b;
   D = D + deltaD;

   % Adaptive control possible here
   if (some condition is met)
      level = level + 1;
   end

end;
```

3.5 Adaptive control

In this section, we discuss several implementations that allowed the nodal mesh to computed adaptively. Several adaptive schemes were implemented, some more successfully than others. In the following, we must keep in mind the integral identities upon which the scheme is based. We compute the update to the coefficient by calculating

$$\Delta D = \frac{\int_0^\tau \Delta g G^*(t) + \Delta h H^*(t) \;\; dt}{\int_0^\tau \int_0^L \lambda(u) \partial_x u(x,t) \partial_x \phi(x,t) \; dx \; dt}$$

The value of the double integral is essentially the derivative of the map from the input pair $(g(t), h(t))$ to the output D. This term involves both $\partial_x u(x,t)$, the space derivative of the forward solution, and $\partial_x \phi$, the derivative of an associated dual problem. We have control of several quantities in this expression,

- The forcing $f(t)$ in the direct problem,
- The data $G^*(t)$ in the g dual problem,
- The data $H^*(t)$ in the h dual problem.

We first considered enforcing a nodal basis *a priori,* which in turn allowed computation of the corresponding nodal times. This had the benefit that the forward solutions could be generated on this known time interval. We also implemented a method where time breaks were imposed which allowed the user to decide whether to apply the current update ΔD to the recovered coefficient, or allow the algorithm to proceed unchecked until the next time break. From experience gained with this method, an automatic adaptive method was designed. The first observation was that the region over which the integration take places grows as a function of time. The function $\lambda(u)\partial_x u(x,t)\partial_x \phi(x,t)$ needed to be computed in this region. Recall that the magnitude of $u(x,t)$ quickly diminishes from its value of $f(t)$ as a function of x, and that this decay is Gaussian and dependent on the unknown coefficient. Evidently, the magnitude of $\partial_x u(x,t)$ is large for small x, and decreases rapidly toward zero as x increases. The rate of this decrease depends on D. The adaptive scheme requires monitoring the solution at a specific internal node, and halting time integration when the value at this node becomes larger than $u(0,t_i) = f(t_i)$, where t_i is the initial time of interest. In this way, the size of the region of integration can be controlled. The magnitude of $\partial_x u(x,t)$, although dependent on the value of the unknown coefficient, might be controlled through control of $f(t)$, boundary forcing. The influence of this type of control remains to be studied with this algorithm, although it has been considered in [20].

3.6 Experiments utilizing only the direct algorithm

Implementation of the direct problem in the Matlab environment allowed many preliminary numerical experiments. Several of these initial experiments are presented here. A variety of diffusion coefficients were used to simulate the boundary data $g(t)$ and $h(t)$. The diffusion coefficient assumed many functional forms, although here we present those based on a family of Sine functions, a family of Arctan functions, and a family of piecewise linear functions. Plots such as figure 3.1 are representative of this series of experiments exploring the influence of coefficient variation on the measurable output data. Although the domain of figure 3.1 is larger than the $[0, 1]$ domain of the later plots, all other aspects are similar. The larger domain serves to reinforce the impression that there is some validity in assuming that boundary data might allow recovery of coefficient information. All experiments presented in the following subsections were produced via the same method. A time scale was first defined to be $[0, 1]$ with boundary forcing $f(t)$ fixed at $f(t) = t$. The space discretization was accomplished with the standard finite difference scheme. This reduces the problem to a system of ODEs in time. A functional form of the coefficient was coded, and both the system of ODEs and this coefficient were supplied to a Matlab time integrator. The resulting numerical solution was used to compute the simulation data $g(t)$ and $h(t)$.

Note that the range of $f(t) = t$ corresponds in this case to the domain of the supplied coefficient. Therefore, coefficients will be have as their

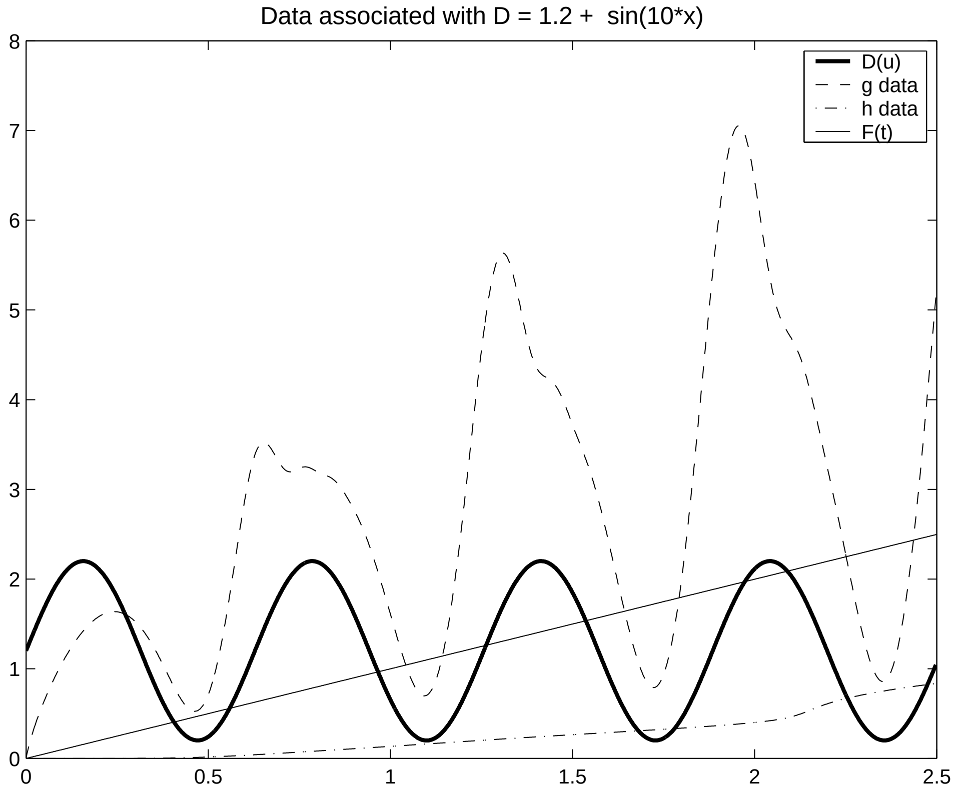


Figure 3.1: Coefficient and corresponding boundary data

domain $[fmin = 0, fmax = 1]$ and all time data will have domain $[tmin = 0, tmax = 1]$, and are therefore equivalent in scale. Plots of coefficients should be understood to be plotted against u, while the plots of boundary data are plotted against t. This convention will allow us to easily compare the coefficient and the corresponding data.

3.6.1 Coefficient taken from a Sine family

Several numerical experiments were conducted that utilized coefficients taken from a family of Sine functions of the form

$$D(u) = \alpha + \beta \sin(\omega t), \tag{3.5}$$

where α, β and ω are parameters. We first describe a series of experiments that fix α and β and allow ω to take integer values from 1 to 20. Plots of the coefficient and corresponding data, such as 3.1, suggest strongly that fluctuation in coefficient produces a noticeable signature in the boundary data. This signature was most notable in the boundary flux data, $g(t)$ and to a lesser extent in the state data $h(t)$. To test the hypothesis that the $g(t)$ data might contain information relating to perturbation in the unknown coefficient, the effect of the mean coefficient was first removed. In the constant coefficient case, the expected response curve of flux data $g(t)$ is proportional to $\sqrt{t}$. The constant coefficient effect was filtered by subtracting from $g(t)$ the signal $\sqrt{t}$. The modified data was then submitted to a simple correlation coefficient analysis. The covariance of the filtered data and the a sequence of sine functions with frequency from 1 to 20 was computed, and the results presented in the waterfall plot 3.2. The x-axis index represents the frequency of the sine function, while the y-axis index refers to $g(t)$ data produced with a diffusion coefficient of the indexed frequency. The correlation matrix was 2×2, and the minimum plotted as the magnitude in the z direction. Notice the high covariance measure when the frequency of the sine function is coincident with the frequency of the diffusion coefficient used to generate the data. Similar experiments were conducted to test the h data. While these produced very small covariance measures, this could easily be attributed to a failure to correctly filter the artifact of the constant portion of the coefficient.

While the h data doesn't appear to be related to the frequency of the forcing, it does seem to contain information about the coefficient's mean contribution. A series of experiments were conducted which allowed the

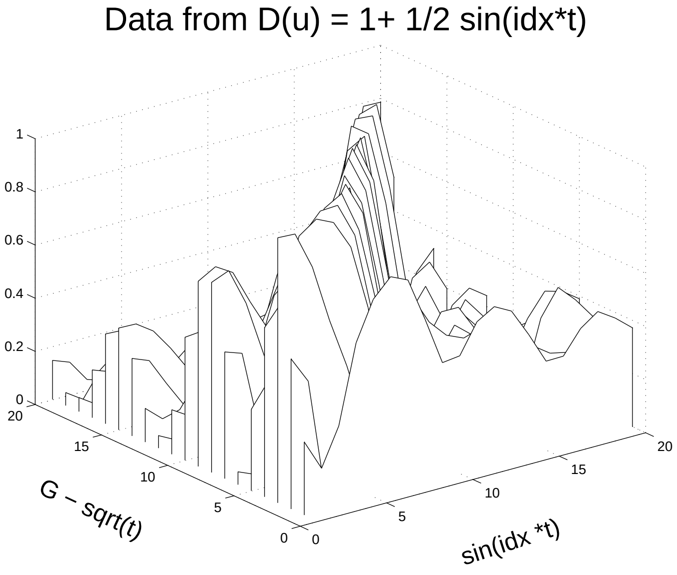


Figure 3.2: Correlation between input D and output g

amplitude (β) of the coefficient from 3.5 to vary. Heuristicly, it appears that breakthrough time is noticeably influenced by the mean of the diffusion coefficient. We have defined breakthrough time, rather arbitrarily, to be time for which the recorded $h(t)$ data first became larger than $1e-3$. In table (3.1), the breakthrough times are presented for several coefficients with the same mean.

Coefficient D(u)	Breakthrough Time
1+1/40 sin(2 π u)	0.0990
1+6/40 sin(2 π u)	0.0985
1+10/40 sin(2 π u)	0.0985
1+15/40 sin(2 π u)	0.0980
1+ 1/2 sin(1 u)	0.0990
1+ 1/2 sin(6 u)	0.0975
1+ 1/2 sin(10 u)	0.0965
1+ 1/2 sin(15 u)	0.0950

Table 3.1: Breakthrough times for various coefficients

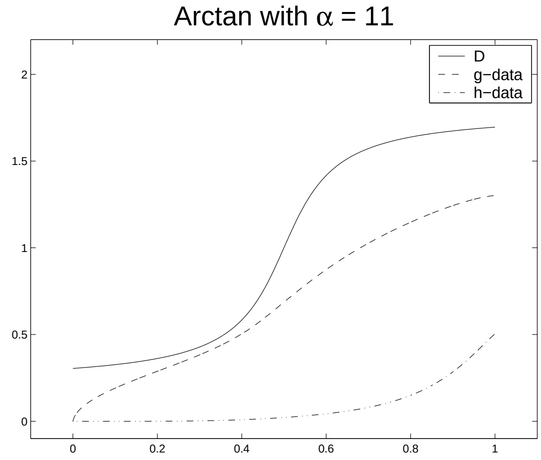


Figure 3.3: Data from Arctan with $\alpha = 11$

3.6.2 Coefficients taken from an Arctan family

In this subsection, we present experiments in which the coefficient is taken from the arctan family

$$D(u) = 1 + \frac{1}{2} \arctan\left(\alpha\left(u - \frac{1}{2}\right)\right). \tag{3.6}$$

The parameter α controls the derivative of this function. Notice that the following plots record that the sudden change in coefficient is reflected immediately in the g data, while the effect is increasing delayed in the h data as the coefficient curve become steeper. Also, note that the effect of the coefficient is more subtle in the h data.

In figure 3.3, breakthrough time of the h data is evident at $t \approx 0.5$, while the influence of the rapid change in the coefficient on the g data is nearly instantaneous. In both of the figures 3.4 and 3.5, the breakthrough time appear nearly identical, at t approximately 0.6. The breakthrough time might be considered the time it takes the boundary information from the left to propagate to the right boundary measurement. Since the same linear in time boundary conditions were used in all the three experiments,

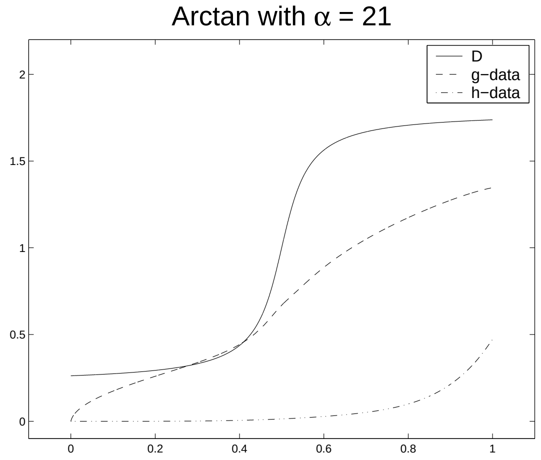


Figure 3.4: Data from Arctan with $\alpha = 21$

the boundary condition increase in time effects the diffusion coefficient, which here increases the speed of the diffusion, and is evident in response of the h data. After the breakthrough is achieved, the increase in the h curve generated with $\alpha = 31$ is less steep than that generated by taking $\alpha = 41$. Before the breakthrough time is reached, both coefficients are effectively equal. After breakthrough is achieved, however, the $\alpha = 41$ case has a higher effective rate of diffusion than does the $\alpha = 31$ experiment. This translates to the more rapid response in the h data curve for $\alpha = 41$ than in the data generated using $\alpha = 31$, which in turn is steeper than that of the $\alpha = 11$ experiment.

Also, notice that g data appears slightly perturbed in the time/u region corresponding to the most rapid increase in D. It appears that the g data reacts to the jump in the coefficient while the h data does not.

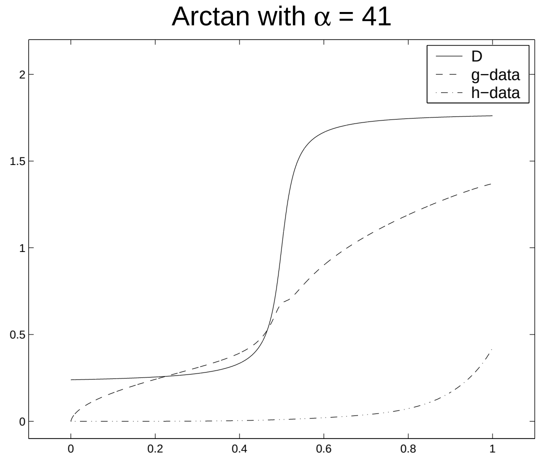


Figure 3.5: Data from Arctan with $\alpha = 41$

3.6.3 Coefficients taken from a Piecewise Linear family

The coefficients used in the following experiments were taken from the three parameter piecewise linear family

$$D(u) = \begin{cases} c + u\beta & u \leq \alpha \\ c + \alpha\beta & u > \alpha. \end{cases} \tag{3.7}$$

With α and β chosen so as keep $D(u)$ positive. Two such series are presented here. The first set fixes $c = 1.5$ and $\beta = -1$ while allowing α to vary from 0 to 1. These plots clearly indicate the breakthrough times in the h data. They also reflect a slight reaction in the g data in the places where there is a discontinuity in $D'(u)$.

Notice that the h data curves appear nearly identical in figures 3.8 and 3.9, an indication that the h data might contain little coefficient information after time approximately 0.75. It should also be noted the g data reacts strongly to the coefficient in the sense that the g data curves in 3.6 through 3.13 at precisely the same point where the coefficient curves differ from one another.

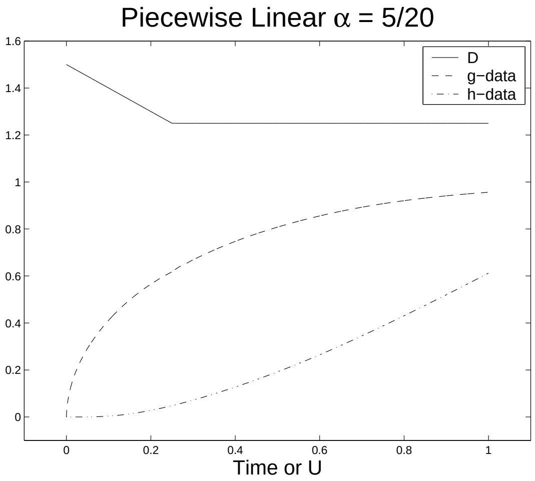


Figure 3.6: Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 0.25$

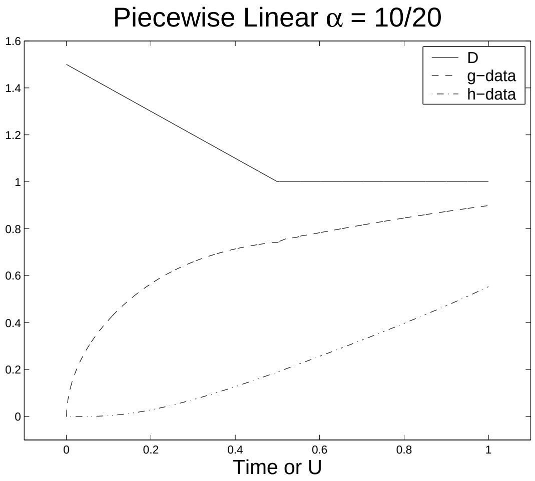


Figure 3.7: Data from piecewise linear family with $c = 1.5, \beta = -1$ and $\alpha = 0.5$

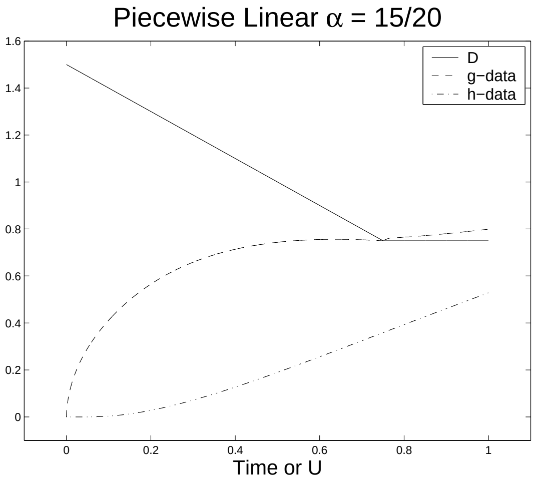


Figure 3.8: Data from piecewise linear family with $c = 1.5, \beta = -1$ and α $= 0.75$

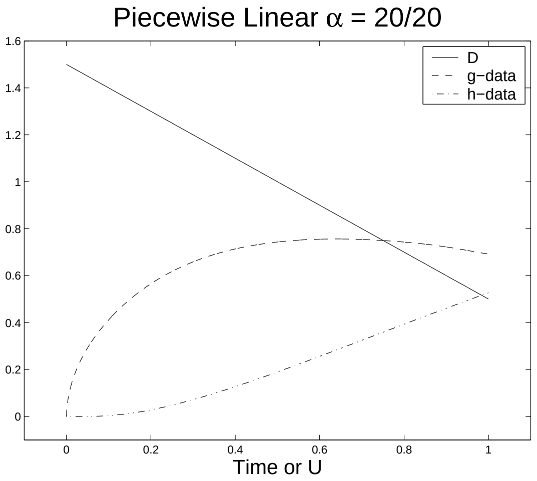


Figure 3.9: Data from piecewise linear family with $c = 1.5, \beta = -1$ and α $= 1$

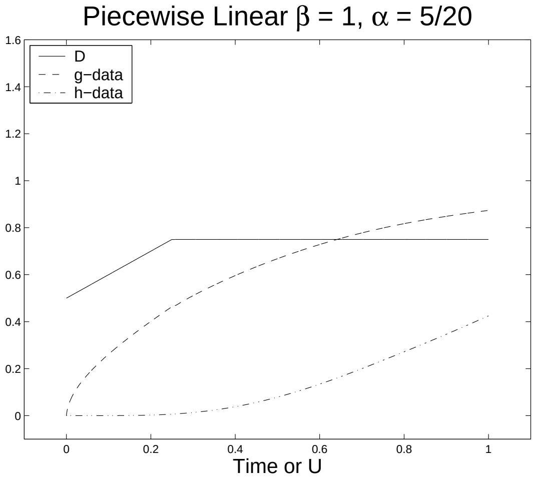


Figure 3.10: Data from piecewise linear family with $c = 0.5, \beta = 1$ and α $= 0.25$

Similarly, several plots from experiments using parameters $c = 0.5, \beta = 1$ and α from 0 to 1 also indicate that there is little coefficient information in the h data for t larger than 0.5. The diffusion coefficient in this series is smaller than that of the previous series, which causes the h data to respond more slowly to coefficent modifications.

Figure 3.14 plots the 2 norm of the consecutive terms in the $h(t_k)$ time series, generated over range of α values. The plot indicates that the coefficient whose initial value is $c = 1.5$ allows the h data to converge more rapidly in the 2 norm.

Although these experiments were extremely simple, they did help develop intuition that proved useful in the study of the inverse problem. These results provided a direction and motivation for much of the later work. They provided a excellent test for the numerical solution of the direct problem under a wide range of coefficients. These experiments also provided a benchmark for the data behavior as a function of diffusion coefficient.

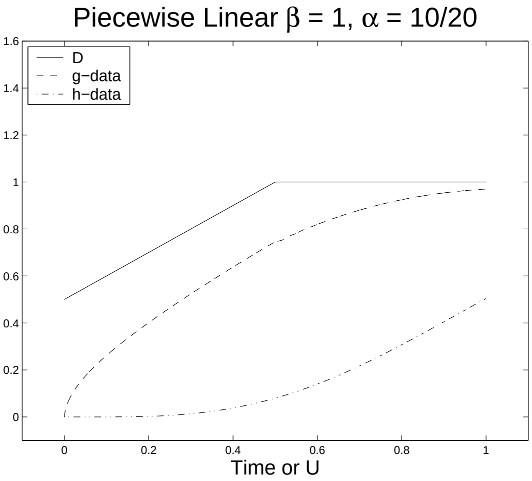


Figure 3.11: Data from piecewise linear family with $c = 0.5, \beta = 1$ and α $= 0.5$

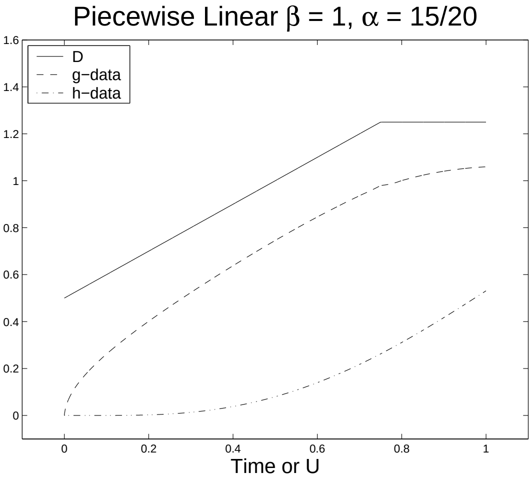


Figure 3.12: Data from piecewise linear family with $c = 0.5, \beta = 1$ and α $= 0.75$

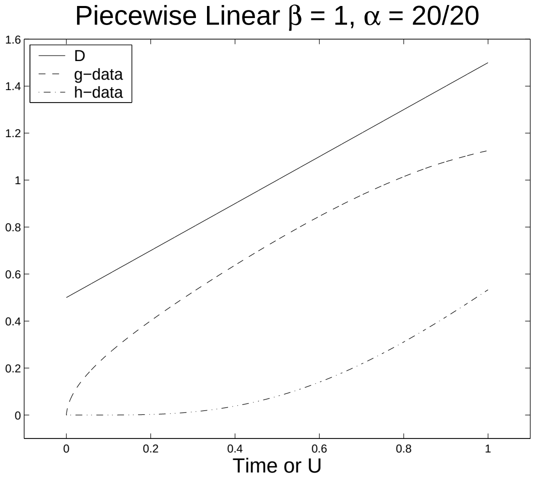


Figure 3.13: Data from piecewise linear family with $c = 0.5, \beta = 1$ and α $= 1$

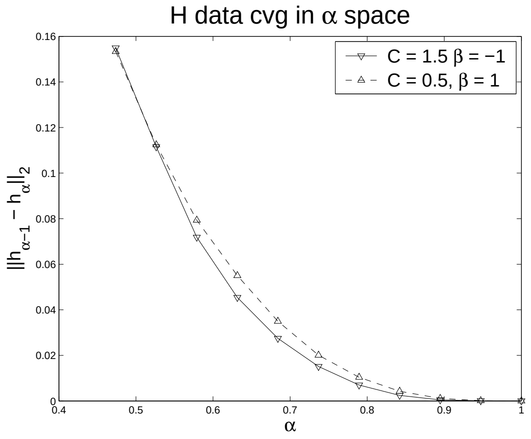


Figure 3.14: Convergence comparison of $h(t)$ data

3.7 Experiments Utilizing Full Recovery Algorithm

In this section, we discuss several experiments which used the full recovery algorithm. In all cases, the data used in recovery was generated numerically. This allowed a direct comparison of the 'true' coefficient with the approximation generated by the algorithm. The following areas were explored:

- Use of g, h and weighted (g, h) pair
- Dimension of uniform nodal basis
- Non uniform nodal basis
- Perturbed nodal basis
- Dependence of solution on time
- Iteration
- Diagonal depth in linear system

3.7.1 Data selection and weighting

In its original form, the integral identity contains both the g and the h data. The high correlation between the g data and the coefficient D made clear the possibility that the g might contain more qualitative information than the h data. A series of experiments in which a weighted average of the two data types was conducted to try to quantify this.

3.7.2 Dual data selection

There are many choices for dual data in the problem. Choosing $G^*(t) = \Delta g(t)$ and $H^*(t) = \Delta h(t)$ makes the numerator into a L^2 norm. A choice of $G^*(t) = \gamma_1 sgn(\Delta g)$ and $H^*(t) = \gamma_2 sgn(\Delta h)$ makes this a weighted L^1 norm. There are many possibilities, and several numerical experiments were performed to gain some insight into what this choice might imply.

In an attempt to isolate the effect data used in the dual problem, the linear function

$$D(u) = 1 + \frac{1}{4}u$$

was used to generate the initial data. This function was chosen to reduce the error induced by approximating the dual problem. We also solved this problem for a single free node.

The following experiments imply that the choice of dual data is important, and that the choice of $G^*(t) = \Delta g$ and $H^*(t) = \gamma$ apparently allows a more accurate recovery of the tested coefficient. In practice, this was the combination of data that was quite effective when recovering a wide range of coefficients.

With G^* fixed at zero, modifications of the state data H^* were considered. The dual data was constructed as a linear combination of the data types in two parameters. Using this data, the coefficient D was recovered over a range of parameter values. The data had the form:

$$H^*(t) = (1 - \lambda)\Delta h(t) + \lambda\gamma_2$$

where λ is a number in $[0, 1]$ and γ_2 held constant. We first present those cases where γ_2 was fixed at 1, and allow λ to vary.

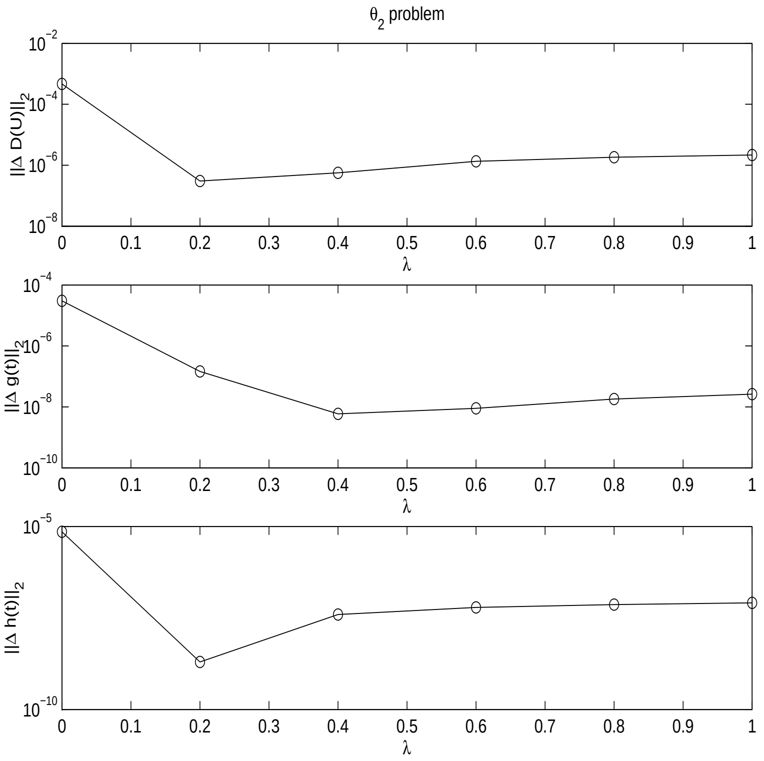


Figure 3.15: Error in λ parameter space for $H^{*}(t)$ data

Figure 3.15 indicates that a choice of dual data which is weighted toward a constant is preferable. There appears to be a minimum in the error measure of coefficient recovery around $\lambda = 0.2$, which corresponding to well defined minima in the error of Δg. Recall that g data is not being used in this experiment, although the plot of the Δg error seems to resemble the plot of error in Δh. Both of the errors in the observable quantities approach $1e - 8$, which is close to the expected error of the numerical integration scheme used. Now we fix λ at one and solve the problem with γ_2 ranging from 10 to 100 by increments of 10. These results are presented in 3.16 in the same format as the preceding plots. This series seems to indicate that the choice of constant γ_2 for values less then 70 appear to have little effect in the error measures. For values of γ_2 larger then 70, however, the errors seem to stabilize. However, they remain at the same average level of the smaller parameter values.

Fixing $H^* = 0$, modifications of the flux data $G^*(t)$ were then considered. The experiments were performed by constructing a linear combination of the data types via two parameters, and then recover and record the approximate coefficient as these parameters were independently traversed. The data had the form:

$$G^*(t) = (1 - \lambda)\Delta g(t) + \lambda\gamma_1$$

where λ is a number in $[0, 1]$ and γ_1 is a constant. We first present those cases where γ_1 was fixed at 1 and λ was allowed to vary.

Figure 3.17 is a logy plot of L^2 error in $D(u)$, Δg and Δh, respectively. This series of plots clearly suggest that setting $G^*(t) = \Delta g(t)$ will lead to a minimum error in $D(u)$, Δg and Δh. This error also appears to increase smoothly as the parameter λ transitions from 0 to 1.

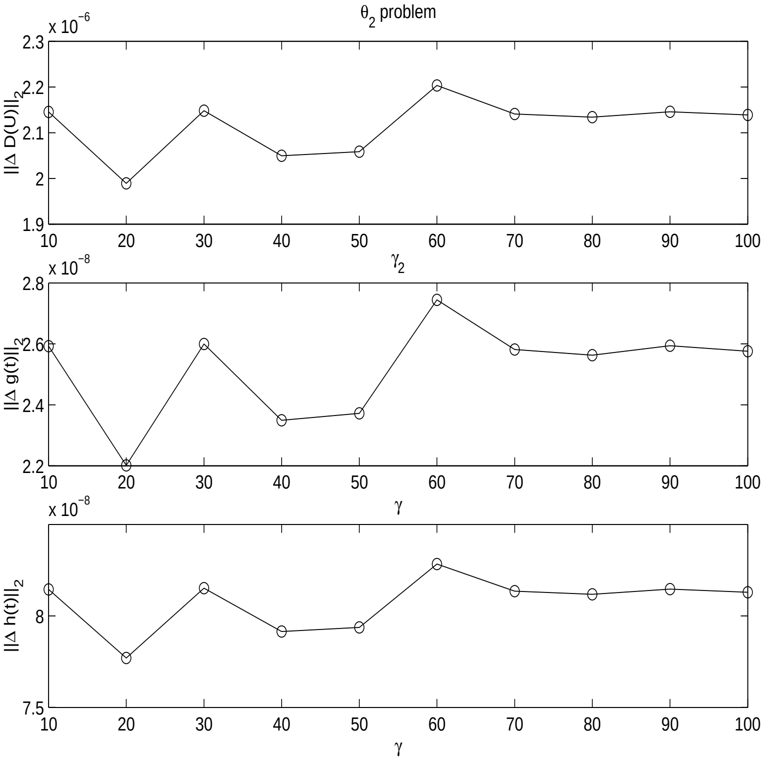


Figure 3.16: Error in γ_2 parameter space

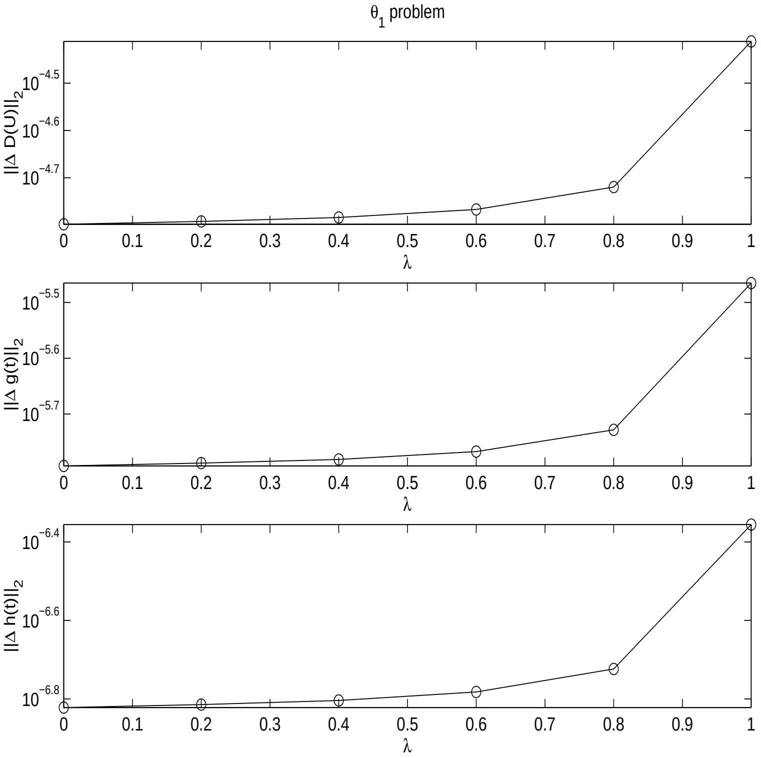


Figure 3.17: Error in λ parameter space for $G^*(t)$ data

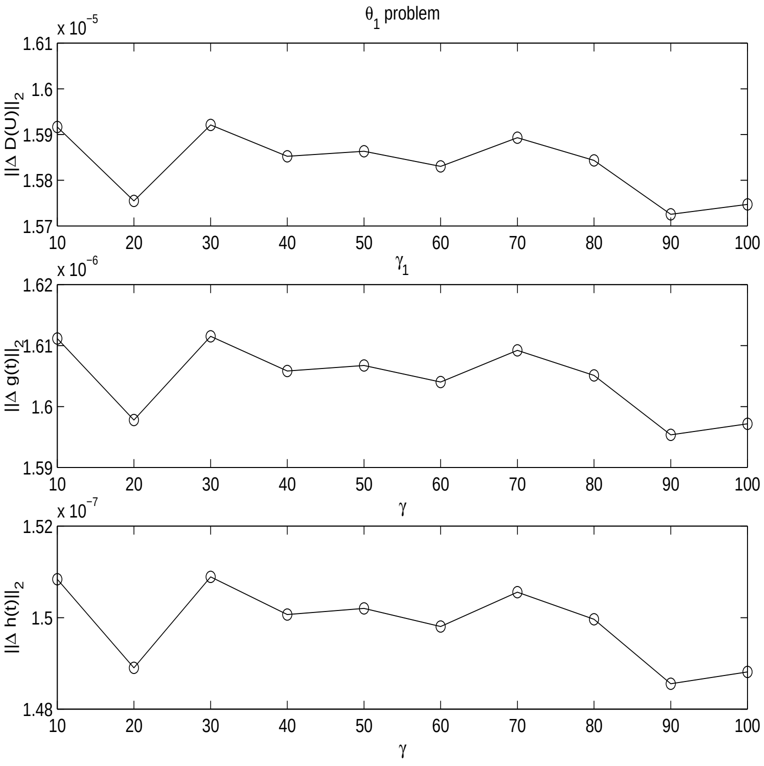


Figure 3.18: Error in γ_1 parameter space

Now fix λ at one and solve the problem with γ_2 ranging from 10 to 100 by increments of 10.

Figure 3.18 is a plot of the L^2 error of $D(u)$, Δg and Δh, respectively. There is a consecutive order of magnitude difference in these plots, although all three appear qualitatively similar. While not definitive, it appears that error tends to decrease at the constant γ is increased, although this effect might be an indication of the order of the approximations used. The measures here are quite small.

3.7.3 Dimension of uniform nodal basis

The numerical recovery algorithm uses a nodal basis to represent the recovered algorithm. There are many possible ways in which to choose this basis, but here we explore the coefficient on a uniform grid. Experiments were performed to locate an optimal uniform grid spacing. The approximation to the coefficient was computed on a grid of n nodes. Here the non iterative algorithm was applied. Since the true coefficient was known, the L^2 error of the recovered coefficient and true coefficient was measured.

Figure 3.19 displays the effect of refining the outer mesh by increasing M, the number of nodes in order to identify the coefficient

$$D(u) = 1 + u, \qquad 0 < u < 1,$$

The non iterative algorithm was applied in this particular experiment. The results for $M = 2, 5$ and 9 are shown in addition to a plot of the $L^2 - error$ versus M. The error cascade is apparent in the plots corresponding to $M = 5$ and 9, as the iteration in each of these individual identifications proceeds. The last subplot summarizes this series, as it shows the error decreasing with increasing M up to about $M = 5$, at which point the error begins again to increase. This result is in qualitative agreement with (2.16). We also notice that in the last figures in 3.19 an overshoot/undershoot feature. Before this point, there appears to be a systematic bias to underestimate the coefficient. This effect will be discussed at a later point.

The series of plots in figure 3.20 represent the approximated coefficient as well as the true coefficient

$$D(u) = 1/2 + 2u - u^2$$

used to generate the observed data. The iterative algorithm was used. In addition, both the (f, g) and (f, h) data pairs were used, which results in the very precise recovery of the unknown coefficient. Notice that the degree

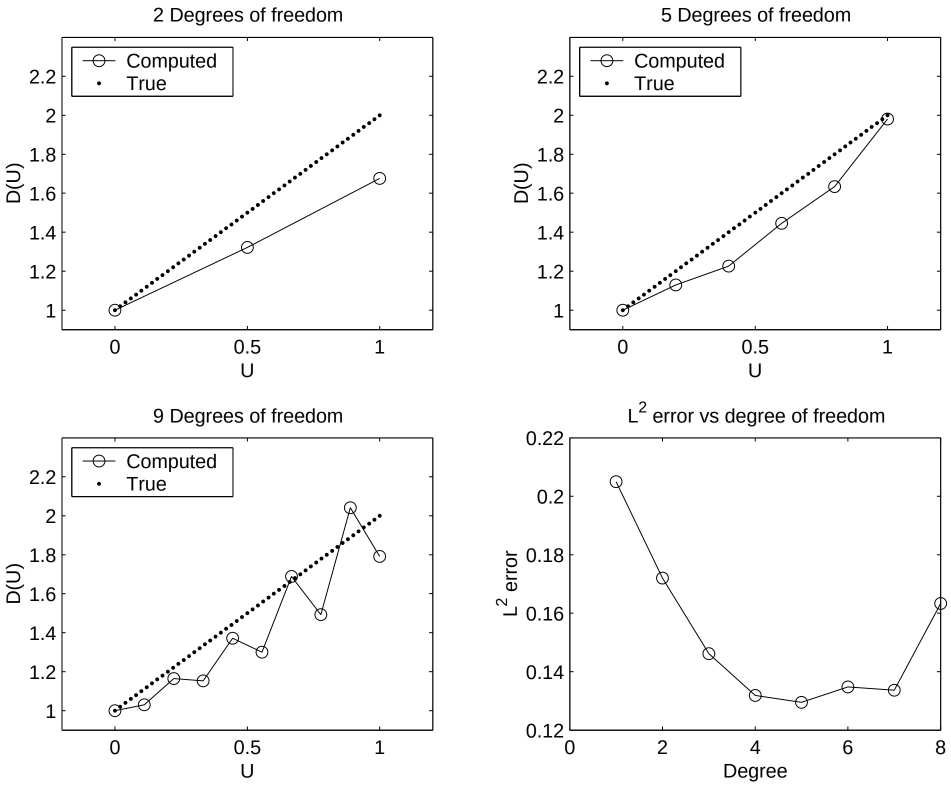


Figure 3.19: Recovery of a linear coefficient - g data only

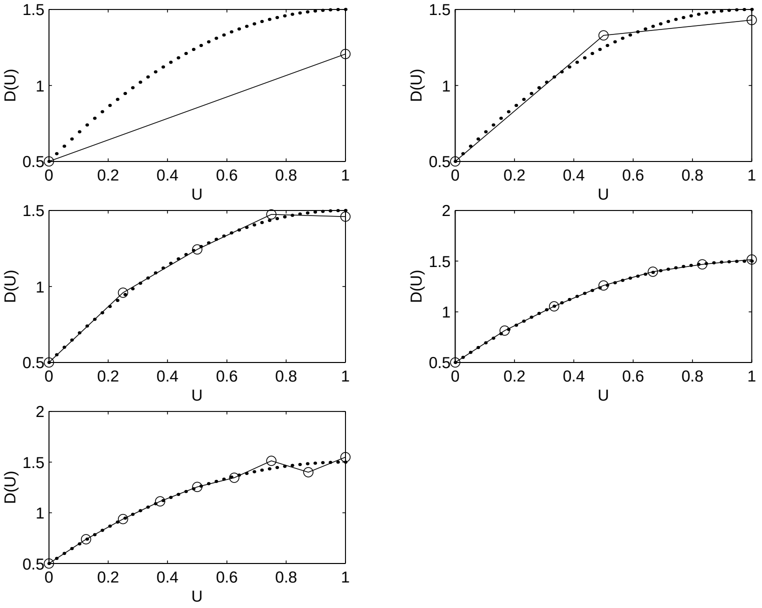


Figure 3.20: Uniform Nodal Basis with g+h data

of the nodal basis still plays a role, as the last plot in figure 3.20 begins to show some oscillatory behavior. This overshoot/undershoot typically manifests when the degree of the approximation space becomes too large.

The nodal values of the approximate coefficient, marked with open circles, appears to closely correspond to the 'true' coefficient. Clearly there is a breakdown in the approximation as the number of nodes is allowed increase. While the more advanced algorithm was able to capture a more difficult coefficient, it still was unable to continue to resolve the coefficient beyond some critical level. In this case, the minimum error occurs around dimension 7, with a slight increase of error on either side. It is interesting to note that in both plots of error, the error rapidly becomes worse as the optimal dimension is exceeded. The might indicate that any adaptive algorithm might make use of the feature, either by constructing a suboptimal grid that stops before the optimal grid, or be able to recognize the overshoot undershoot error and reduce the dimension.

3.8 Local Nodal Refinement

A coefficient basis of the form

$$D(u) = \begin{cases} 1 & u < 1/4 \\ 2(u - 1/2) & 1/4 < u < 3/4 \\ 2 & u > 3/4. \end{cases}$$

was used in this series to generate observation data. This piecewise coefficient allows us to examine the effect of local refinement near a feature. For example, this coefficient might represent the best nodal basis of a differentiable function, such as a function from the Arctan family. The experiment could then be interpreted as testing recovery on structured refinement near a region of rapid coefficient change. For the given coefficient, there are two critical nodes, one at $u = 1/4$ and the other at $u = 3/4$. Here we present only those concerning refinement of the nodal basis after $u = 1/2$. Four runs were performed in which the nodal breaks were prescribed. The following nodal values were used

```
run 1 = [ .25 .50 1.0 ];
run 2 = [ .25 .50 .75 1.0];
run 3 = [ .25 .50 .625 .75 .875 1.0];
run 4 = [ .25 .50 .5625 .625 .6875 .75 .8125 .875 .9375 1.0];
```

The figure 3.21 presents the true coefficient with recovered approximation generated by the algorithm on the nodal basis given above. `Run 2`, represented with the solid line plot, uses the true nodal basis for the optimal recovery if no approximations were necessary, and would correspond to the algorithmic coefficient $\tilde{D}(u)$, which would be the true piecewise linear coefficient. We make several approximations, however. The dashed plot, the coefficient recovered using the nodal basis of assigned to `run 1`, represents a recovery in which the node at $u = 3/4$ is omitted. While this recovery uses

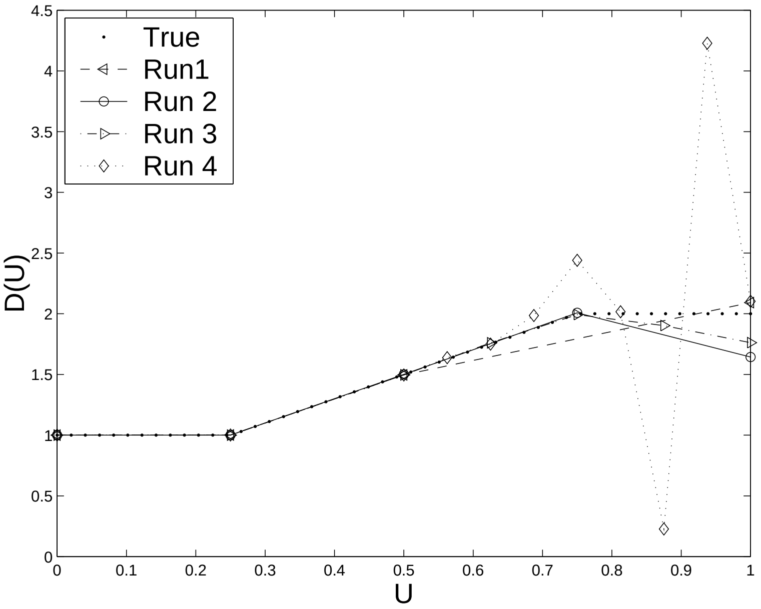


Figure 3.21: Local Grid refinement

fewer points then does the first run, the overall recovery appears, at least visually, to be slightly better then the recovery of `run 2`. `Run 1` captures the rapid change in the coefficient in what appears to be an average. Again, the overshoot undershoot effect is seen in the most highly refined experiment, `run 4`. Also notice the plots in both `run 2` and `run 3` underestimate the true coefficient.

3.9 Minimum Resolution

In the experiments which the dimension of the optimal uniform nodal basis, it became clear that the choice of too large a basis leads to overshoot/undershoot effects. In this section, we discuss a series of experiments designed to explore the minimum resolution of a nodal basis.

Once again we build our observable data with

$$D(u) = 1 + u.$$

over the coefficient range from 0 to 1. We then apply the adaptive algorithm and produce a nodal basis of

```
[ 0.0, 0.1000, 0.2123, 0.3547, 0.5162, 0.6877, 0.8570, 1.0 ].
```

The difference of these nodes were computed, and then multiplied by a scaling factor ranging from 36% to 120%. The resulting vector was cumulatively summed and submitted to the non-adaptive recovery algorithm. The plot 3.22 provides a summary of these computer runs. The x axis in this figure indicates the relative scaling of the experimental basis to the original basis. The experiments were conducted over the full range of scaled basis nodes, resulting in maximum time intervals ranging from 0.36 to 1.20 time units. Those coefficients whose intervals exceeded $[0, 1]$ were linearly interpolated to $[0, 1]$. The square error was computed over the domain, and then normalized based on the measure of the domain. As expected, the linear coefficient was recovered with increasing accuracy as the grid became increasingly coarse. Notice that the error begins to decrease rapidly from its maximum of approximately 1.5 once the mesh scaling moves above 60% of the baseline mesh scale. This could be interpreted to mean that there is insufficient information available for accurate recovery on the fine grid. Once the error contributions in the integral identity are balanced by the significant information, the recovery algorithm begins to perform well. Also of interest is the observation that the baseline nodal basis appears to perform well in this series. The error increases noticeably as the grid becomes refined, while the error for coarser grids appear to be approaching some asymptotic limit. Although the nodal basis is non uniform, grid refinement shows features consistent with refinement of the uniform nodal grid.

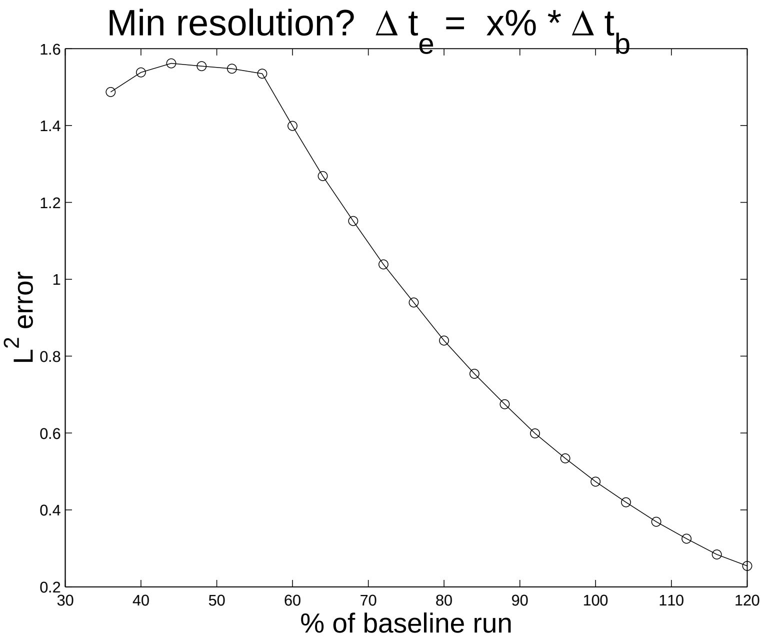


Figure 3.22: Minimum Resolution

The main feature of this experiment is the indication that once a minimum resolution is reached, further refinement of this mesh will not improve the recovery. This series also suggests that the adaptive grid selection appears to work well in this linear coefficient case.

3.10 Width of Data strips

In this section, we consider the numeric structure of the matrix A. Recall that iterative method involves a lower triangular matrix, entries of which are integrals over the active coefficient range. As we uniformly refine the width of the data strips, this matrix increases in dimension. Here a uniform time nodal basis is used over $[0, 1]$, with $\Delta t = 2^{-6}$. This choice generates a 64×64 matrix. We consider the main diagonal, a vector of length 64, and 5 subdiagonals, the shortest of which contains 58 entries. In figure 3.10, we plot these 6 vectors. The horizontal axis indicates the time node index for the matrix entry, while the vertical axis is the magnitude of the entry. Notice that the main diagonal, plotted with the thickest line, appears to be much smoother than the other diagonals. This indicates that the diagonal entry, while not dominate in magnitude, shows little numeric error from onset. The subdiagonal terms initially show a perturbation around what might be considered the actual value. There are many contributions to the error in these calculations, but figure 3.10 suggests that these errors become less influential as the width of the strip increases and the algorithm subsequently becomes more numerically robust. As expected, the index at which the perturbations appear to die occur at later times as we move farther from the main diagonal. For example, the entries in the first subdiagonal become more stable after index 11, while the entries taken from the 3^{rd} subdiagonal require almost twice as many time nodes to stablize. The size of both the active coefficient region and the magnitude of the integrand rapidly decrease as we move more deeply into the interior of the region. The first subdiagonal has the largest magnitude, a feature attributable to the fact that the region of activity is roughly trapezoidal, as opposed to the more triangular region corresponding to main diagonal entry.

3.11 Iteration

In this section, we compare the non iterative methods described in the earlier chapter with the iterative method in this chapter. Here we demonstrate that suppressing the iteration leads to cascading errors in the sequentially computed nodal values d_k as shown in figure 3.24. The coefficient shown in this figure,

$$D(u) = 2 - \arctan\left[6(u - 1/2)\right], \qquad 0 < u < 1, \tag{3.8}$$

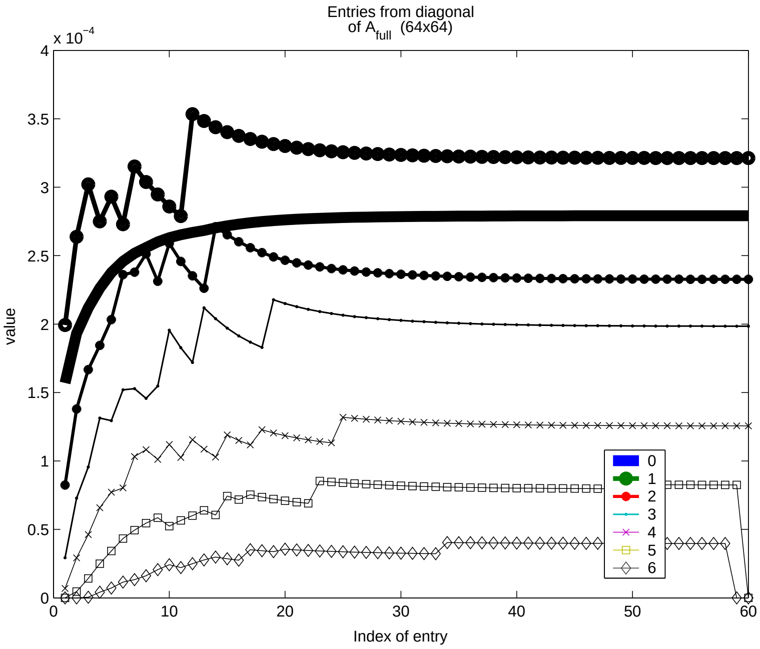


Figure 3.23: Diagonal elements

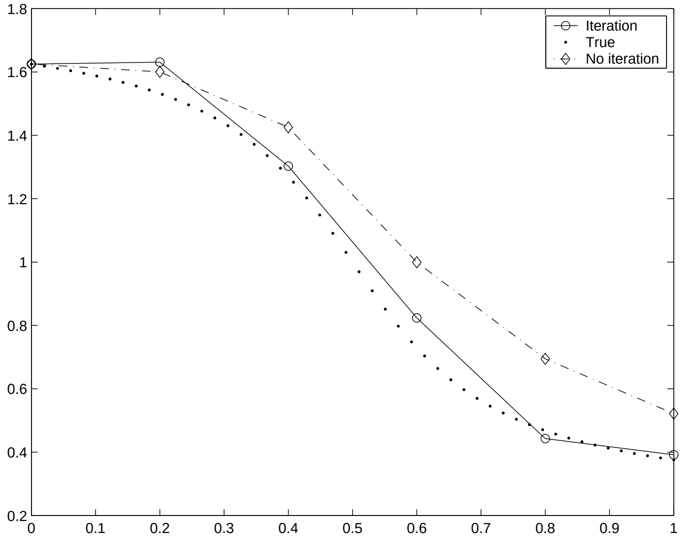


Figure 3.24: Recovery with iteration of function 3.8

was recovered in two ways. In the first, the non-iterative algorithm applied to the data $\{f, g\}$ to produce the dashed line plot, while the iterative algorithm was applied to produce the solid line plot. The data was generated by solving the direct problem (2.1) using a functional form of the coefficient (3.8) on a mesh of 70 nodes with the Matlab solver `ode15s`. The flux, $g(t)$, was then computed using a difference formula. This flux data was submitted to the recovery algorithms, both which used a 40 node mesh and `ode15s` to compute solutions to the direct and adjoint problem. It is clear from the figure that the errors in non-iterated nodal values for $D(u)$ accumulate as the values are sequentially determined. We point out that determining d_k we are obliged to integrate over the approximately triangular region $\{0 < x < x_k(t), T_{k-1} < t < T_k\}$. However, the algorithm must numerically approximate $x_0(t_j)$ on the inner mesh, and this leads to a systematic overestimation of the value of A_{kk} which, in turn leads to a correction term that is too small. The fact that D is a decreasing function of u, as given in equation (3.8), leads to a negative $\Delta g(t)$ and a negative correction, b_k/A_{kk}. This is evident in the dashed-line plot of Figure 3.24. The fact that A_{kk} is too large causes the negative correction to be too small so that the graph of the computed polygonal function lies above the graph of the true coefficient. Since the integrals for A_{kk} *and* b_k involve only the interval $[T_{k-1}, T_k]$, each identified value, d_k, can do nothing to diminish errors in previously identified values, hence the identification error accumulates.

This suggests that iteration might prove useful. The solid line plot in Figure 3.24 shows the result of identifying the coefficient 3.8) but now iterating as follows. We use the identity (2.7) on Q_1 together with the known value, d_0, to identify $d_1^{(1)}$. Here the known value, d_0, is used to

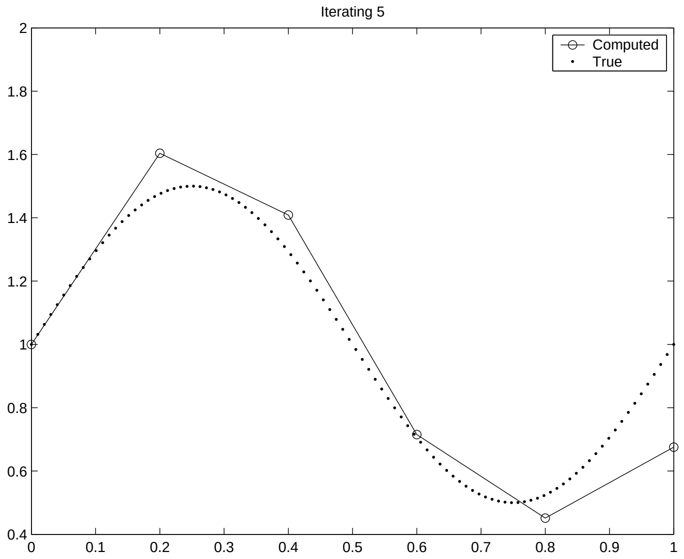


Figure 3.25: Recovery of $D(u) = 1 + \frac{1}{2}\sin(2\pi u)$

compute $u_2(x,t)$, $g_2(t)$ and $\hat{\phi}(x,t)$. Next we use the identity (2.7) on Q_1 and Q_2 together with known values, d_0, $d_1^{(1)}$ to identify $d_1^{(2)}$ and $d_2^{(1)}$. In the next step, we use the identity (2.7) on Q_1, Q_2 and Q_3 together with known values, $d_0, d_1^{(2)}$ and $d_2^{(1)}$ to identify $d_1^{(3)}, d_2^{(2)}$ and $d_3^{(1)}$. At each stage, the known nodal values are used to compute $u_2(x,t), g_2(t)$ and $\hat{\phi}(x,t)$. Continuing in this way, we eventually obtain $d_M^{(1)}, d_{M-1}^{(2)}, ..., d_1^{(M)}$. It is evident from the solid line plot in Figure 3.24 that as a result of the iteration, the errors no longer exhibit the cumulative character seen in the dashed line plot, where iteration was not applied. Here the coefficient

$$D(u) = 1 + \frac{1}{2}\sin(2\pi u), \qquad 0 < u < 1, \tag{3.9}$$

was used to generate flux data as in the previous example, although here the Matlab solver `ode23s` was used. This data was passed to the iterative recovery algorithm, the results of which are plotted in figure 3.25 The qualitative agreement between the computed and true coefficient appears

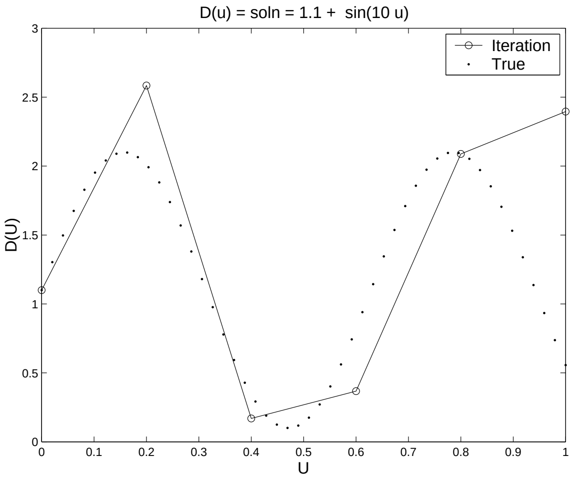


Figure 3.26: Recovery of $D(u) = 1.1 + \sin(10u)$

reasonable in this figure. Notice that the approximation initially lies above the plot of the true coefficient, denoted by the dotted line, in regions where D is increasing. This is in agreement with the analysis of the previous experiment. The value at the last nodal is not iterated in this scheme, and is visibly less accurate than the computed values on other nodes.

The algorithm is able to capture a variety of coefficient types. In figure 3.11, uniform time breaks induce a uniform nodal basis under linear forcing. The sine function

$$D(u) = 1.1 + \sin(10u)$$

is captured quite well. Although the nodal basis used was uniform, the algorithm identified both the amplitude and the frequency of the coefficient quite accurately. Again, the recovery is not iterated in the last interval from $u = [0.8, 1]$ and as a consequence, is less accurate.

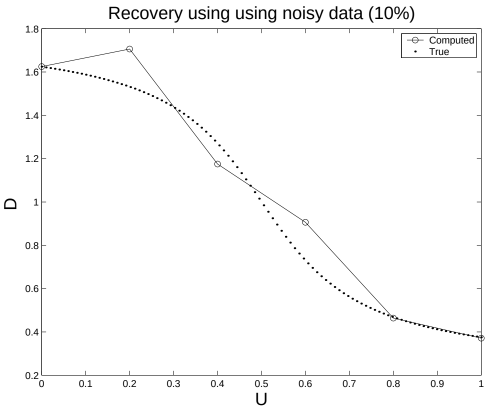


Figure 3.27: Recovery from data with 10% noise

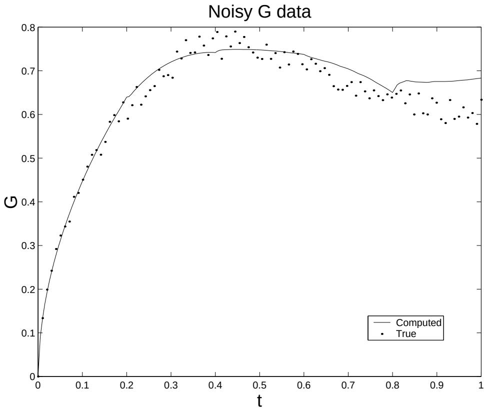


Figure 3.28: Flux data used by 3.27

3.12 Noisy data

In this section, we explore the effect that a noisy data set has on the recovery process.

Figure 3.27 represents coefficient recovery in which the data contained induced error. A relative uniform random error of 10% was induced in the flux data, and the iterative algorithm was applied. The flux data used for recovery is plotted in figure 3.28. The recovered coefficient, plotted in Figure 3.27 appears to capture the general structure of the true coefficient. No preprocessing was applied to this data, which was possible since the

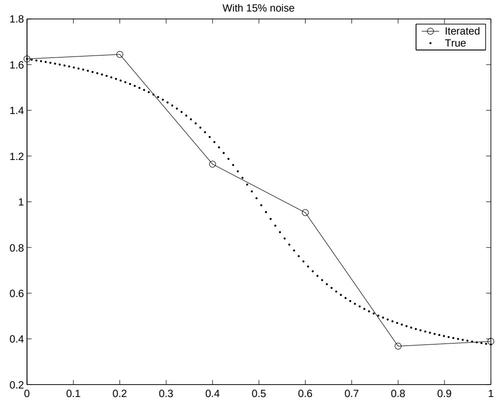


Figure 3.29: Recovery from data with 15% noise

error had mean zero. The integration of the g data in (2.13) allows much of this error to cancel. This is a significant observation. Often, even slight error will cause the parameter estimation difficulty [20]. In practice, noisy data is often fit with a spline, with the subsequent smooth fit replacing the noisy measurement. In figure 3.29, a 15% uniform random error was introduced. While noticeably worse than the previous recovery, the main features of the coefficient were still successfully identified.

Chapter 4

TWO PARAMETER IDENTIFICATION

Consider the quasilinear conduction diffusion equation given by

$$\begin{aligned} C(h)\partial_t h(z,t) - \partial_z(K(h)(\partial_z h(z,t) - \cos(\vartheta))) =& S(z,t), \qquad \text{in } Q_T \\ h(z,0) =& 0, \qquad 0 < z < L, \\ \partial_z h(0,t) - 1 = 0 \quad h(L,t) =& g(t) \qquad 0 < t < T. \end{aligned} \tag{4.1}$$

Although this equation can represent many physical systems, we consider it here to be the governing equation for ground water flow through porous media. In this application, Equation (4.1) is referred to as the one dimensional Richards equation in capacity conductivity form. The variable $h(z,t)$ represents capillary pressure head, the angle ϑ indicates flow declination, and $S(z,t)$ introduces a source/sink term. The function $C(h)$ represents the water capacity function of the soil while $K(h)$ represents the hydraulic conductivity function. In our application, these functions will be considered unknown. In this chapter we develop theory which allows the identification of these unknown quantities using observable boundary measurements, under suitable restrictions.

4.1 Phase 1 problem

Consider a vertical soil column which is completely saturated and then allowed to drain under gravity. If there are no sources or sinks, no flow across the top of the column and if the bottom of the column is at the water table, the capillary pressure head $h(z,t)$ can be shown to satisfy

$$\begin{aligned} C(h)\partial_t h(z,t) &= \partial_z(K(h)(\partial_z h(z,t)-1)) && \text{for } (z,t)\in Q_T \\ h(z,0) &= 0 && \text{for } 0<z<L, \\ \partial_z h(0,t)-1 = 0 \quad h(L,t) &= 0 && \text{for } 0<t<T \end{aligned} \tag{4.2}$$

where $Q_T = \{(z,t) : 0 < z < L, 0 < t < T\}$. The column is assumed to be of length L with $z = 0$ at the top of the column and $z = L > 0$ at the bottom. Here ϑ was taken to be the angle from the positive z axis.

Problem (4.2) will be referred to as the Phase 1 direct problem. For suitable coefficients C and K this direct problem has a unique smooth solution [8, 22]. The solution tends toward the steady state equal to $h(z) = z - L$ for $0 < z < L$ as t tends to infinity. While this state is never reached in finite time, for any choice of ϵ, a time T can be found such that $h(z,T)$ is within ϵ of the steady state. For (z,t) in the region Q_T, the range of $h(z,t)$ is $[-L+\epsilon, 0]$ and not the interval $(-L, 0]$. We, however, will take T to be a sufficiently large fixed constant so as to allow the head values $h(z,t)$ to vary between 0 and $-L$ and omit reference to ϵ in future discussions.

Coefficients C and K are *admissible* if they satisfy

$$C \in \mathcal{C}(h) \text{ and } C^\flat \le C(h) \le C^\sharp \text{ for } h \in J, \tag{4.3}$$

$$|K(h_1) - K(h_2)| \le \kappa|h_1 - h_2| \text{ and } K^\flat \le K(h) \le K^\sharp \text{ for } h_1, h_2 \in J, \tag{4.4}$$

where $J = [h(0,T), h(0,0)]$ and $C^\flat, C^\sharp, K^\flat, K^\sharp$ and κ are positive constants. Then any polygonal function bounded away from zero satisfies these conditions, and the difference of two such functions has at most finitely many zeros for h in J. For each pair of admissible coefficients, the direct problem (4.2) has a unique solution $h(z,t)$.

Restricting our discussion to admissible coefficients, we seek to recover coefficients C and K from measured output. We recall that simultaneous control of h and $\partial_z h$ at a single point is not possible. However, one can be controlled and the other observed. In equation (4.2), $\partial_z h(0,t)$ and $h(L,t)$ are the controlled quantities, allowing the observation of $h(0,t) = p(t)$ and $\partial_z(h(L,t)) - 1 = q(t)$. Although several outputs are experimentally feasible, we restrict to these observations made on the boundary.

We define the mappings

$$\begin{aligned} &\Phi : W(J) \longrightarrow L^2[0,T] \\ &\Phi[C,K] = h(z,t) \end{aligned} \tag{4.5}$$

with $W(J)$ representing the class of admissible coefficient pairs. Evidently, Φ is the solution map from the input coefficients $[C,K]$ to h. We also define the projection map Γ, which assigns to each solution the pair of accessible output measurements p and q. We denote this by $\Gamma \cdot \Phi[C,K] = (p,q)$. The solution of the inverse problem will amount to the inversion of this map. We also write $\Gamma_0 \cdot \Phi[C,K] = p$ and $\Gamma_L \cdot \Phi[C,K] = q$, denoting the evaluation of this map at $z = 0$ and $z = L$, respectively.

We choose in this paper to explore the map Φ using adjoint techniques. First, note that we may rewrite several term in (4.2) in a form that will later prove useful. Let

$$a(h(z,t)) = \int_0^{h(z,t)} C(s)\ ds \quad \text{and} \quad b(h(z,t)) = \int_0^{h(z,t)} K(s)\ ds.$$

Notice that $C(h)\partial_t h = \partial_t a(h)$ and $\partial_z(K(h)\partial_z h)) = \partial_{zz} b(h)$.

We also recall a maximum principle for the heat equation.

The Maximum Principle *Suppose that $w(z,t) \in \mathcal{C}^2(Q_T) \cap \mathcal{C}(\overline{Q_T})$. Let M and m denote, respectively, the maximum and minimum of w on the parabolic boundary of Q_T. Then,*

- *if $\partial_t w - \nabla^2 w \leq 0$ in Q_T, then $w \leq M$ in $\overline{Q_T}$*
- *if $\partial_t w - \nabla^2 w \geq 0$ in Q_T, then $w \geq m$ in $\overline{Q_T}$*
- *if $\partial_t w - \nabla^2 w = 0$ in Q_T, then $m \leq w \leq M$ in $\overline{Q_T}$*

Lemma 4.1.1. *For admissible coefficients C(h) and K(h), let $h = \Phi[C, K]$, be a solution to* (4.2) *with $q = \Gamma_L \cdot \Phi[C, K]$. Then for each $t, 0 < t < T$,*

$$q(t) \in \mathcal{C}[0,T) \quad q(0) = -K(0) \quad \textit{and} \quad q(t) < 0$$

Proof. The smoothness of the solution implies that $q(t)$ is continuous. Initial and boundary conditions immediately imply that $q(0) = -K(0)$.

We now show that $q(t) < 0$ through an adjoint method. For $q = \Gamma_L \cdot \Phi[C, K]$ and for an arbitrary smooth function $\phi(z,t)$, we write

$$\iint_{Q_T} [\partial_t a(h) - \partial_z(K(h)(\partial_z h(z,t) - 1)]\, \partial_z \phi \; dz \; dt = 0$$

Integration by parts yields

$$\begin{aligned}
&\iint_{Q_T} \partial_t a(h)\; \partial_z \phi \; dz\; dt \\
&\qquad = \int_0^L a(h)\partial_z\phi\Big|_{t=0}^{t=T} dz - \iint_{Q_T} a(h)\partial_{tz}\phi \; dz\; dt \\
&\qquad = \int_0^L a(h)\partial_z\phi\Big|_{t=0}^{t=T} dz - \int_0^T a(h)\partial_t\phi\Big|_{z=0}^{z=L} dt \\
&\qquad\qquad + \iint_{Q_T} C(h)\partial_z h \partial_t \phi \; dz\; dt
\end{aligned}$$

and

$$\iint_{Q_T} \left[\partial_z(K(h)(\partial_z h(z,t)-1))\right]\partial_z\phi\right]\, dz\; dt$$

$$= \int_0^T (K(h)(\partial_z h(z,t)-1))\,\partial_z\phi\Big|_{z=0}^{z=L}\, dt$$

$$-\iint_{Q_T} (K(h)(\partial_z h(z,t)-1))\,\partial_{zz}\phi\; dz\; dt.$$

So, we have the identity

$$\iint_{Q_T} [(\partial_z h-1)\{\,C(h)\partial_t\phi + K(h)\partial_{zz}\phi\,\}{+}C(h)\partial_t\phi]\;\; dz\; dt$$

$$= \int_0^T a(h)\partial_t\phi + K(h)(\partial_z h-1)\partial_z\phi\Big|_{z=0}^{z=L}\, dt - \int_0^L a(h)\partial_z\phi\Big|_{t=0}^{t=T}\, dz. \quad (4.6)$$

Notice that the initial and boundary conditions of the direct problem make

$$a(h(z,0)) = \int_0^{h(z,0)} C(s)\; ds = 0,$$

$$a(h(L,t)) = \int_0^{h(L,t)} C(s)\; ds = 0 \quad \text{and}$$

$$\partial_z h(0,t) - 1 = 0.$$

Applying these facts to (4.6) results in

$$\iint_{Q_T} [(\partial_z h-1)\{\,C(h)\partial_t\phi + K(h)\partial_{zz}\phi\,\}{+}C(h)\partial_t\phi]\;\; dz\; dt$$

$$= \int_0^T -a(h(0,t))\partial_t\phi(0,t) + q(t)\partial_z\phi(L,t) - \int_0^L a(h(z,T)\partial_z\phi(z,T)\; dz, \quad (4.7)$$

where $K(h(L,t))(\partial_z h(L,t)-1)$ has been replaced with the measurement $q(t)$. We now introduce an adjoint problem that will reduce (4.7) further. Assume $\phi(z,t)$ solves the adjoint problem

$$C(h)\partial_t\phi(z,t) + K(h)\partial_{zz}\phi(z,t) = 0 \qquad \text{in } Q_T,$$

$$\phi(z,T) = 0 \qquad 0<z<L,$$

$$\phi(0,t) = 0 \quad \partial_z\phi(L,t) = \theta(t) \qquad 0<t<T.$$

Then the initial condition implies that $\partial_z \phi(z,T) = 0$ and the boundary condition implies that $\partial_t \phi(0,t) = 0$.

Equation (4.6) finally collapses to the simple integral identity

$$\int_0^T q(t)\theta(t)dt = \iint_{Q_T} C(h)\partial_t \phi(z,t)\ dz\ dt. \tag{4.8}$$

Now notice that the right side can be written

$$\begin{aligned}\iint_{Q_T} C(h)\partial_t \phi(z,t)\ dz\ dt =& C(\tilde{h}) \int_0^L \phi(z,s)\Big|_0^T\ ds\ dz \\ =& C(\tilde{h}) \int_0^L \phi(z,0)\ dz,\end{aligned}$$

for some $\tilde{h} = h(\tilde{z}, \tilde{t})$ with $(\tilde{z}, \tilde{t}) \in Q_T$. Since C is strictly positive and $\phi(z,0)$ is strictly negative almost everywhere in Q_T, this term is strictly negative. Evidently, the integral identity implies that $q(t) < 0$ on $(0,T)$. □

We now develop an integral identity relating the input pair (C,K) to the output pair (p,q). We begin with two lemmas that will be essential in the analysis of the inverse problem.

Lemma 4.1.2. *For admissible coefficients C(h) and K(h), let* $h = \Phi[C,K]$ *and* $q = \Gamma_L \cdot \Phi[C,K]$. *Then* $\partial_z h(z,t) - 1 < 0$ *almost everywhere in* Q_T.

Proof. For $h = \Phi[C,K]$ and arbitrary smooth function $\phi(z,t)$, the integral identity (4.8) holds. Suppose $\phi(z,t)$ now solves the adjoint problem

$$\begin{aligned} C(h)\partial_t \phi(z,t) + K(h)\partial_{zz}\phi(z,t) = F(z,t) \qquad & \text{in } Q_T, \\ \phi(z,T) = 0, \qquad & 0 < z < L, \\ \phi(0,t) = 0, \quad \phi(L,t) = 0, \qquad & 0 < t < T. \end{aligned}$$

Since $a(h(z,0)) = 0, \quad a(h(L,t)) = 0$ and $\partial_z h(0,t) - 1 = 0$, and both $\partial_t \phi(0,t) = 0$ and $\partial_z \phi(z,T) = 0$, identity (4.6) collapses to

$$\begin{aligned}\iint_{Q_T} &(\partial_z h - 1) F(z,t)\ dz\ dt \\ &= -\iint_{Q_T} C(h) \partial_t \phi\ dz\ dt + \int_0^T q(t) \partial_z \phi(L,t)\ dt.\end{aligned}$$

We consider each term independently. Choosing the function $F(z,t)$ to be nonnegative in Q_T, the maximum priciple ensures that the solution $\phi(z,t)$ is also nonnegative in Q_T. By an argument similar to the one made in the previous lemma, it follows from the reduced integral identity that $\partial_z h(z,t) - 1 < 0$ almost everywhere in Q_T. □

Lemma 4.1.3. *For admissible coefficients C(h) and K(h), let* $h = \Phi[C,K]$ *and* $p = \Gamma_0 \Phi[C,K]$*. Then* $\partial_t h(z,t) < 0$ *almost everywhwere in* Q_T*.*

Proof. Let $h = \Phi[C,K]$ and $\phi(z,t)$ be an arbitrary smooth function. Consider

$$\iint_{Q_T} \left[C(h) \partial_t h - \partial_z (K(h)(\partial_z h - 1)) \right] \partial_t \phi\ dz\ dt = 0.$$

Integration by parts implies

$$\begin{aligned}-\iint_{Q_T} &\partial_z (K(h)(\partial_z h - 1)) \partial_t \phi\ dz\ dt \\ &= \iint_{Q_T} K(h)(\partial_z h - 1) \partial_{zt} \phi\ dzdt \\ &\quad - \int_0^T K(h)(\partial_z h - 1) \partial_t \phi \Big|_0^L\ dt \\ &= -\iint_{Q_T} \partial_t \left[K(h)(\partial_z h - 1) \right] \partial_z \phi\ dz\ dt \\ &\quad + \int_0^L K(h)(\partial_z h - 1) \partial_z \phi \Big|_0^T dz \\ &\quad - \int_0^T K(h)(\partial_z h - 1) \partial_t \phi \Big|_0^L\ dt.\end{aligned}$$

Now,

$$\begin{aligned}
&-\iint_{Q_T} \partial_t \left[K(h)(\partial_z h - 1)\right] \partial_z \phi \; dz \; dt \\
&\qquad = -\iint_{Q_T} \partial_{zt} b(h) \partial_z \phi \; dz \; dt + \iint_{Q_T} \partial_t K(h) \partial_z \phi \; dz \; dt \\
&\qquad = \iint_{Q_T} \partial_t b(h) \partial_{zz} \phi \; dz \; dt - \int_0^T \partial_t b(h) \partial_z \phi \Big|_0^L \; dt \\
&\qquad\quad + \iint_{Q_T} K'(h) \partial_t h \partial_t \phi dz \; dt.
\end{aligned}$$

Putting this together, we get

$$\begin{aligned}
\iint_{Q_T} \big[C(h)\partial_t\phi + K(h)\partial_{zz}\phi + K'(h)\partial_z\phi\big]\partial_t h \; dz \; dt \\
= \int_0^T K(h)(\partial_z h - 1)\partial_t\phi \Big|_0^L \; dt \qquad (4.9)\\
- \int_0^L K(h)(\partial_z h - 1)\partial_z\phi \Big|_0^T \; dz \qquad (4.10)\\
+ \int_0^T K(h)\partial_t h \partial_z\phi \Big|_0^L \; dz. \qquad (4.11)
\end{aligned}$$

Since $K(h)(\partial_z h - 1) = q(t)$ and $\partial_z h(0,t) - 1 = 0$, (4.9) becomes

$$\int_0^T K(h)(\partial_z h - 1)\partial_t\phi \Big|_0^L \; dt = \int_0^T q(t)\partial_t\phi(L,t) \; dt$$

if we make $\phi(L,T) = 0$. Taking $\phi(z,T) = 0$ implies that $\partial_z\phi(z,T) = 0$. Combining this with the fact that $h(z,0) = 0$ implies $\partial_z h(z,0) = 0$, (4.10) may be written

$$\begin{aligned}
-\int_0^L K(h)(\partial_z h - 1)\partial_z\phi \Big|_0^T \; dz \\
= \int_0^L K(h)(\partial_z h - 1)\partial_z\phi \Big|_0 \; dz \\
= K(0)\int_0^L \partial_z\phi(z,0) \; dz \\
= K(0)\left[\phi(L,0) - \phi(0,0)\right].
\end{aligned}$$

Finally, if we set both $\partial_z\phi(0,t) = 0$ and $\partial_z\phi(L,t) = 0$, then (4.11) will vanish. So, if we allow $\phi(z,t)$ to satisfy the adjoint problem

$$\begin{aligned} C(h)\partial_t\phi + K(h)\partial_{zz}\phi + K'(h)\partial_z\phi = F(z,t) \qquad & \text{in } Q_T, \\ \phi(z,T) = 0, \qquad & 0 < z < L, \\ \partial_z\phi(0,t) = 0, \quad \partial_z\phi(L,t) = 0 \qquad & 0 < t < T, \end{aligned}$$

then we may write

$$\iint_{Q_T} \partial_t h F(z,t)\ dz\ dt = K(0)[\phi(L,0) - \phi(0,0)] + \int_0^t q(t)\partial_t\phi(L,t).$$

This can be rewritten as

$$\begin{aligned} \iint_{Q_T} \partial_t h F(z,t)\ dz\ dt =& K(0)\left[\phi(L,0) - \phi(0,0)\right] + q(\tilde{t}) \int_0^t \partial_t\phi(L,t)dt \\ =& K(0)\left[\phi(L,0) - \phi(0,0)\right] + q(\tilde{t})\phi(L,t)\Big|_0^T \\ =& K(0)\left[\phi(L,0) - \phi(0,0)\right] - q(\tilde{t})\phi(L,0), \end{aligned}$$

where $0 < \tilde{t} < t$. Restricting $F(z,t)$ to be nonnegative but otherwise arbitrary, the maximum principle will ensure that $\phi(z,0)$ will be negative almost everywhere. It has been shown that $q(t) < 0$. Since $K(0)$ is nonnegative, the entire right side is negative if we force $\phi(L,0) - \phi(0,0)$ negative by adjusting F. Evidently, $\partial_t h$ must be nonnegative almost everywhere in Q_T. □

Lemma 4.1.4. *For admissible coefficients C(h) and K(h), let* $h = \Phi[C,K]$. *Then for each* $t, 0 < t < T$,

$$z - L < h(z,t) < 0 \quad \textit{and} \quad h(0,\tau) < h(z,t) < 0$$

for $0 < z < L$ *and* $0 < t < \tau$.

Proof. Let $h(z,t) = \Phi[C,K]$. Define $u(z,t) =: h(z,t) - (z - L)$. Then consider the transformed problem

$$\begin{aligned} C(h)\partial_t u - \partial_z(K(h)\partial_z u = 0 & \qquad \text{in } Q_T, \\ u(z,T) = 0, & \qquad 0 < z < L, \\ \partial_z u(0,t) = 0, \quad u(L,t) = 0 & \qquad 0 < t < T, \end{aligned}$$

By the maximum-minimum principle, $0 < u(z,t) < L - z$ for each t in $(0,T]$, which in turn implies $z - L < h(z,t) < 0$ in Q_T. Since $\partial_z h(0,t) = 1$, the maximum must occur on the boundary $z = 0$. Therefore, $h(0,t) < h(z,t) < 0$ for all (z,t) in Q_T. □

Lemma 4.1.5. *For admissible coefficients C(h) and K(h), let* $p = \Gamma_0 \cdot \Phi[C,K]$. *Then*

$$p \in \mathcal{C}^1[0,T], \quad p(0) = 0 \quad \textit{and} \quad p'(t) < 0$$

Proof. The smoothness of p follows from the smoothness of the solution, as does the consistency at $t = 0$. To show that $p'(t)$ is negative, we choose a smooth function $\phi(z,t)$ which satisfies the adjoint problem

$$\begin{aligned} C(h)\partial_t\phi + K(h)\partial_{zz}\phi + K'(h)\partial_z\phi = 0 & \qquad \text{in } Q_T \\ \phi(z,T) = 0 & \qquad 0 < z < L \\ \partial_z\phi(0,t) = \theta(t) \quad \phi(L,t) = 0 & \qquad 0 < t < T. \end{aligned}$$

Then the earlier equation identity reduces to

$$K(0)\int_0^L \partial_z\phi(z,0)\ dz = -\int_0^T K(h(0,t))\,\partial_t h(0,t)\theta(t)\ dt$$

The fundamental theorem of calculus coupled with the boundary conditions allows us to write

$$K(0)\phi(0,0) = \int_0^T K(h(0,t))p'(t)\theta(t)\ dt.$$

Now select $\theta(t)$ to be negative for t in $[0, T)$ and $\theta(T) = 0$. The extended maximum principle implies that ϕ assumes its minimum at $z = 0$, and that $\phi(z, t)$ is strictly positive in the interior Q_T. The positivity of K then makes clear that the left side is strictly positive. Evidently, our choice of θ allows us to conclude that $p'(t)$ must be negative for $0 < t < T$. $\square$

Before we begin the proof of the main result, we adopt the notations

$$a_i(h) := \int_0^{h(z,t)} C_i(s)\ ds \quad \text{and} \quad b_i(h) := \int_0^{h(z,t)} K_i(s)\ ds,$$

which will streamline the presentation. We are able to make several observations that will prove useful. Notice that

$$\begin{aligned} a_1(h_1) - a_1(h_2) &= \int_{h_2}^{h_1} C_1(s)\ ds \\ &= \int_0^1 C_1(\lambda h_1 + (1-\lambda)h_2)\ d\lambda\ (h_1 - h_2) \\ &= C_1(\tilde{h}(z,t))(h_1 - h_2) \\ &= C_1^*(z,t)\ (h_1 - h_2) \end{aligned}$$

for some $\tilde{h}$ between h_1 and h_2. Similarly,

$$\begin{aligned} b_1(h_1) - b_1(h_2) &= \int_{h_2}^{h_1} K_1(s)\ ds \\ &= \int_0^1 K_1(\lambda h_1 + (1-\lambda)h_2)\ d\lambda\ (h_1 - h_2) \\ &= K_1(\hat{h}(z,t))(h_1 - h_2) \\ &= K_1^*\ (h_1 - h_2) \end{aligned}$$

and

$$\begin{aligned} K_1(h_1) - K_1(h_2) &= \int_{h_2}^{h_1} K_1'(s)\ ds \\ &= \int_0^1 K_1'(\lambda h_1 + (1-\lambda)h_2)\ d\lambda (h_1 - h_2) \\ &= K_1^{'*}\ (h_1 - h_2). \end{aligned}$$

We remark that C_1^*, K_1^* and $K_1^{'*}$ all take their values at independent and indeterminant values of h. Also,

$$\begin{aligned}
\partial_t[a_2(h_2) - a_1(h_2)] &= \partial_t \int_0^{h_2} C_2(s) - C_1(s) ds \\
&= [C_2(h_2) - C_1(h_2)]\, \partial_t h_2 \\
\partial_z[b_2(h_2) - b_1(h_2)] &= \partial_z \int_0^{h_2} K_2(s) - K_1(s) ds \\
&= [K_2(h_2) - K_1(h_2)]\, \partial_z h_2 \\
\partial_z[K_2(h_2) - K_1(h_2)] &= \partial_z \int_0^{h_2} K_2'(s) - K_1'(s) ds \\
&= [K_2'(h_2) - K_1'(h_2)]\, \partial_z h_2
\end{aligned}$$

We now present the general integral identity. This identity is fundamental to the analysis of the inverse problem.

Theorem 4.1.1. *For admissible coefficients $C_i(h)$ and $K_i(h)$, let $h_i = \Phi[C_i, K_i]$ denote the solution and $(p_i, q_i) = \Gamma \cdot \Phi[C_i, K_i]$ the observation data for $i = 1, 2$. For arbitrary smooth functions $P^*(t), Q^*(t)$, let $\phi = \Phi^*[P^*, Q^*]$ represent the solution to the adjoint initial value problem*

$$\begin{aligned}
C_1^* \partial_t \phi(z,t) + K_1^*(z,t) \partial_{zz} \phi(z,t) + K_1^{'*}(z,t) \partial_z \phi(z,t) = 0 \qquad & \textit{in } Q_T \\
\phi(z,\tau) = 0 \qquad & 0 < z < L \\
K_1^*(0,t)(\partial_z(0,t) - 1) = P^*(t), \quad \phi(L,t) = Q^*(t), \quad & 0 < t < \tau
\end{aligned} \tag{4.12}$$

where

$$\begin{aligned}
C_1^*(z,t)\ (h_1 - h_2) &= \int_{h_2(z,t)}^{h_1(z,t)} C_1(s)\ ds \\
K_1^*(z,t)\ (h_1 - h_2) &= \int_{h_2(z,t)}^{h_1(z,t)} K_1(s)\ ds \\
K_1^{'*}(z,t)\ (h_1 - h_2) &= \int_{h_2(z,t)}^{h_1(z,t)} K_1'(s)\ ds
\end{aligned}$$

Changes in the inputs $\Delta C = C_1 - C_2$ and $\Delta K = K_1 - K_2$ are related to changes in the output $\Delta p = p_1 - p_2$ and $\Delta q = q_1 - q_2$. For any $\tau,\ 0 < \tau < T$, this relationship is

$$\int_0^T \int_0^L \partial_z h[\Delta K(h_2)(\partial_z h_2(z,t) - 1)\partial_z \phi + \Delta C(h_2)\phi \partial_t h_2]\ dzdt \\ = \int_0^T [\Delta p P^*(t) + \Delta q Q^*(t)]\ dt \quad (4.13)$$

Proof. Consider the pair of initial IBVPs

$$\begin{aligned} \partial_t a_i(h_i) &= \partial_z(K_i(h_i)(\partial_z h_i - 1)) && \text{in } Q_T \\ h_i(z,0) &= 0 && \text{for } 0 < z < L, \\ \partial_z h_i(0,t) - 1 = 0 \quad h_i(1,t) &= 0 && \text{for } 0 < t < T. \end{aligned}$$

for $i = 1$ and 2.

Subtracting the two yields

$$\begin{aligned} \partial_t(a_1(h_1) - a_2(h_2)) &= \partial_z(K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1)) \\ h_1(z,0) - h_2(z,0) &= 0 \\ \partial_z(h_1(0,t) - h_2(0,t)) = 0 \quad h_1(L,t) - h_2(L,t) &= 0 \end{aligned}$$

We now multiply each term of the above equation by a smooth function $\phi(z,t)$ and integrate over space and time,

$$\int_0^T \int_0^L \partial_t(a_1(h_1) - a_2(h_2))\phi\ dz\ dt \\ = \int_0^T \int_0^L \partial_z(K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1)\phi\ dz\ dt$$

A slight rearrangement produces

$$\int_0^T \int_0^L \partial_t(a_1(h_1) - a_1(h_2))\phi\ dz\ dt \\ - \int_0^T \int_0^L \partial_z(K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\phi\ dz\ dt \\ = \int_0^T \int_0^L \partial_t(a_2(h_2) - a_1(h_2))\phi\ dz\ dt. \quad (4.14)$$

We now integrate each term on the left side by parts. The first term becomes

$$\begin{aligned}
&\int_0^T \int_0^L \partial_t(a_1(h_1) - a_1(h_2))\phi(z,t)dz\ dt \\
&= \int_0^L (a_1(h_1) - a_1(h_2))\phi(z,t)\Big|_0^T\ dz - \int_0^T \int_0^L (a_1(h_1) - a_1(h_2))\partial_t\phi(z,t)dz\ dt \\
&= \int_0^L C_1^*\ (h_1 - h_2)\phi(z,t)\Big|_0^T\ dz - \int_0^T \int_0^L C_1^*\ (h_1 - h_2)\partial_t\phi(z,t)dz\ dt,
\end{aligned} \tag{4.15}$$

and the second,

$$\int_0^T \int_0^L \partial_z(K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\phi\ dz\ dt$$

$$= \int_0^T (K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\phi\Big|_0^L\ dt \tag{4.16}$$

$$- \int_0^T \int_0^L (K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\partial_z\phi\ dt. \tag{4.17}$$

We remark that the flux term $K(h)(\partial_z h - 1)$ is controlled at $z = 0$ and observed at $z = L$. Notice that the spatial boundary term (4.16) above contains the flux quantity. Now, consider (4.17).

$$\int_0^T \int_0^L (K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\partial_z\phi\ dt.$$

$$= \int_0^T \int_0^L \left[K_1(h_1)\partial_z h_1 - K_2(h_2)\partial_z h_2\right]\partial_z\phi\ dz\ dt \tag{4.18}$$

$$+ \int_0^T \int_0^L \left[K_2(h_2) - K_1(h_1)\right]\partial_z\phi\ dz\ dt \tag{4.19}$$

A reformulation of (4.18) leads to

$$\begin{aligned}
&\int_0^T \int_0^L [K_1(h_1)\partial_z h_1 - K_2(h_2)\partial_z h_2]\,\partial_z \phi \; dz \; dt \\
&\quad = \int_0^T \int_0^L \partial_z(b_1(h_1) - b_2(h_2)\partial_z \phi \; dz \; dt \\
&\quad = \int_0^T \int_0^L (b_1(h_1) - b_1(h_2)\partial_z \phi \; dz \; dt - \int_0^T \int_0^L \partial_z(b_2(h_2) - b_1(h_2)\partial_z \phi \; dz \; dt \\
&\quad = \int_0^T (b_1(h_1) - b_1(h_2)\partial_z \phi \Big|_0^L \; dt - \int_0^T \int_0^L (b_1(h_1) - b_1(h_2)\partial_{zz} \phi \; dz \; dt \\
&\qquad - \int_0^T \int_0^L \partial_z(b_2(h_2) - b_1(h_2)\partial_z \phi \; dz \; dt \\
&\quad = \int_0^T K_1^*(h_1 - h_2)\partial_z \phi \Big|_0^L \; dt - \int_0^T \int_0^L K_1^*(h_1 - h_2)\partial_{zz} \phi \; dz \; dt \\
&\qquad - \int_0^T \int_0^L (K_2(h_2) - K_1(h_2))\partial_z h_2 \partial_z \phi \; dz \; dt. \qquad (4.20)
\end{aligned}$$

We also consider (4.19)

$$\begin{aligned}
&\int_0^T \int_0^L [K_1(h_1) - K_1(h_2)]\partial_z \phi \; dz \; dt \\
&- \int_0^T \int_0^L [K_2(h_2) - K_1(h_2)]\partial_z \phi \; dz \; dt \\
&\qquad = \int_0^T \int_0^L K_1^{\prime *}(h_1 - h_2)\partial_z \phi \; dz \; dt \\
&\qquad\quad - \int_0^T \int_0^L [K_2(h_2) - K_1(h_2)]\partial_z \phi \; dz \; dt \qquad (4.21)
\end{aligned}$$

The final term of (4.14) can be written

$$\begin{aligned}
&\int_0^T \int_0^L \partial_t(a_2(h_2) - a_1(h_2))\phi \; dz \; dt \\
&\qquad = \int_0^T \int_0^L (C_2(h_2) - C_1(h_2))\partial_t h_2 \; dz \; dt \\
&\qquad = \int_0^T \int_0^L \Delta C(h_2)\partial_t h_2 \; dz \; dt \qquad (4.22)
\end{aligned}$$

Substitute (4.16, 4.20, 4.21) and this last expression into the full integral equation (4.14). After some rearrangement, we have

$$\begin{aligned}
&\int_0^T \int_0^L (h_1 - h_2)[C_1^* \partial_t \phi(z,t) + K_1^* \partial_{zz}\phi(z,t) + K_1^{'*} \partial_z \phi(z,t)]\ dz\ dt \\
&+ \int_0^T \int_0^L (C_1(h_2) - C_2(h_2))\partial_t h_2 \phi(z,t) dz\ dt \\
&+ \int_0^T \int_0^L (K_1(h_2) - K_2(h_2))(\partial_z h_2 - 1)\partial_z \phi(z,t) dz\ dt \\
&\qquad = \int_0^T (K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2)(\partial_z h_2 - 1))\phi(z,t)\Big|_0^L\ dt \\
&\qquad + \int_0^T K_1^*(h_1 - h_2)\partial_z \phi(z,t)\Big|_0^L dt \\
&\qquad + \int_0^L C_1^*(h_1 - h_2)\phi(z,t)\Big|_0^T\ dz.
\end{aligned}$$

Several terms now vanish. The initial and boundary conditions of the forward problem, which we recall to be

$$\begin{aligned}
h_1(z,0) - h_2(z,0) &= 0, \\
\partial_z(h_1(0,t) - h_2(0,t)) &= 0, \\
\text{and} \quad h_1(L,t) - h_2(L,t) &= 0,
\end{aligned}$$

allow us to write the reduced expression

$$\begin{aligned}
&\int_0^T \int_0^L (h_1 - h_2)[C_1^* \partial_t \phi(z,t) + K_1^* \partial_{zz}\phi(z,t) + K_1^{'*} \partial_z \phi(z,t)]\ dz\ dt \\
&\qquad + \int_0^T \int_0^L (C_1(h_2) - C_2(h_2))\partial_t h_2 \phi(z,t) dz\ dt \\
&\qquad + \int_0^T \int_0^L (K_1(h_2) - K_2(h_2))(\partial_z h_2 - 1)\partial_z \phi(z,t) dz\ dt \\
&\qquad\quad = \int_0^T (K_1(h_1)(\partial_z h_1 - 1) - K_2(h_2))(\partial_z h_2 - 1)\phi(z,t)\Big|^L\ dt \\
&\qquad\quad + \int_0^T K_1^*(z,t)(h_1 - h_2)\partial_z \phi(z,t)\Big|_0 dt \\
&\qquad\quad - \int_0^L C_1^*(h_1 - h_2)\phi(z,t)\Big|^T\ dz.
\end{aligned}$$

If we define

$$q_i(t) := K_i(L,t)(\partial_z h_i(L,t) - 1)$$
$$p_i(t) := h_i(0,t) \quad \text{for } i = 1,2,$$

we have

$$\begin{aligned}
\int_0^T \int_0^L (h_1 - h_2)[C_1^* \partial_t \phi(z,t) + K_1^* \partial_{zz}\phi(z,t) + K_1^{'*} \partial_z \phi(z,t)] \; dz \; dt \\
+ \int_0^T \int_0^L (C_1(h_2) - C_2(h_2))\partial_t h_2 \phi(z,t) dz \; dt \\
+ \int_0^T \int_0^L (K_1(h_2) - K_2(h_2))(\partial_z h_2 - 1)\partial_z \phi(z,t) dz \; dt \\
= \int_0^T (q_1(t)) - q_2(t))\phi(z,t) \; dt \\
+ \int_0^T K_1^*(z,t)(p_1(t) - p_2(t))\partial_z \phi(z,t) dt \\
- \int_0^L C_1^*(h_1 - h_2)\phi(z,t)\Big|^T \; dz.
\end{aligned}$$

Let $\phi = \phi(z, t : P^*, Q^*)$ now satisfy the associated adjoint problem

$$\begin{aligned}
C_1^* \partial_t \phi(z,t) + K_1^* \partial_{zz}\phi(z,t) + K_1^{'*}(z,t)\partial_z \phi(z,t) = 0 \qquad & \text{for } (z,t) \in Q_T \\
h_i(z,T) = 0 \qquad & \text{for } 0 < z < L, \\
K_1^*(0,t)\partial_z \phi(0,t) = P^*(t), \quad \phi(L,t) = Q^*(t) \qquad & \text{for } 0 < t < T.
\end{aligned} \tag{4.23}$$

Finally, we have

$$\begin{aligned}
\int_0^T \int_0^L (C_1(h_2) - C_2(h_2)) \; \partial_t h_2 \; \phi(z,t) dz \; dt \\
+ \int_0^T \int_0^L (K_1(h_2) - K_2(h_2)) \; (\partial_z h_2 - 1) \; \partial_z \phi(z,t) dz \; dt \\
= \int_0^T (q_1(t)) - q_2(t)) \; Q^*(t) \; dt \\
+ \int_0^T (p_1(t) - p_2(t)) \; P^*(t) \; dt,
\end{aligned} \tag{4.24}$$

which is the final form of the general integral identity for the two parameter problem. □

The integral identity above, (4.24), is an explicit relationship between the measurable output, quantities associated with the adjoint problem, and difference in the unknown coefficients $C(h)$ and $K(h)$. Since we wish to identify both C and K simultaneously, we consider two distinct solutions to the adjoint problem (4.23). If we take $\phi = \Phi^*[P^*, 0]$ and $\psi = \Phi^*[0, Q^*]$, it is easy to generate the pair of integral identities

$$\begin{aligned}\int_0^\tau P^*(t)[p(t) - p_2(t)]\, dt = &\int_0^\tau \int_0^L (K(h_2) - K_2(h_2))\partial_z\phi(\partial_z h_2 - 1)\ dz\ dt \\ &+ \int_0^\tau \int_0^L (C(h_2) - C_2(h_2))\partial_t h_2 \phi\ dt\ dz, \quad (4.25)\end{aligned}$$

which we refer to as the p-integral identity, and

$$\begin{aligned}\int_0^\tau Q^*(t)[q(t) - q_2(t)]\, dt = &\int_0^\tau \int_0^L (K(h_2) - K_2(h_2))\partial_z\psi(\partial_z h_2 - 1)\ dz\ dt \\ &+ \int_0^\tau \int_0^L (C(h_2) - C_2(h_2))\partial_t h_2 \psi\ dt\ dz \quad (4.26)\end{aligned}$$

which we call the q-integral identity.

Before we begin a discussion of the identifiability of C and K from the data pair (p, q), we first make some observations about the adjoint solutions $\phi = \Phi^*[P^*, 0]$ and $\psi = \Phi^*[0, Q^*]$. Recall that $\phi(z,t)$ and $\psi(z,t)$ solve

$$\begin{aligned} C^*(h)\partial_t\phi(z,t) + K^*\partial_{zz}\phi(z,t) + K^{'*}\partial_z\phi(z,t) &= 0 \qquad \text{in } Q_T \\ \phi(z,\tau) &= 0 \qquad 0 < z < L \\ K^*(0,t)\partial_z(0,t) = P^*(t), \quad \phi(L,t) &= 0, \qquad 0 < t < \tau \end{aligned}$$

and

$$\begin{aligned} C^*(h)\partial_t\psi(z,t) + K^*\partial_{zz}\psi(z,t) + K^{'*}\partial_z\psi(z,t) = 0 & \qquad\qquad \text{in } Q_T \\ \psi(z,\tau) = 0 & \qquad\quad 0 < z < L \\ K^*(0,t)\partial_z\psi(0,t) = 0, \quad \psi(L,t) = Q^*(t), & \qquad 0 < t < \tau, \end{aligned}$$

respectively. Recasting these final value problems as initial value problems, the maximum principle allows us to assert that sufficiently large and negative value of P^* will lead to a solution ϕ of the final value problem with the property that $\partial_z\phi(z,t) > 0$. Similarly, a choice of $Q^*(t)$ to be sufficiently monotone negative makes $\partial_z\psi(z,t) < 0$.

Now, the lemmas 4.1.2 and 4.1.3, coupled with the previous theorem, allow us to quickly establish the identifiability of C and K from the data pair (p,q).

Lemma 4.1.6. *For admissible coefficients C_i and K_i, let $(p_i,q_i) = \Gamma\cdot\Phi[C_i,K_i]$. If $\Delta p = p_1 - p_2$ and $\Delta q = q_1 - q_2$ are both identically zero, then $\Delta C = C_1 - C_2$ and $\Delta K = K_1 - K_2$ are also both identically zero.*

Proof. Let C_i and K_i be admissible coefficients for $i = 1$ and 2. If ΔC and ΔK are not identically zero, then there exist numbers, $\tau > 0$ and $h_* < 0$ such that

$$h_\flat \le h_2(z,t) \le 0 \text{ for } 0 \le z \le L,\ 0 \le t \le \tau.$$

Then, at least one of the functions $\Delta C(h_2(z,t))$ or $\Delta K(h_2(z,t))$ is of one sign in the region $(0,L)\times(0,\tau)$ on a positive length subinterval of $[h_\flat, 0]$. Now apply the p and q identities (4.25, 4.26) over this region $(0,L)\times(0,\tau)$, and utilize the fact that $\Delta p = \Delta q = 0$ on $[0, \tau]$. This gives

$$\int_0^\tau \int_0^L \Delta K(h_2))\partial_z \phi(\partial_z h_2 - 1)\ dz\ dt = -\int_0^\tau \int_0^L \Delta C(h_2)\partial_t h_2 \phi\ dz\ dt \tag{4.27}$$

$$\int_0^\tau \int_0^L \Delta K(h_2)\partial_z \psi(\partial_z h_2 - 1)\ dz\ dt = -\int_0^\tau \int_0^L \Delta C(h_2)\partial_t h_2 \psi\ dz\ dt \tag{4.28}$$

Lemma (4.1.2) affirms that $\partial_z h_2 - 1 > 0$ and lemma (4.1.3 that $\partial_t h_2 < 0$. In the remark above, we have argued that suitable choices of P^* and Q^* force $p_z \phi(z, t) > 0$ and $\partial_z \psi(z, t) < 0$. But equation (4.27) implies that ΔC and ΔK are both of the same sign, while (4.28) implies that they are of different sign. We initially assumed that both are not identically zero. We have reached a contradiction. Evidently, an identical data pair must lead to an identical coefficient pair. □

4.2 Phase 2 problem

The Phase 1 experiment allows exploration of the coefficients C and K in the parameter range from $h \in (0, L]$. We now turn to coefficient recovery in the Phase 2 experiment, in which a much larger parameter range may be visited. Recall that in this situation, the initial condition is given by $h(z, 0) = z - L$, which is the equilibrium solution to the Phase 1 problem. Imposing no flow conditions at the top of the column ($z = 0$) and applying suction at the base of the column ($z = L$), the capillary pressure head $h(z, t)$ can be shown to satisfy

$$\begin{aligned} C(h)\partial_t h(z, t) &= \partial_z(K(h)(\partial_z h(z, t) - 1)) && \text{for } (z, t) \in Q_T \\ h(z, 0) &= z - L && \text{for } 0 < z < L, \\ \partial_z h(0, t) - 1 = 0 \quad h(L, t) &= s(t) && \text{for } 0 < t < T \end{aligned} \tag{4.29}$$

where $Q_T = \{(z,t) : 0 < z < L, 0 < t < T\}$ and $s(t)$ satisfies the conditions

$$s(t) \in \mathcal{C}^1[0,T], \quad s(0) = 0 \quad \text{and} \quad s'(t) < 0 \tag{4.30}$$

for $0 < t < T$. We denote the solution to this initial value problem by $h = \Phi^s[C,K]$.

Lemma 4.2.1. *For admissible coefficients C and K, let $(p,q) = \Gamma{\cdot}\Phi^s[C,K]$. If $s(t)$ satisfies* (4.30)*, then*

$$q(t) \in \mathcal{C}[0,T) \quad q(0) = 0 \quad \textit{and} \quad q(t) < 0$$

in $0 < t < T$.

Proof. As before, the first two statements follow immediately from the solution form. For the last statement, define the new variable $u(z,t) := h(z,t) - (z - L)$. Then

$$\partial_t u = \partial_t h \quad \text{and} \quad \partial_z u = \partial_z h - 1$$

If $h = \Phi^s[C,K]$ solves the direct problem (4.29), then u satisfies the initial boundary value problem

$$\begin{aligned} C(h)\partial_t u(z,t) - \partial_z(K(h)\partial_z u(z,t)) &= 0 && \text{for } (z,t) \in Q_T \\ u(z,0) &= 0 && \text{for } 0 < z < L, \\ \partial_z u(0,t) = 0 \quad u(L,t) &= s(t) && \text{for } 0 < t < T \end{aligned} \tag{4.31}$$

Now multiply each term by an arbitrary smooth function $\phi(z,t)$ and integrate by parts to generate

$$\begin{aligned} \int_0^T \int_0^L C(h)\partial_t u \partial_z \phi \; dz \; dt =& C(\tilde{h}) \left[\int_0^L u\partial_z\phi \Big|_0^T dz - \int_0^T \int_0^L u\partial_{tz}\phi \; dz \; dt \right] \\ =& C(\tilde{h}) \left[\int_0^L u\partial_z\phi \Big|_0^T dz - \int_0^T u\partial_t\phi \Big|_0^L dt \right. \\ & \left. + \int_0^T \int_0^L \partial_z u \partial_t \phi \; dz \; dt \right], \end{aligned} \tag{4.32}$$

where $\tilde{h} = h(\tilde{z}, \tilde{t})$ for some (z, t) in the region $Q_T = (0, T) \times (0, L)$. and

$$\begin{aligned} \int_0^T \int_0^L & \partial_z (K(h) \partial_z u) \partial_z \phi \; dz \; dt \\ &= \int_0^T K(h) \partial_z u \partial_z \phi \Big|_0^L - \int_0^T \int_0^L (K(h) \partial_z u \partial_{zz} \phi \; dz \; dt \end{aligned} \tag{4.33}$$

Now let $\phi(z, t)$ solve the adjoint problem

$$\begin{aligned} C(\tilde{h}) \partial_t \phi + K(h) \partial_{zz} \phi = 0 & \qquad \text{in } Q_T, \\ \phi(z, T) = 0 & \qquad 0 < z < L, \\ \phi(0, t) = 0 \quad \partial_z \phi(L, t) = \theta(t) & \qquad \text{in } 0 < t < T, \end{aligned}$$

where $\theta(T) = 0$, but is otherwise arbitrary. Notice that the initial condition of this adjoint problem implies $\partial_z \phi(z, T) = 0$ and that the left boundary condition implies $\phi_t(0, t) = 0$. With these observations, we now combine (4.32) with (4.33), and slightly rearrange terms to form the full expression

$$\begin{aligned} \int_0^T \int_0^L & [C(\tilde{h}) \partial_t \phi + K(h) \partial_{zz} \phi] \partial_z u \; dz \; dt \qquad (4.34) \\ &= -C(\tilde{h}) \left[\int_0^L u \partial_z \phi \Big|_0^T dz - \int_0^T u \partial_t \phi \Big|_0^L dt \right] - \int_0^T K(h) \partial_z u \partial_z \phi \Big|_0^L dt. \end{aligned}$$

The homogeneous adjoint equation causes the first integrand to vanish. Similarly, since $u(z, 0) = 0$ and $\partial_z \phi(z, T) = 0$, the integral $\displaystyle\int_0^L u \partial_z \phi \Big|_0^T$ is zero as well. The side conditions $u(L, t) = s(t)$ and $\phi(0, t) = 0$ imply that the integral

$$\begin{aligned} \int_0^T u \partial_t \phi \Big|_0^L dt &= \int_0^T u \partial_t \phi \Big|^L dt \\ &= \int_0^T s(t) \theta'(t) dt \\ &= s(t) \theta(t) \Big|_0^T - \int_0^T s'(t) \theta(t) \; dt \\ &= - \int_0^T s'(t) \theta(t) \; dt, \end{aligned} \tag{4.35}$$

since $s(0) = 0$ and $\theta(T) = 0$. Now consider (4.33),

$$\begin{aligned} \int_0^T K(h)\partial_z u \partial_z \phi \Big|_0^L \, dt &= \int_0^T K(h)\partial_z u \partial_z \phi \Big|^L \, dt \\ &= \int_0^T K(h)(\partial_z h - 1)\phi \Big|^L \, dt \\ &= \int_0^T q(t)\theta(t) \, dt. \end{aligned} \tag{4.36}$$

Substituting (4.35) and (4.36) into (4.34) yields the simple integral relationship

$$C(\tilde{h}) \int_0^T s'(t)\theta(t) \, dt = \int_0^T q(t)\theta(t) \, dt \tag{4.37}$$

Recall that the suction $s(t)$ satisfies $s'(t) < 0$, which implies that a choice of positive $\theta(t)$ will make the left side of (4.37) strictly negative. Since θ is otherwise arbitrary, $q(t)$ must be strictly negative. □

The proof of the next lemma is very similar to the Phase 1 case, and it omitted.

Lemma 4.2.2. *For admissible coefficients C and K, let $h = \Phi^s[C, K]$. If $s(t)$ satisfies 4.30, then $\partial_z h(L, t) - 1 < 0$ almost everywhere in Q_T.*

Lemma 4.2.3. *For admissible coefficients C and K, let $h = \Phi^s[C, K]$. If $s(t)$ satisfies 4.30, then $\partial_t h(z, t) < 0$ almost everywhere in Q_T.*

Proof. Proceed as in lemma 4.1.3 to reach

$$\begin{aligned} \iint_{Q_T} \big[C(h)\partial_t \phi + K(h)\partial_{zz}\phi + K'(h)\partial_z \phi\big]\partial_t h \, dz \, dt \\ = \int_0^T K(h)(\partial_z h - 1)\partial_t \phi \Big|_0^L \, dt \quad &(4.38) \\ - \int_0^L K(h)(\partial_z h - 1)\partial_z \phi \Big|_0^T \, dz \quad &(4.39) \\ + \int_0^T K(h)\partial_t h \partial_z \phi \Big|_0^L \, dz. \quad &(4.40) \end{aligned}$$

Since $K(h(L,t))(\partial_z h(L,t)-1) = q(t)$ and $\partial_z h(0,t)-1 = 0$, (4.38) becomes

$$\int_0^T K(h)(\partial_z h - 1)\partial_t \phi\Big|_0^L \, dt = \int_0^T q(t)\partial_t \phi(L,t) \, dt.$$

Taking $\phi(z,T) = 0$ implies that $\partial_z \phi(z,T) = 0$. The initial condition $h(z,0) = z - L$ eliminates the term (4.39). Finally, if we set both $\partial_z \phi(0,t) = 0$ and $\partial_z \phi(L,t) = 0$, then (4.40) will vanish. So, if we allow $\phi(z,t)$ to satisfy the adjoint problem

$$\begin{aligned} C(h)\partial_t \phi + K(h)\partial_{zz}\phi + K'(h)\partial_z \phi = F(z,t) & \qquad \text{in } Q_T, \\ \phi(z,T) = 0, & \qquad 0 < z < L, \\ \partial_z \phi(0,t) = 0, \quad \partial_z \phi(L,t) = 0 & \qquad 0 < t < T, \end{aligned}$$

then we may write

$$\iint_{Q_T} \partial_t h F(z,t) \, dz \, dt = \int_0^t q(t)\partial_t \phi(L,t)$$

This can be rewritten as

$$\begin{aligned} \iint_{Q_T} \partial_t h F(z,t) \, dz \, dt &= q(t^*) \int_0^T \partial_t \phi(L,t) dt \\ &= \quad q(t^*)\phi(L,t)\Big|_0^T \\ &= -\, q(t^*)\phi(L,0), \end{aligned}$$

noting the initial condition. Restricting F to be nonnegative but otherwise arbitrary, the maximum principle will ensure that $\phi(z,0)$ will be negative almost everywhere. It has been shown that $q(t) < 0$, for $0 < t < T$, implying that the right side is negative. Evidently, since F is nonnegative, $\partial_t h$ must be negative almost everywhere in Q_T. □

Lemma 4.2.4. *For admissible coefficients C and K, let $h = \Phi^s[C, K]$ and $(p, q) = \Gamma \cdot \Phi^s[C, K]$. If $s(t)$ satisfies 4.30, then for each τ, $0 < \tau < t$, h satisfies*

$$p(\tau) + \frac{z}{L}\left(s(\tau) - p(\tau)\right) < h(z, \tau) < z - L$$

Proof. As in 4.2.1, we recast the equation (4.29) using $u(z, t) := h(z, t) - (z - L)$. As before, this leads to

$$\begin{aligned} C(h)\partial_t u(z, t) - \partial_z(K(h)(\partial_z u(z, t)) = 0 \qquad & \text{for } (z, t) \in Q_T \\ u(z, 0) = 0 \qquad & \text{for } 0 < z < L, \\ \partial_z u(0, t) = 0 \quad u(L, t) = s(t) \qquad & \text{for } 0 < t < T \end{aligned} \tag{4.41}$$

Appealing to the maximum-minimum principle, the function $u(z, t)$ must attain its both its maximum and minimum on the parabolic boundary. The maximum principle allow us to deduce that this maximum must be zero. Since $s(t)$ is a decreasing function and $\partial_z u(0, t) = 0$, the minimum must occur on the right boundary, where $u(L, t) = s(t)$. Since $\partial_z h(z, t) < 1$ almost everywhere, it follows that $\partial_z u(z, t) < 0$. Therefore, $u(z, t)$ is a convex function. Since $u(0, t) = p(t)$ and $u(L, t) = s(t)$, $u(z, \tau)$ satisfies

$$u(0, \tau) + \frac{z}{L}(u(L, \tau) - u(0, t)) < u(z, \tau) < 0.$$

This is simply the statement that the solution profile lies above the line segment connecting the value of u on the right boundary to the value of u on the left boundary, and below the constant zero function. Recasting in

h, and noting that $p(\tau) = u(0, \tau) - L$ and $s(\tau) = u(L, \tau)$, this becomes

$$\begin{aligned} u(0,\tau) + (z-L) + \frac{z}{L}(u(L,\tau) - u(0,t)) \ &< \ u(z,\tau) + (z-L) \ < \ (z-L) \\ (u(0,\tau) - L) + z + \frac{z}{L}(u(L,\tau) - (u(0,\tau) - L) - L) \ &< \ h(z,\tau) < \ (z-L) \\ p(\tau) + z - z + \frac{z}{L}(s(\tau) - p(\tau)) \ &< \ h(z,\tau) < \ (z-L) \\ p(\tau) + \frac{z}{L}(s(\tau) - p(\tau)) \ &< \ h(z,\tau) < \ (z-L), \end{aligned}$$

which holds for $h(z,\tau)$ for almost every z in $[0, L]$. The monotonicity of $s(t)$ from (4.30) (and consequently $p(t)$) allows this to hold for all $t, 0 < t < \tau$. □

As in the Phase 1 setting, we can show,

Theorem 4.2.1. *For admissible coefficients $C(h)$ and $K(h)$, let $h_i(z,t) = \Phi^s[C_i, K_i]$ and $(p_i, h_i) = \Gamma \cdot \Phi^s[C_i, K_i]$ for $i = 1, 2$. For arbitrary smooth functions $P^*(t), Q^*(t)$, let $\phi = \Phi^*[P^*, 0]$ and $\psi = \Phi^*[0, Q^*]$. Changes in the inputs $\Delta C = C_1 - C_2$ and $\Delta K = K_1 - K_2$ are related to changes in the output $\Delta p = p_1 - p_2$ and $\Delta q = q_1 - q_2$. For any τ, $0 < \tau < T$, this relationship is*

$$\int_0^T \int_0^L \{\Delta K(h_2)(\partial_z h_2(z,t) - 1)\partial_z \phi + \Delta C(h_2)\phi \partial_t h_2\} \ dz \ dt = \int_0^T \Delta p \ P^*(t) \ dt \quad (4.42)$$

and

$$\int_0^T \int_0^L \{\Delta K(h_2)(\partial_z h_2(z,t) - 1)\partial_z \psi + \Delta C(h_2)\psi \partial_t h_2\} \ dz \ dt = \int_0^T \Delta q \ Q^*(t) \ dt \quad (4.43)$$

The identifiability of C and K quickly follow, with the result

Lemma 4.2.5. *For admissible coefficients C_i and K_i, let $(p_i, q_i) = \Gamma \cdot \Phi^s[C_i, K_i]$. If $\Delta p = p_1 - p_2$ and $\Delta q = q_1 - q_2$ are both identically zero, then $\Delta C = C_1 - C_2$ and $\Delta K = K_1 - K_2$ are also both identically zero.*

The last two proofs are omitted, as they are nearly identical to the arguments of theorem 4.1.1 and lemma 4.1.5.

Chapter 5

TWO PARAMETER NUMERICAL EXPERIMENTS

In this section, we present a numerical implementation of the recovery algorithm and analyze this process via a series of numerical experiments. We consider several numerical experiments designed to gain insight into the recovery of the two unknown coefficients $C(h)$ and $K(h)$ in the two parameter quasilinear conduction diffusion equation given by

$$C(h(z,t))\partial_t h(z,t) - \partial_z (K(h(z,t))(\partial_z h(z,t) - 1)) = 0. \tag{5.1}$$

In porous media applications, (5.1) is referred to as the Richards Equation, and is widely used to model fluid flow in porous media. Meaningful solutions require an accurate description of soil characteristics, reflected in the coefficients $C(h)$ and $K(h)$. The values $C(h)$ and $K(h)$ must be experimentally determined, in either a direct or indirect manner. Here we focus on the indirect simultaneous determination of these parameters via an algorithm based on the integral identities developed in the preceding chapter. It is hoped that this approach will provide a more complete understanding of the identification process. The integral method can be used independently in coefficient recovery, or viewed as a tool to examine cases where identification fails.

The method presented here is based on the integral identities

$$\begin{aligned}\int_0^\tau P^*(t)[p(t)-p_2(t)]\,dt = &\int_0^\tau\int_0^L (K(h_2)-K_2(h_2))\partial_z\phi(\partial_z h_2 - 1)\ dz\ dt\\ &+\int_0^\tau\int_0^L (C(h_2)-C_2(h_2))\partial_t h_2\phi\ dz\ dt \qquad (5.2)\end{aligned}$$

which we refer to as the p-integral identity, and

$$\begin{aligned}\int_0^\tau Q^*(t)[q(t)-q_2(t)]\,dt = &\int_0^\tau\int_0^L (K(h_2)-K_2(h_2))\partial_z\psi(\partial_z h_2 - 1)\ dz\ dt\\ &+\int_0^\tau\int_0^L (C(h_2)-C_2(h_2))\partial_t h_2\psi\ dz\ dt \qquad (5.3)\end{aligned}$$

which we call the q-integral identity. The general identities on which these are based was developed in the previous chapter, equation (4.13). We again note that (4.13) is only exact if we evaluate C^*, K^* and K'^* in the adjoint problems at possibly different indeterminate values of h. In the usual fashion, we instead solve an approximate adjoint problem, and use these approximate values in place of their exact representations in the p and q integral identities. The error of this approximation approaches zero as the numerical solution h_2 approaches the true solution h.

The algorithm presented in this chapter seeks to create linear polygonal approximations to the unknown coefficients $C(h)$ and $K(h)$ utilizing observations made of the system. The measurements are considered to be taken on the boundary of the media, $z = 0$ and $z = L$, since placing measurement devices in the interior region might be difficult or impossible. While not the only observable boundary measurements, the state $p(t) = h(0, t)$ and the flux $q(t) = K(h(L, t))(\partial_z h(L, t) - 1)$ are both easily obtained. In chapter 4, the map $\Phi : [C, K] \to (p, q)$ has been shown to be continuous and invertible under the monotone forcing (ie drainage and/or suction) constraint via the integral identities.

We now begin with a description of the numerical details.

5.1 Numerical methodology

The nonlinear PDE 5.1 was discretized on a non-uniform space grid. The resulting system of ODEs was then submitted to a implicit time integration scheme.

The standard finite difference scheme was used. Using a space discretization over the grid $\{0 = x_1, x_2, ..., x_{n-1}, x_n = L\}$ and the convention that $h(x_i, t) = h_i^t$, the scheme can be written

$$C(h_i^t)\ \dot{h}_i^t = \frac{\Big(K(h_{i+1/2}^t)\left(h_{i+1}^t - h_i^t - \Delta x\right)\Big) - \Big(K(h_{i-1/2}^t)\left(h_i^t - h_{i-1}^t\right) - \Delta x\Big)}{(\Delta x)^2},$$

although here it was implemented for use on a possibly non uniform grid, and was written

$$C(h_i^t)\ \dot{h}_i^t = \frac{\Big(K(h_{i+1/2}^t)\left(\frac{h_{i+1}^t - h_i^t}{\Delta x_i} - 1\right)\Big) - \Big(K(h_{i-1/2}^t)\left(\frac{h_i^t - h_{i-1}^t}{\Delta x_{i-1}} - 1\right)\Big)}{\Delta x_{i-1/2}}.$$

5.2 Recovery Algorithm

We consider the inverse problem in which the two coefficients $C = C(h)$ and $K = K(h)$ are to be identified from data that is assumed to be recorded at the fixed time nodes $0 = t_0 < t_1 < \ldots < t_N = T$ in the interval $[0, T]$:

$$p(t_k) = h(0, t_k) \text{ and } q(t_k) = K(h(L, t_k))(\partial_z u(L, t) - 1).$$

This data will be referred to as the p and q data, respectively. We will use both pieces of data to construct a piecewise linear continuous approximation to the unknown coefficients C and K. This data, which we further denote $p_k = p(t_k)$ and $q_k = q(t_k)$ for $k = 0, 1, ..., N$, partitions the interval $I = [0, T]$ into what we will term the *inner mesh*. To parameterize the coefficient space, we first define $\mu_i = \min(h(x, t_i))$, where

$\mu_0 > \mu_1 > \ldots > \mu_M$. Recall that the pressure head $h(x,t)$ has been shown to be monotone decreasing under a gravity drainage restriction, thus making this parameterization possible. We can now define an associated *outer mesh* $J = [\mu_0, \mu_1, , \ldots, \mu_M]$, which partitions the domain of the coefficients of C and K in the drainage experiment. The outer mesh determines the degrees of freedom of the recovered parameters C and K.

The integral identity requires both the integration of $g(t)$ and $h(t)$ for t in $[0,T]$ and $h(x,t)$ in $[0,L] \times [0,T]$. The use of numerical integration methods require that the inner mesh, on which the observed data are represented, be sufficiently finer than the outer mesh. This limits the ability to arbitrarily refine the outer mesh in order to improve accuracy of identification.

We consider the family of polygonal functions, $\hat{C}$ and $\hat{K}$. Define as in the one parameter problem the basis functions $\{\lambda_i\}_0^M$, given by

$$\lambda_i(u) = \begin{cases} 0 & \text{if } u < \mu_{i-1} \\ \dfrac{u - \mu_{i-1}}{\mu_i - \mu_{i-1}} & \text{if } \mu_{i-1} \geq u \geq \mu_i \\ 1 & \text{otherwise.} \end{cases} \tag{5.4}$$

We can now define

$$\hat{C}(h) = \sum_{i=0}^{M} c_i \lambda_i(h) \qquad \hat{K}(h) = \sum_{i=0}^{M} k_i \lambda_i(h). \tag{5.5}$$

We introduce several notations:

- $\hat{C}(u) = P_M[c_0, c_1, \ldots, c_M]$ denotes the polygonal coefficient given by (5.5) based on nodal values $[c_0, c_1, \ldots, c_M]$.

- $\hat{K}(u) = P_M[k_0, k_1, \ldots, k_M]$ denotes the polygonal coefficient given by (5.5) based on nodal values $[k_0, k_1, \ldots, k_M]$.

- $h(x,t;C,K)$ denotes the solution of the direct problem (2.1) with coefficient C and K.

- $\phi(x,t;C,K,P^*,0)$ denotes the solution of the adjoint problem (4.12) with coefficients $C \overset{def}{=} C(\mu(x,t)), \quad K \overset{def}{=} K(\mu(x,t))$ and boundary data $(P^*,0)$ where $K(0,t)\partial_z\phi(0,t) = P^*(t)$ and $\phi(L,t) = 0$.

- $\psi(x,t;C,K,0,Q^*)$ denotes the solution of the adjoint problem (4.12) with coefficients $C \overset{def}{=} C(\mu(x,t)), \quad K \overset{def}{=} K(\mu(x,t))$ and boundary data $(0,Q^*)$ where $K^*(0,t)\partial_z\psi(0,t) = 0$ and $\psi(L,t) = Q^*(t)$.

Assume now that the data pair $(p(t),q(t))$ are produced by an unknown coefficient pair (C,K). Fix an outer partition, calling it $\Pi = \{0 = \mu_0 < mu_1 < \ldots < \mu_M\}$ of J. We will now define the polygonal coefficients C and K using a recursive algorithm which utilizes the observed (p,q) data pair as follows:

1. c_0 and k_0 are assumed to be given.

2. for $i = 1, 2, \ldots, t_M$

   (a) Compute integrals

$$M_{11} = \int_{t_{i-1}}^{t_i} \int_0^1 \lambda_k(h_2)(\partial_z h_2 - 1)\partial_z\phi \; dt \; dx$$
$$M_{21} = \int_{t_{i-1}}^{t_i} \int_0^1 \lambda_k(h_2)(\partial_z h_2 - 1)\partial_z\psi \; dt \; dx$$
$$M_{12} = \int_{t_{i-1}}^{t_i} \int_0^1 \lambda_k(h_2)(\partial_t h_2)\phi \; dt \; dx$$
$$M_{22} = \int_{t_{i-1}}^{t_i} \int_0^1 \lambda_k(h_2)(\partial_t h_2)\psi \; dt \; dx$$

(b) and

$$b_1 = \int_{t_{i-1}}^{t_i} (p(t) - p_2(t))P^*(t)dt \tag{5.6}$$

$$b_2 = \int_{t_{i-1}}^{t_i} (q(t) - q_2(t))Q^*(t)dt \tag{5.7}$$

(c) Solve the linear system

$$\begin{bmatrix} k_i \\ c_i \end{bmatrix} = \begin{bmatrix} k_{i-1} \\ c_{i-1} \end{bmatrix} + \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{5.8}$$

where

$$\begin{aligned} C_1(h) &= P_M[c_0, c_1, \ldots, c_{j-1}, c_j], \\ C_2(h) &= P_M[c_0, c_1, \ldots, c_{j-1}, c_{j-1}], \\ K_1(h) &= P_M[k_0, k_1, \ldots, k_{j-1}, k_j], \\ K_2(h) &= P_M[k_0, k_1, \ldots, k_{j-1}, k_{j-1}], \\ h_2(x,t) &= h_2(x,t; C_2, K_2), \\ p_2(t) &= h_2(0,t) \\ q_2(t) &= K_2(h(L,t))(\partial_z h_2(L,t) - 1) \\ \phi(x,t) &= \phi(x,t; C_1, K_1) \end{aligned}$$

In this way, n pairs of of nodal values can be generated. Note that the nodal values $[c_0, c_1, \ldots, c_{j-1}, c_j]$ are not actual coefficient values. Instead, the cumulative sum of these values represents the coefficient value.

Experimentally, however, the flux is often reported in an integrated form, which we call the cumulative flux. This smoothes the often noisy flux data measurements. While it is possible to numerically differentiate this cumulative flux data, an alternative expression that directly allows cumulative flux is quite easy to develop. Once again we turn to integration by parts. Consider the general b_2 term in the identity, given by

$$b_2 = \int_0^T (q(t) - q_2(t))Q^*(t)dt$$

Integrating this by parts yields the formal expression

$$= (q(t) - q_2(t))Q^*(t)\Big|_0^T + \int_0^T \left\{ \int_0^t (q(s) - q_2(s))\ ds \right\} Q^{'*}(t)dt.$$

Note that $Q(0) - Q_2(0) = 0$ and that $Q^*(T) = 0$. Denoting the cumulative quantities as $Q(t) := \int_0^t q(s)dt$ and $Q_2(t) := \int_0^t q_2(s)dt$, we write

$$b_2 = \int_0^T (Q(t) - Q_2(t))Q^{'*}(t)\ dt. \tag{5.9}$$

Evidently, if $Q^*(t)$ is chosen to be differentiable, we are then able to directly apply the cumulative flux measurement in the integral identity. We note that this method may be employed any number of times, limited only by the smoothness of the dual data. This suggests a simple technique to filter noisy data. This formulation only requires that we solve the q-adjoint problem numerically for a time dependent, and suitably smooth, data function $Q^*(t)$. An alternative expression for b_1 involving the integral of the state data $p(t)$ follows similarly.

5.3 Numerical Code

As in the one parameter recovery algorithm, the code was constructed in several parts. The first, the direct algorithm, generates boundary data $h(0,t) = p(t)$ and $K(h(L,t))(\partial_z h(L,t) - 1) = q(t)$ by first computing a numerical solution of the direct problem. The second part of the code was the development of the dual algorithm, used to produce a numerical approximation to the adjoint problem. The final portion was the recovery algorithm, which assembled the components generated by the direct and adjoint algorithms, and used the integral identities to produce an approximate coefficient pair.

All numerical methods were coded in the `Matlab 6.1` environment. In particular, the ability of several of the solvers in the ODE suite to solve problems containing a mass matrix was of great use. As in the one parameter implementation, the return of a solution structure was beneficial in several numerical experiments.

The PDE was discretized using Finite Difference (FD) methods in space to produce a system of ODEs. Finite Element methods (FEM) could also have been used but were not. Boundary Value methods (BVM) could also have been implemented, although experiments in the one parameter case indicated that this would prove computationally inefficient.

5.3.1 Direct algorithm implementation

In this section we discuss the code used to generate a numerical solution to the direct problem. Subsequent work required that this code be fairly efficient and flexible. The nonlinear terms of the equation were managed at each time level as a linear interpolation of a passed call-out table. The resulting system of ODEs was submitted to Matlab's time integration methods, as chosen by the user. We now present a template for the code used to generate the so called direct solution.

5.3.2 Direct Problem

The boundary conditions are passed as call-out tables, which are evaluated with a linear interpolation via Matlab's `interp1` command at each time step. The boundary conditions are once again handled via ghost nodes.

Notice that this code does not use the value of C. It was certainly possible to include the C value in the inputs, and then divide by this quantity in the final line of the code. In practice, however, the magnitude of C is small. In several preliminary experiments, the direct implementation of C occasionally led to difficulty. As a remedy, C was considered to be a mass matrix, which allowed the solver to manage this term.

5.4 Phase 2

Phase 1 experiment allowed exploration of the coefficients C and K in the parameter range from $h \in (0, L]$. We now turn to coefficient recovery in the Phase 2 experiment, in which a much larger parameter range may be visited. Recall that in this situation, the initial condition is given by $h(z, 0) = z - L$, which is the equilibrium solution to the Phase 1 problem. Imposing no flow conditions at the top of the column ($z = 0$) and applying suction at the base of the column ($z = L$), the capillary pressure head $h(z, t)$ satisfies

$$\begin{aligned} C(h)\partial_t h(z,t) &= \partial_z(K(h)(\partial_z h(z,t) - 1)) && \text{for } (z,t) \in Q_T \\ h(z,0) &= z - L && \text{for } 0 < z < L, \\ \partial_z h(0,t) - 1 = 0 \quad h(L,t) &= s(t) && \text{for } 0 < t < T \end{aligned}$$

where $Q_T = \{(z,t) : 0 < z < L, 0 < t < T\}$, and the suction function $s(t)$ is smooth and monotone in time.

5.5 Comparison of Phase 1 and Phase 2 Experiments

Mathematically simpler than the Phase 2 suction experiment, the Phase 1 drainage experiment provides some valuable insight into the physical process. While much time was spent considering this case, only a brief synopsis is presented. The Phase 2 recovery provide a much richer basis for discussion than do the experimental results of the Phase 1 experiments. The experimental device, when limited to a simple drainage experiment, is able to yield data probing only pressure heads ranging from 0 to the depth of the soil column. In practice this is approximately 3 to 5 cm. For simplicity, pressure head will be measured in centimeters of water. In comparison, the suction experiment is constrained only by the working limitations of the laboratory apparatus, thereby admitting a parameter range of over 200 cm.

5.6 Experiments utilizing only the forward solution

We begin with a discussion of coefficient recovery in the Phase 1 experiment. Recall that this is the situation is which a completely saturated vertical soil column is allowed to drain under gravity. Imposing no flow conditions at the top of the column ($z = 0$) and zero head at the base of the column ($z = L$), the capillary pressure head $h(z,t)$ satisfies

$$\begin{aligned} C(h)\partial_t h(z,t) &= \partial_z(K(h)(\partial_z h(z,t) - 1)) && \text{for } (z,t) \in Q_T \\ h(z,0) &= 0 && \text{for } 0 < z < L, \\ \partial_z h(0,t) - 1 = 0 \quad & h(L,t) = 0 && \text{for } 0 < t < T \end{aligned}$$

where $Q_T = \{(z,t) : 0 < z < L, 0 < t < T\}$.

We begin by presenting experiments in which the coefficients are taken from the families

$$\begin{aligned} C(h,\alpha) &= h\ (1-h)^\alpha\ ||h(1-h)^\alpha||_\infty^{-1} + 1/2 \\ K(h,\beta) &= (1+\beta h)\ \mathcal{H}(h + 1/\beta) + 1/2, \end{aligned}$$

where α ranges between 1/3 and 3 and β takes values from 0 to 8. Also, $\mathcal{H}(h + 1/\beta)$ is the Heaviside function centered at $-1/\beta$. In application, the capacity and conductivity coefficients will be qualitatively similar to elements in these families. Individual coefficients from these families were used to generate $p(t)$ and $q(t)$ data over the time span from $t = 0$ to $t = 3$ with 75 uniform nodes.

While mathematically interesting, the drainage experiment was viewed as a preliminary stage of coefficient recovery in the Phase 2 experiment. As such, we present two coefficient experiments simulated in a unit long column, but selected the scale of the coefficients to probe behavior consistent with the Phase 2 experiment. Although physical arguments indicate that the true conductivity coefficient K should be monotone decreasing as h decreases, we have not limited the discussion to this case.

5.6.1 Allowing C(h) to vary

Here we fix $\beta = 2$ and let α take values from $\{1/3, 1, 3\}$, and plot over the time range from $t = 0$ to $t = 3$. In the plot of p and q data, the triangles indicate every 5^{th} time observation. Similarly, the crosses in the plots of the coefficient represent the state of the system $h(t)$ when observed on the same time nodes as the data plots above them. In this way we have an indication of both the speed of the process and the initial value the coefficients. We first observe that time at which p and q are coincident occurs earlier as the maximum of C moves closer to zero. Notice also that the crosses occur more rapidly in h space as α increases. While the time scales differ, the data appears qualitatively similar. The same sharp drop in the P data occurs as the does the apparent change in curvature of the q line plot. In addition, if we call this crossing time t_c, then value at the crossing time $p(t_c) = q(t_c)$ seems remarkably consistent. Table 5.1 makes this more apparent. The response in the data is encouraging. This indicates that

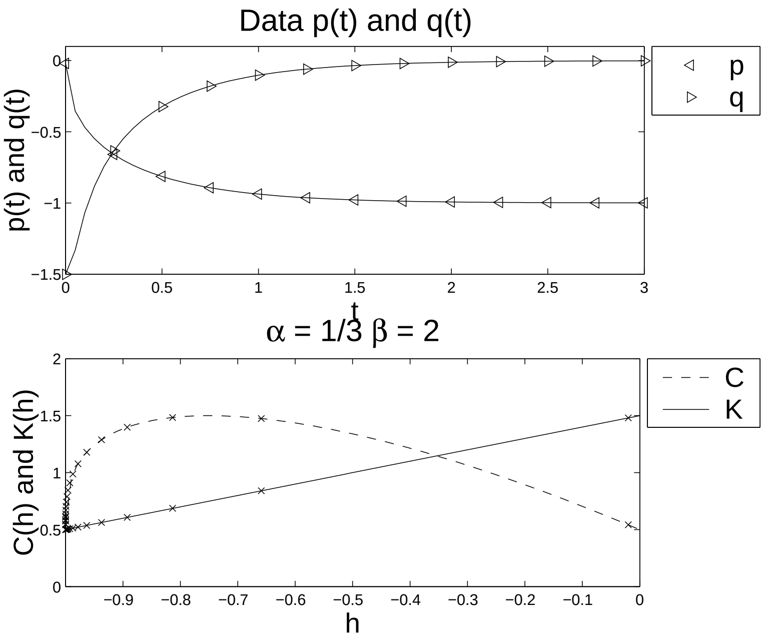


Figure 5.1: Data and coefficients with $\alpha = 1/3$ and $\beta = 2$

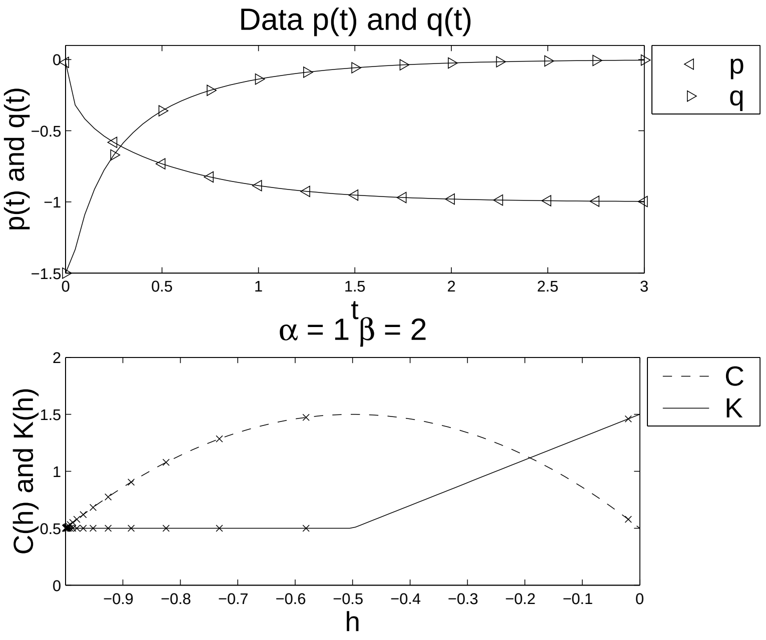


Figure 5.2: Data and coefficients with $\alpha = 1$ and $\beta = 2$

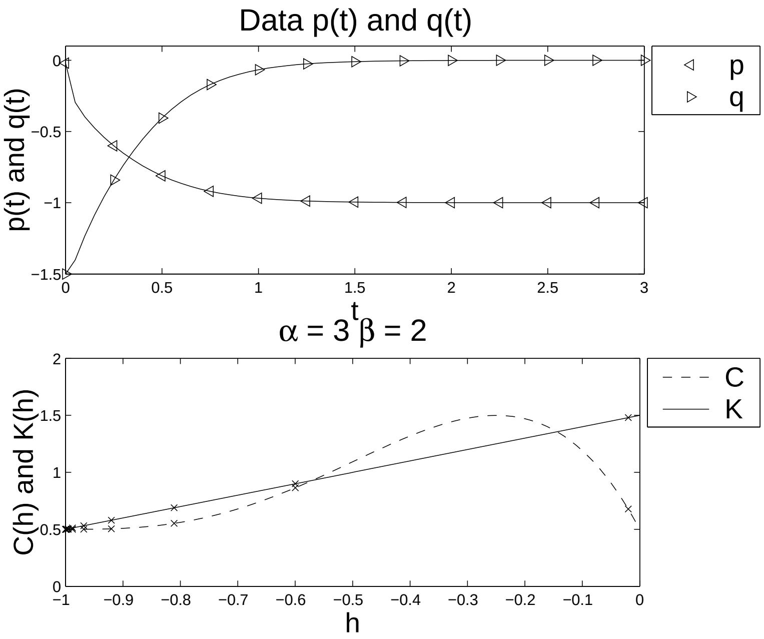


Figure 5.3: Data and coefficients with $\alpha = 3$ and $\beta = 2$

Parameter values	Crossing time	Crossing value	final p	final q
$\alpha = 1/3,\ \beta = 2$	0.2450	-0.6545	-0.9991	-0.0014
$\alpha = 1/2,\ \beta = 2$	0.2600	-0.6534	-0.9991	-0.0014
$\alpha = 1,\ \beta = 2$	0.2950	-0.6571	-0.9995	-0.0008
$\alpha = 2,\ \beta = 2$	0.3250	-0.6679	-0.9999	-0.0002
$\alpha = 3,\ \beta = 2$	0.3300	-0.6809	-1.0000	-0.0000
$\alpha = 1,\ \beta = 0$	0.3500	-0.4170	-0.9553	-0.0365
$\alpha = 1,\ \beta = 1$	0.2950	-0.6571	-0.9995	-0.0008
$\alpha = 1,\ \beta = 2$	0.2900	-0.6116	-0.9967	-0.0040
$\alpha = 1,\ \beta = 3$	0.2700	-0.5606	-0.9917	-0.0088
$\alpha = 1,\ \beta = 4$	0.2500	-0.5216	-0.9870	-0.0130
$\alpha = 1,\ \beta = 5$	0.2350	-0.4935	-0.9832	-0.0162

Table 5.1: Data crossing times and values

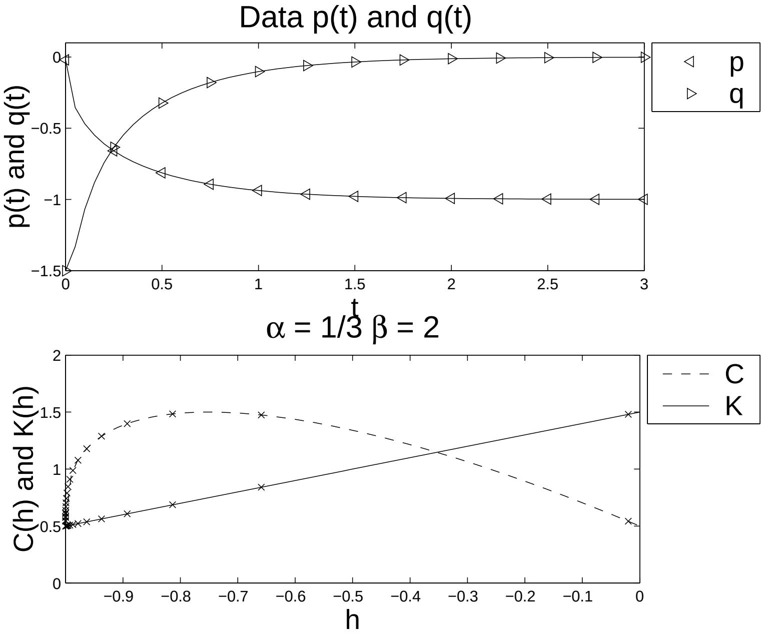


Figure 5.4: Data and coefficients with $\alpha = 1$ and $\beta = 1$

both p and q respond to the variation in the C coefficient. The crossing time seems to be similar to the breakthrough time of the one parameter experiments.

5.6.2 Allowing K(h) to vary

Here we fix $\alpha = 1$ and let β take values from 1 to 5 and plot over the time range from $t = 0$ to $t = 3$. The initial slope of the coefficient K is given by the parameter β. In practice, this coefficient is often represented as a function with rapid initial decrease. Here we attempt to gain some intuition about the effect that this has on the data p and q. As in the plots in the last section, the triangles indicate every 5^{th} time observation. Similarly, the crosses in the plots of the coefficient represent the state of the system $h(t)$ when observed on the same time nodes as the data plots above them.

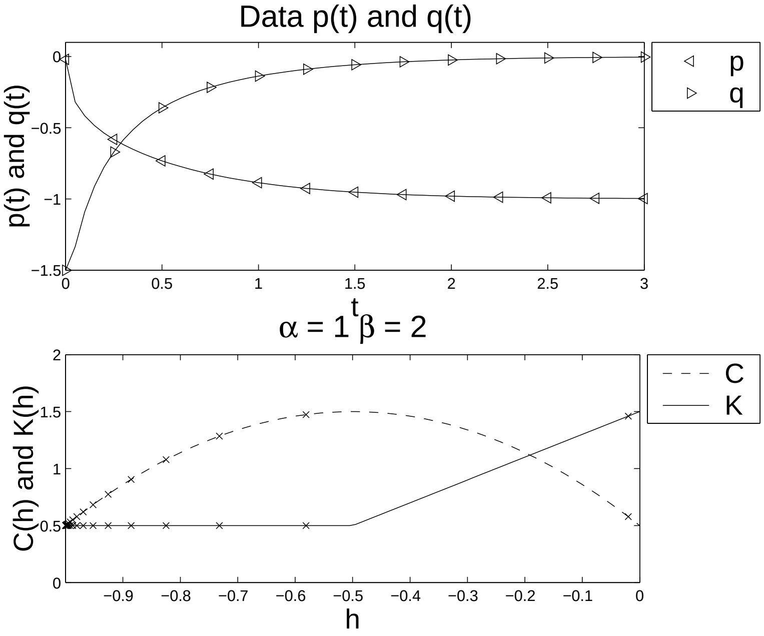


Figure 5.5: Data and coefficients with $\alpha = 1$ and $\beta = 2$

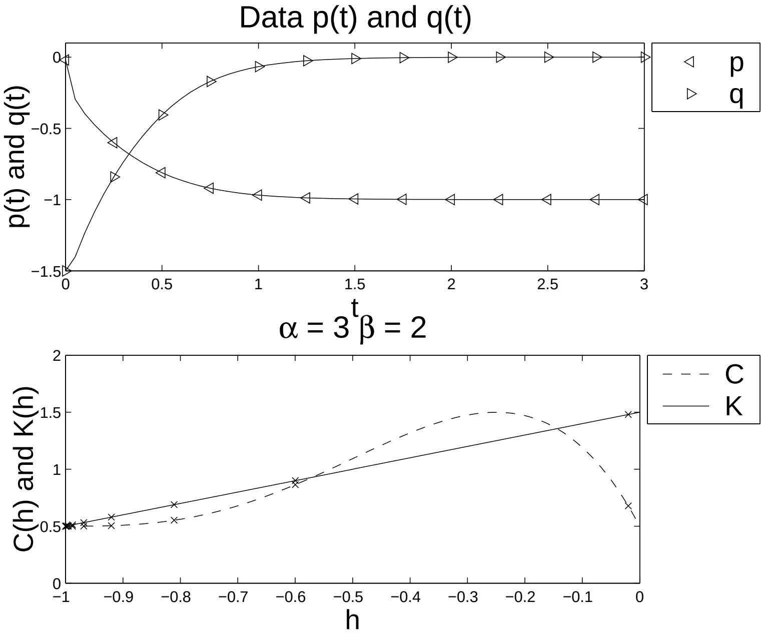


Figure 5.6: Data and coefficients with $\alpha = 1$ and $\beta = 4$

We first observe that time at which p and q cross again diminishes as the effective region of the coefficient is grows as evidenced in the plots and the table 5.1. We think of the effective coefficient as an average value of the coefficient in the time period of interest. Notice also that the $q(t)$ appears relatively similar in both 5.4 and 5.5, but shows large response in figure 5.6. Table 5.1 makes this more apparent.

5.7 Experiments requiring full Recovery algorithm

The integral identity allows exploration of

- Iteration
- Dimension of Nodal Basis
- Boundary Condition selection
- Scaling of inversion
- Selection of Dual data
- Dependence of solution on time
- Dimensional aspects of the coefficients $C(h)$
- Noisy Data

Before we begin the discussion of error we describe the error measures that were used. Many of the following plots are of a relative error indicator. This was computed via the formula

$$\left| \frac{\displaystyle\int_0^t \Delta p(t) P^*(t)\ dt}{\displaystyle\int_0^t p(t)\ dt} \right| + \left| \frac{\displaystyle\int_0^t \Delta q(t) Q^*(t)\ dt}{\displaystyle\int_0^t q(t)\ dt} \right| .$$

While not a true norm, this indicator proved to be useful as an tolerance measure for iteration. The ability of error to cancel in individual time strips allowed iteration to proceed efficiently. Since the integral identity allows much of the observation error to cancel per strip, the error indicator was formulated to reflect this. If the standard L^1 or L^2 measures had been used, the error introduced numerically and or experimental would have been more noticeable.

Recall that the one parameter error was typically the L^2 norm of the coefficient error. A similar form could have been implemented in the two parameter experiments in cases where the true coefficient is known. However, the initial goal was recovery of unknown coefficients. To preserve continuity in all cases presented here, an alternative error formulation involving only observable quantities was used. A relative error indicator was used in recognition that the true physical process is quite likely to contain a certain level of uniform noise, and that this should become less influential as the true signal magnitude increased. These heuristic arguments might be made concrete by one wishing to explore the numerical analysis of the recovery process, which we do not attempt here.

We begin the discussion with iteration. These experiments are fundamental to achieve reasonable recovery via the integral identity method.

5.7.1 Iteration

In this section, we consider the effect of iteration in the recovery algorithm. By iteration, we mean the repeated application of the integral identity on a single time strip and utilizing the previous pair of coefficient approximations. Initial experiments made clear that while the first coefficient estimates are fairly accurate, a small number of iterations greatly improves the recovery. We present a series of numerical experiments based on the Phase 1 problem with basic dynamics in the coefficients C and K. Restriction to the drainage situation allows the time scale of the process is more easily understood than the Phase 2 type experiment, in which the suction may alter the relative time scales. We take

$$C(h) = \gamma_C \pm m_C h \quad \text{and} \quad K(h) = \gamma_K \pm m_K h$$

To limit the difficulty in recovery, we generate data using these coefficients on 1001 equally space time nodes. We then choose a single node in coefficient space, based on the maximum state at time $t = 1$ in the coefficient space, and perform recovery.

We provide plots of two such runs. In the first, (5.7), we attempt to recover the coefficient pair

$$C(h) = 1 - (1/2)\ h \quad \text{and} \quad K(h) = 2 + (1/2)h$$

while in the second, we attempt recovery of

$$C(h) = 2 + (1/2)\ h \quad \text{and} \quad K(h) = 1 - (1/2)h.$$

We plotted the sequence of iterates of the approximate coefficient. In both cases, the method converges very quickly. There is, however, a noticeable difference in the initial estimate. Initially, the method has more difficulty in computing an accurate update in first of the two plots (5.7), but after a single iteration, the approximation and the true coefficient are virtually indistinguishable. In the second plot (5.8), the initial estimates are quite good, however the iteration appears to converge more slowly. It is possible that the shorter state interval used in the second recovery might be contributing to the visibly less accurate initial approximation of C. We note that both recovery utilized the same number of observations. In subsequent experiments, we use iteration unless otherwise noted.

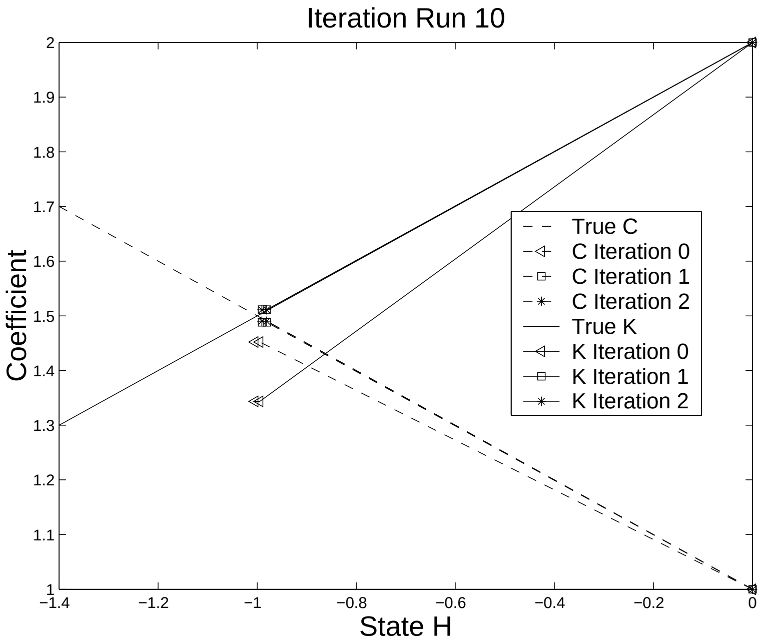


Figure 5.7: Iteration, $C(h) = 1 - (1/2)h$ and $K(h) = 2 + (1/2)h$

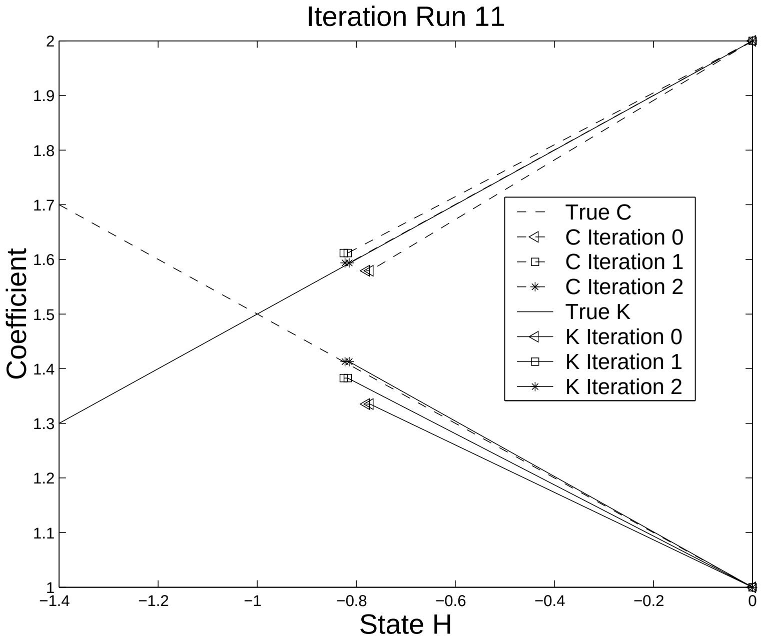


Figure 5.8: Iteration, $C(h) = 2 + (1/2)h$ and $K(h) = 1 - (1/2)h$

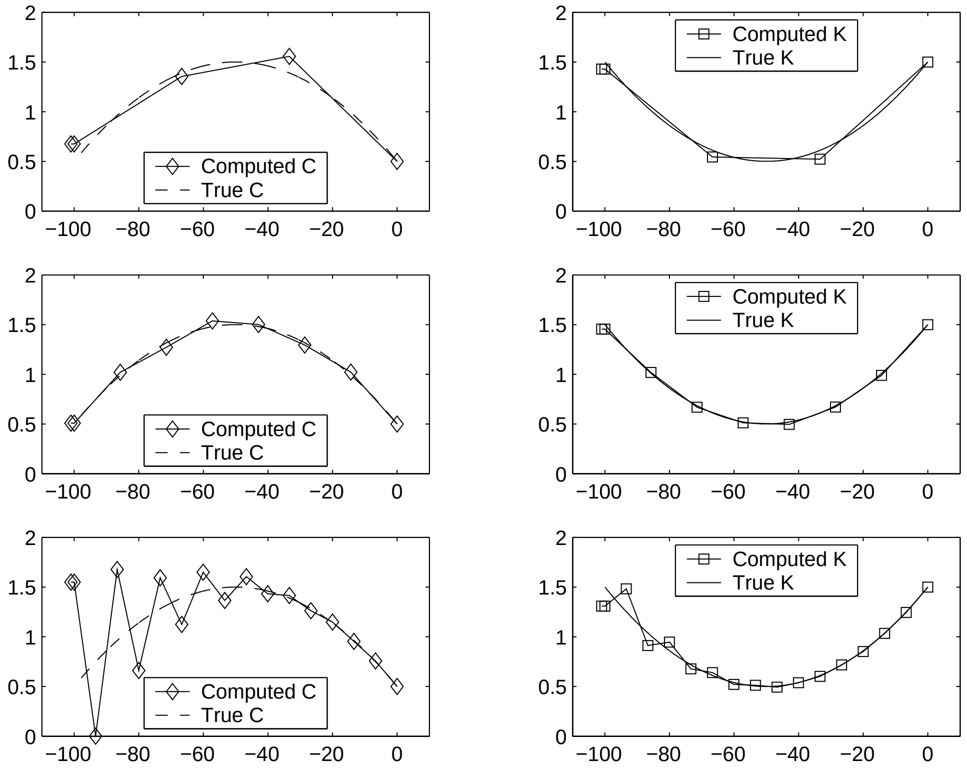


Figure 5.9: Error for various uniform nodal bases

5.7.2 Dimension of Nodal Basis

The dimension of the nodal basis plays a role in accuracy of recovery. Here we consider the recovery of the coefficients

$$C(h) = 1/2 - h/25\ (h/100 + 1) \quad \text{and}$$
$$K(h) = 3/2 + h/25\ (h/100 + 1).$$

The system was simulated under gravity drainage for 1 unit of time, at which point suction was applied and the lower boundary pressure linearly pulled to $h = -100$ at $t = 100$. Uniform grids ranging from 1 free node to 10 free nodes were used as the nodal basis for both C and K. As is clear from figure 5.9, a coarse nodal basis leads to error in the recovery, as does a

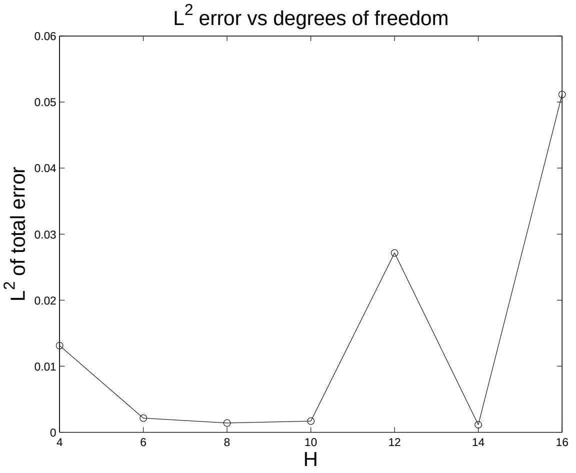


Figure 5.10: Error summary for uniform nodal bases

nodal basis that is highly refined. There appears to be an optimal uniform grid in the interval of 6 to 10 free nodes. As remarked upon in the one parameter recovery, here again the overshoot undershoot feature is evident in the case of the 16 dimensional C nodal experiment, and to a lesser extent in the related K coefficient.

Arguments similar to those in chapters 2 and 3 suggest that the error in a lower dimensional basis arise from the failure of the polygonal basis to approximate the true coefficient. In addition, there is increasing uncertainty inherent in the linearization of the adjoint problem. Recall that the relative size of the time interval between successive nodal elements corresponds to a larger possible region from which the adjoint coefficient approximation is chosen. Evidently, the error resulting from solving an approximate adjoint problem decreases as the nodal basis is refined. Therefore, the error seen in the highly refined experiments are not a result of the failure of the polygonal basis to approximate the coefficients nor the adjoint approximation. Instead, we consider this error to arise from numerical instabilities as the

Index	function for $t > 1$
BC1	$s(t) = -t$
BC2	$s(t) = -t^2$
BC3	$s(t) = -t^{1/2}$
BC4	$s(t) = -1/2t$
BC5	$s(t) = -2t$
BC6	$s(t) = -4t$
BC7	$s(t) = \begin{cases} -10 & 1 < t < 10 \\ -25 & 10 < t < 25 \end{cases}$

Table 5.2: Suction functions used in boundary experiments

integrals quantities involved in the recovery become increasing smaller. At some point, the numerical error begins to dominate this calculation, and the error begins to build. Also, as in the one parameter case, an error cascade develops as errors in previous time strips are compounded. While this is not in the scope of the current research, a detailed numerical analysis of the error is possible to make these statements more precise.

5.7.3 Boundary Condition

We now consider the effect of various boundary conditions on both the data and the recovered coefficients. In the Phase 2 experiment, we are free to chose the boundary state data at the left endpoint, which we denote $BC(t)$. In all experiments in this section, we allow the system to drain under gravity for 1 unit of time. We then apply suction at $z = L$, choosing this control from either a linear family, a power family or a step function, for a total of seven distinct boundary conditions. We refer to these as {BC1,BC2,BC3,BC4,BC5,BC6,BC7}, and provide the equations of each in table 5.7.3.

The coefficient recovery algorithm is applied with the single break point prescribed at $h = -10$.

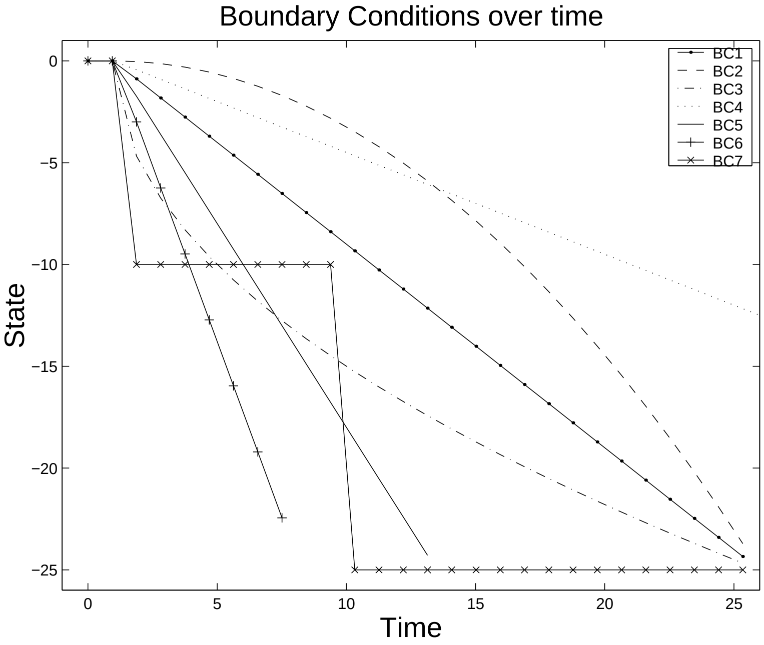


Figure 5.11: Boundary Conditions

Figure 5.11 suggests another classification of the boundary conditions. Notice that `BC1, BC2` and `BC4` all reach $h = -10$ after time 11. The other suction functions reach the cutoff state much more rapidly. Since the observation nodes are uniformly distributed in time, the subsequent experiments may be classified according to the number of data nodes used. The ordering in this method would then be {`BC4,BC2,BC1,BC5,BC3,BC6,BC7`}, where we begin with the condition leading to the largest observation set.

The pressure profiles are exhibited as recorded on equidistant time nodes for all combinations of the linear coefficients $C(h) = 3 \pm (1/10)h$ and $K(h) = 3 \pm (1/10)h$. In the physical setting, runs 3 and 4 are the most relevant, since K is generally assumed to be monotone decreasing. We remark that the data for run 1 and run 4 appear nearly identical for all simulated boundary conditions.

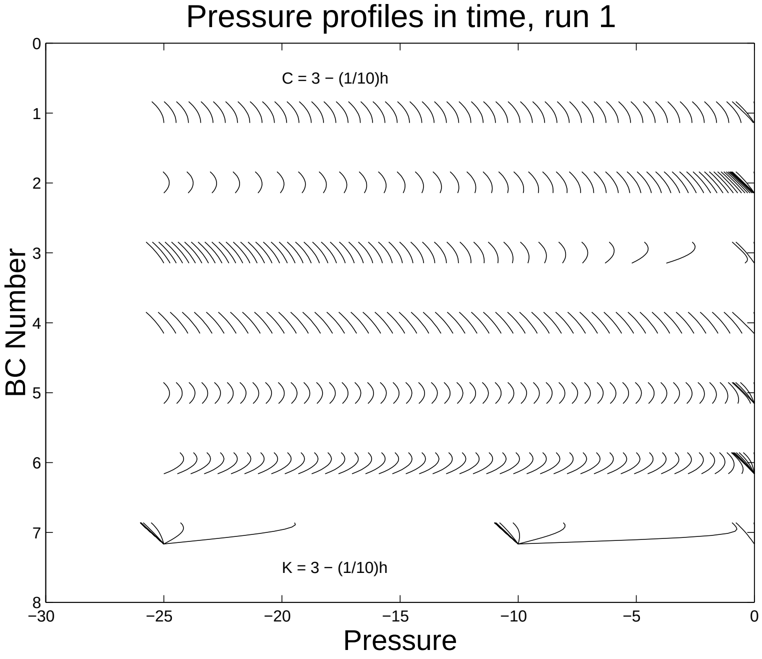


Figure 5.12: Simulations with boundary condition and C up K up

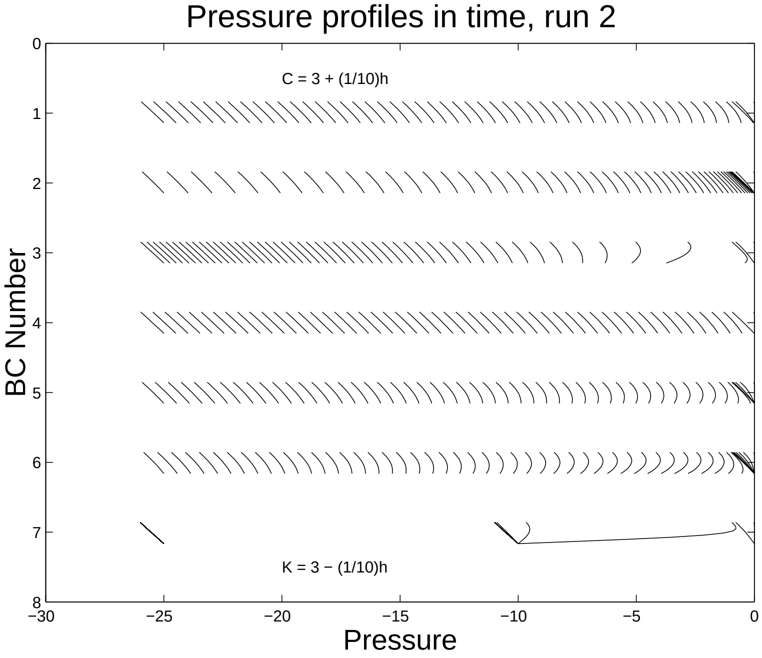


Figure 5.13: Simulations with boundary condition and C down K up

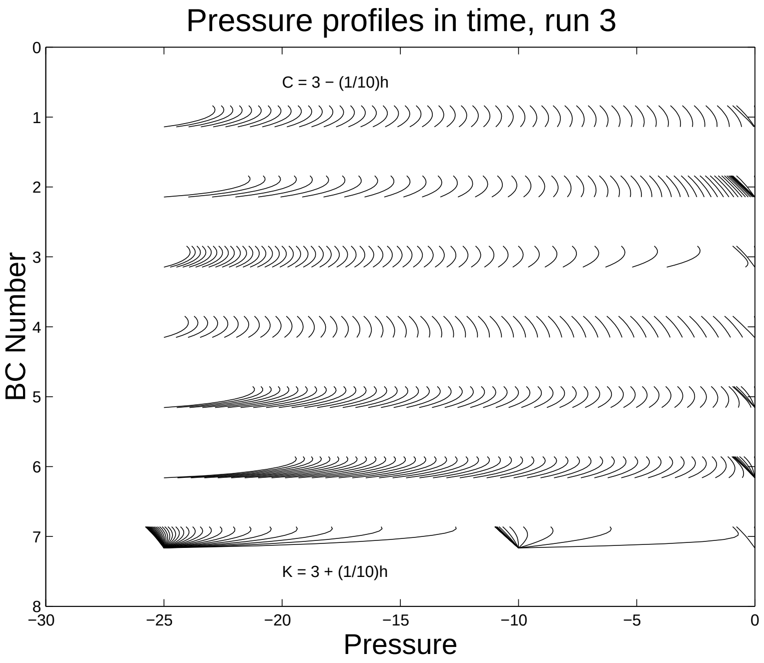


Figure 5.14: Simulations with boundary condition and C up K down

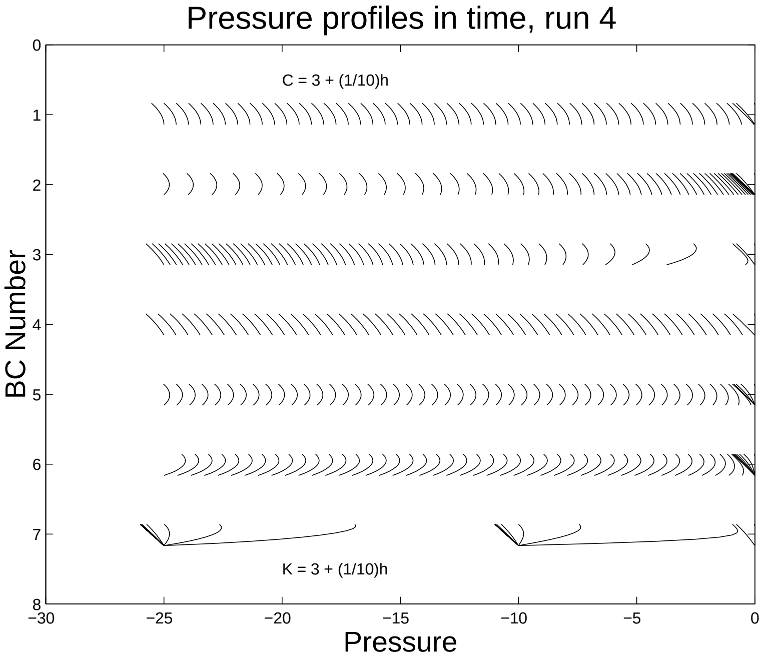


Figure 5.15: Simulations with boundary condition and C down K down

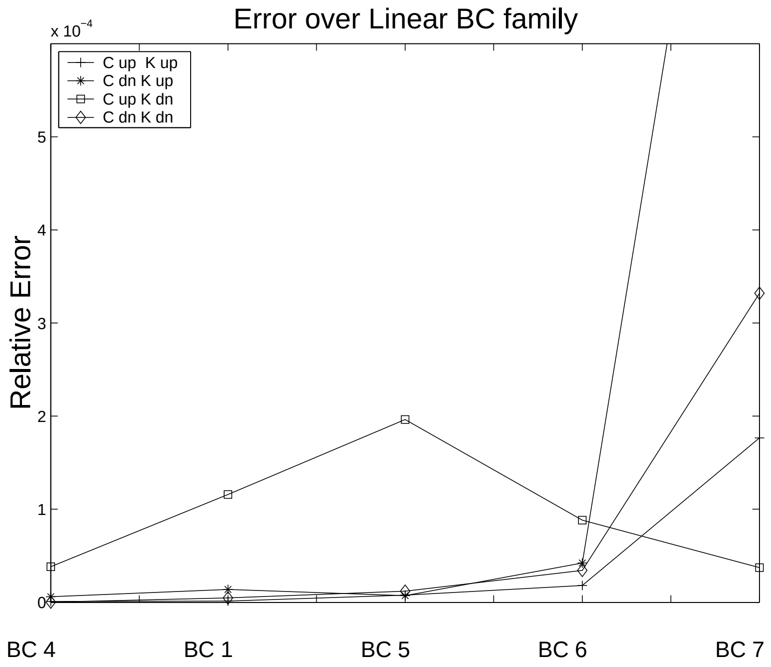


Figure 5.16: Error over Linear family

Once the numerical simulations were complete, the relative error indicator was computed for each family. Figure 5.16 depicts the error in the linear family recovery, while figure 5.17 contains information about the power family. Notice that both figures contain BC1 as the base reference. These two figures suggest that applying suction more slowly leads to better recovery. Is is likely that the increase in observation data is the reason for this. This effect is slight in comparison to the large error jump in the BC7 experiment for nearly all coefficient combinations. It appears that there might be a point at which the identification suddenly become accurate, and that once beyond this critical point, there is little improvement in the identification.

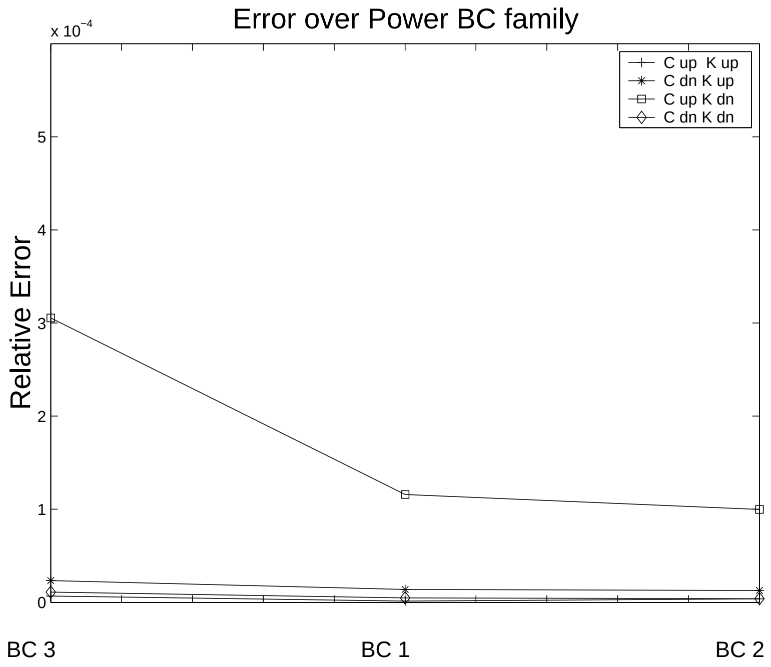


Figure 5.17: Error over Power family

5.7.4 Scaling of the Inversion

We now consider applying the integral identities over increasing smaller intervals, in an attempt to explore the scaling of the inversion process. Here we solved both the direct and adjoint constant coefficient problems via Fourier transform methods in the space variable and considered time to be continuous. The resulting truncated approximate solution was computed Maple environment for time intervals ranging from 2^{-1} to 2^{-10}. Derivatives were implemented symbolically, and the subsequent log log plots indicate relative scaling of of the matrix entries. The M_{11} entries were all negative, and so the absolute values were used in the log computation. The M_{22} entry was difficult to construct via our numeric method. In an effort to admit a classical solution, a linear function was chosen as data. The Fourier solution series in space, however, introduced a large variation in the time derivative

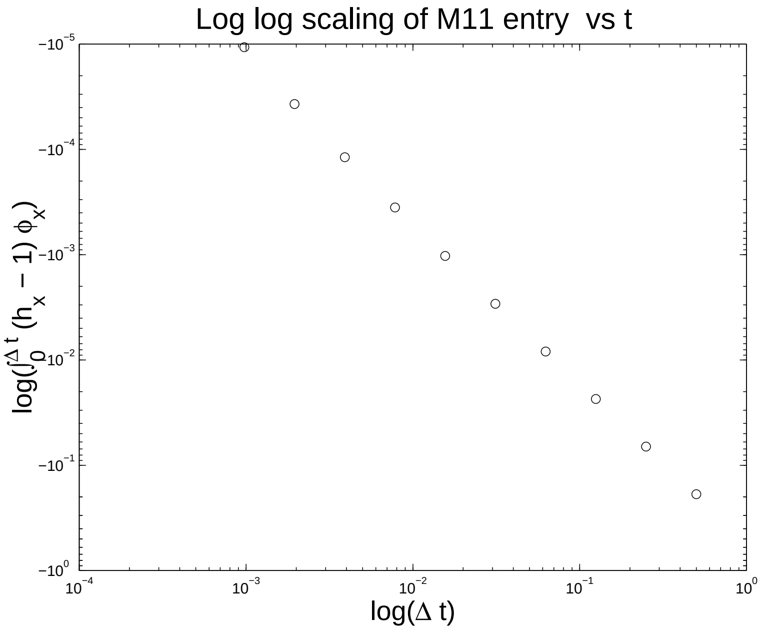


Figure 5.18: Scaling of M_{11} entry

computation, a term which was required to compute the scaling of the M_{22} entry. A Laplace solution might lead to a more robust computation, but this was not pursued.

The manner in which these terms scale can provide interesting information about the recovery process. If, for example, the first column entries were to scale more quickly to zero than the second, this might indicate that the recovery of C might be more robust that that of K. Similarly, if the first row were to scale more quickly than than the second, this might lead one to believe that the state data p contains relatively less information than does q as the time interval decreases. Here, however, the entries appear to all scale in an approximately linear fashion, suggesting that the recovery fails in the case of nodal basis refinement not because of an algorithmic instability, but rather from numeric limitations.

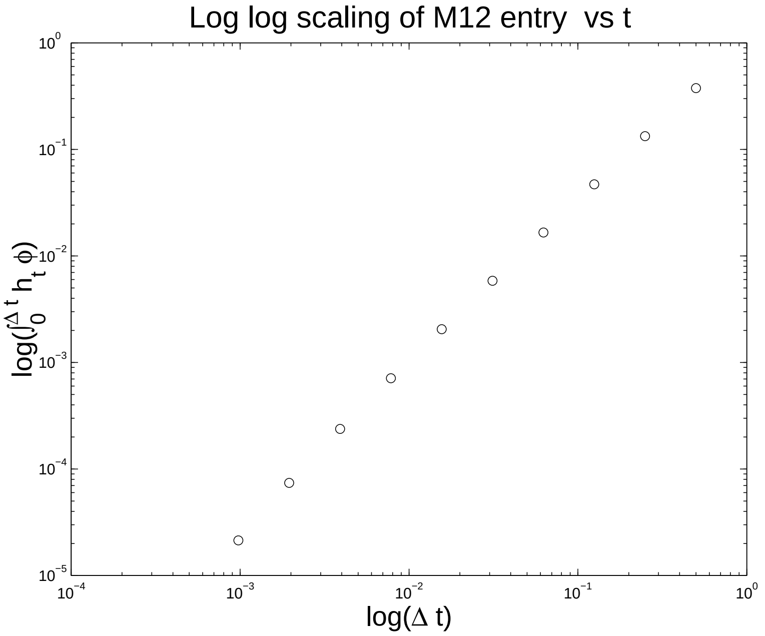


Figure 5.19: Scaling of M_{12} entry

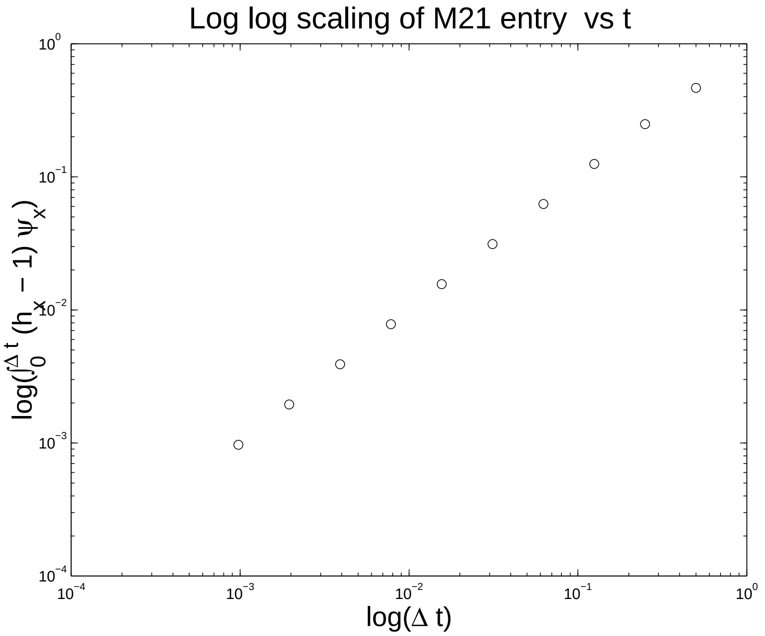


Figure 5.20: Scaling of M_{21} entry

5.7.5 Scaling of the C coefficient

In this section the relative scaling of C to K is explored for two different parameter families. We fix $K(h) = 1$, and allow C to take values from either

$$C(h) = \alpha \quad \text{or} \quad C(h) = \alpha(1 - 1/4h),$$

where α is chosen from the set $\{1/100, 1/10, 1, 10, 100\}$. Often the capacity function $C(h)$ is assumed to be several orders of magnitude smaller that that $K(h)$. In this series of experiments we seek to understand the effect that this might have on recovery. Only the recovery error on the first strip is recorded, so as to limit the cumulative error effect. Also, this problem was solved in the Phase 1 setting, by observing the system under drainage for 1 unit to time, and maintaining zero head at the base of the column. The experimental observations were taken in from the state interval $[0 \ -0.8]$. The variable rate resulted in experiments containing different number of observations used in the recovery. For example, the choice of $\alpha = 1/100$ lead to rapid equilibration, and the system reached the state value after only 6 observation nodes. Conversely, $\alpha = 1$ required 594 time observations to reach this same state. For values of α greater than 1, all experimental observations were used in for recovery. As one might expect, the increase in nodal information used in the $\alpha = 1e + 1$ recovery experiment apparently leads to better coefficient recovery. While not conclusive, this suggests that there is a slight improvement in recovery as α increases. This effect is small, on the order of $1e - 5$, using the error indicator. This would appear larger had a norm been used.

A linear family was also explored. In figure 5.22, the relative error attains a well defined minimum for α in the neighborhood of 1. Again, the

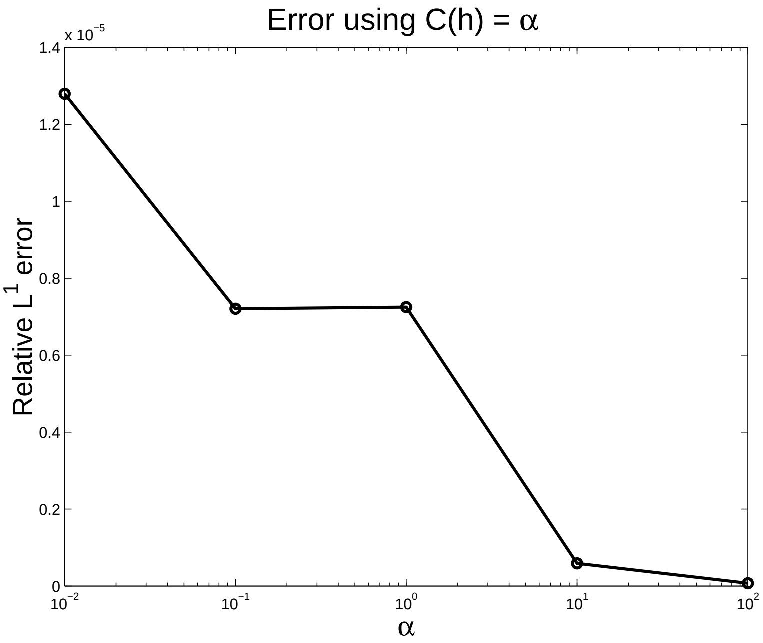


Figure 5.21: C scale const

scale is quite small in this plot, but the sharp drop between the error at $\alpha = 1e - 2$ and $\alpha = 1e - 1$ is marked. Recall that only 6 observations are used to compute this update, and therefore this might be attributable to lack of data. A more interesting feature of this plot is the increase in error for α larger than 1. In this range, all observation data was used in recovery, and so the size of the observed data set does not explain this feature. While not conclusive, this suggests that there is a real and measurable decrease in recovery as α increases. The error plot suggests that large values of C relative to K might be more difficult to recover than smaller values of C.

5.8 Recovery from Matlab generated data

In this section, we consider recovery using Matlab generated datasets. We plot both the true coefficient as well as the recovered coefficient. In the

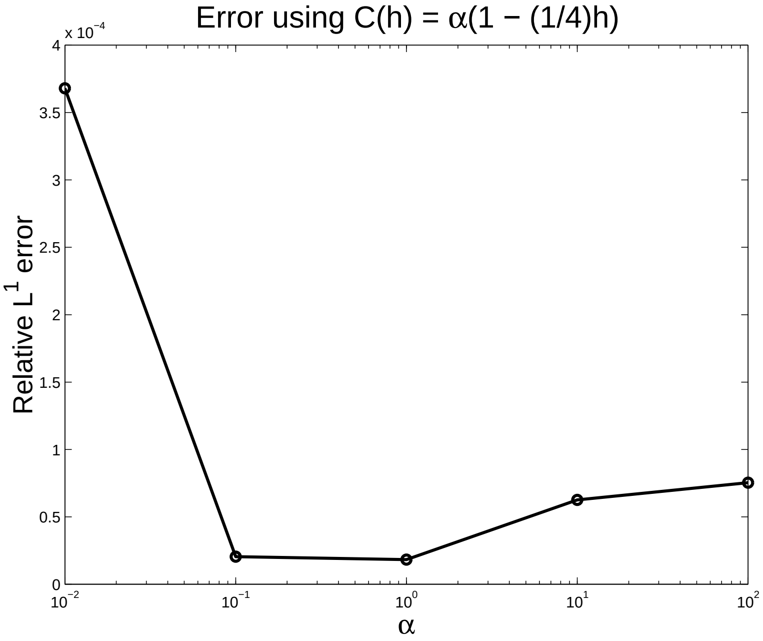


Figure 5.22: C scale linear

first series of plots, various unphysical parameters are recovered in an effort to understand the features that might be changing to recover accurately.

In figure 5.23, K is taken to be 1 and $C(h) = 1 - 0.4\sin(h/15)$. The non-iterative algorithm was applied. The resulting solution is visually accurate, with the K coefficient approximating the correct constant function is some integral mean sense. As noted earlier, an increase in the dimension of the nodal basis leads to a much larger overshoot undershoot effects then is evidenced in this in this figure of the six dimensional basis. More difficult coefficients were then attempted. In figure 5.24, the K coefficient was changed to $K(h) = 2 - (|h - 1|/50)^{(1/5)}$. The nodal basis and $C(h)$ used in the previous example were preserved. A sample of other coefficient recoveries are provided in the figures 5.25,5.26 and 5.27. The recovery apparently becomes more difficult as a function of the gradient in the true coefficient. If the true coefficient is steep, as it is in figure 5.27, the method

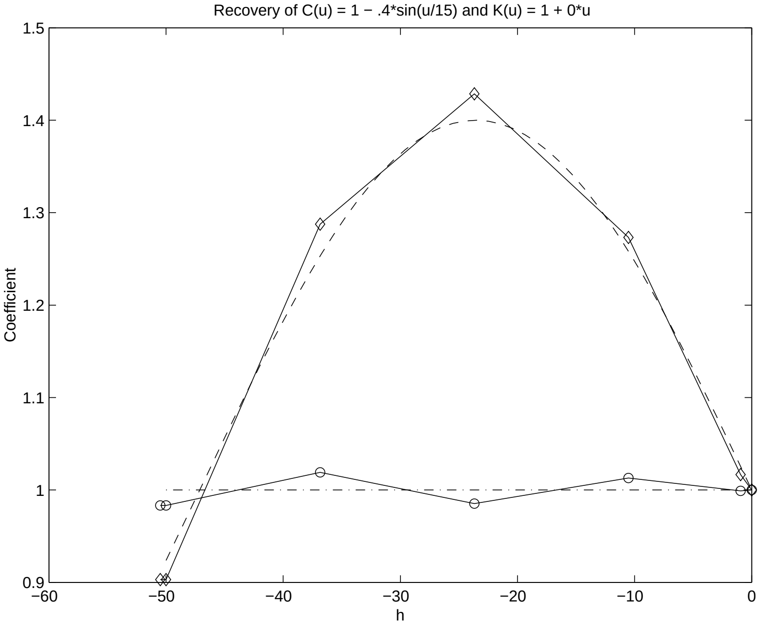


Figure 5.23: A Simple Recovery

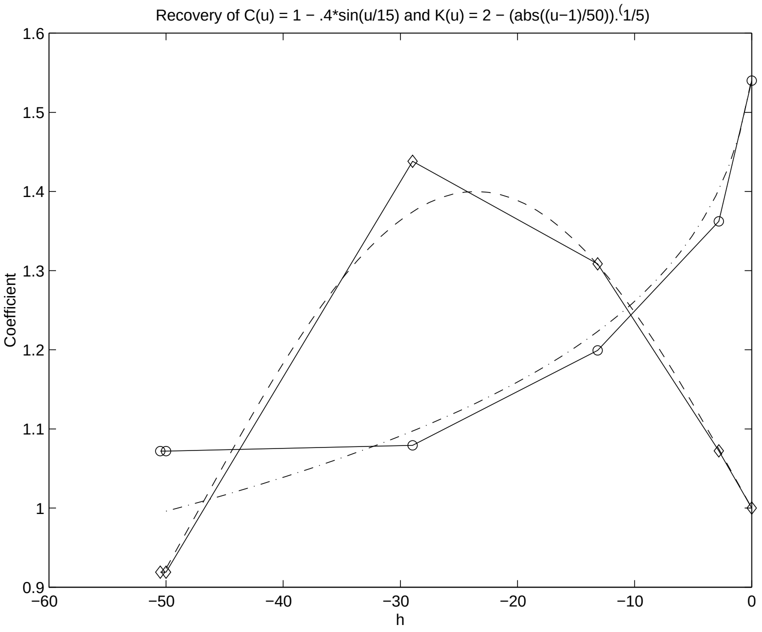


Figure 5.24: A Slightly Harder example

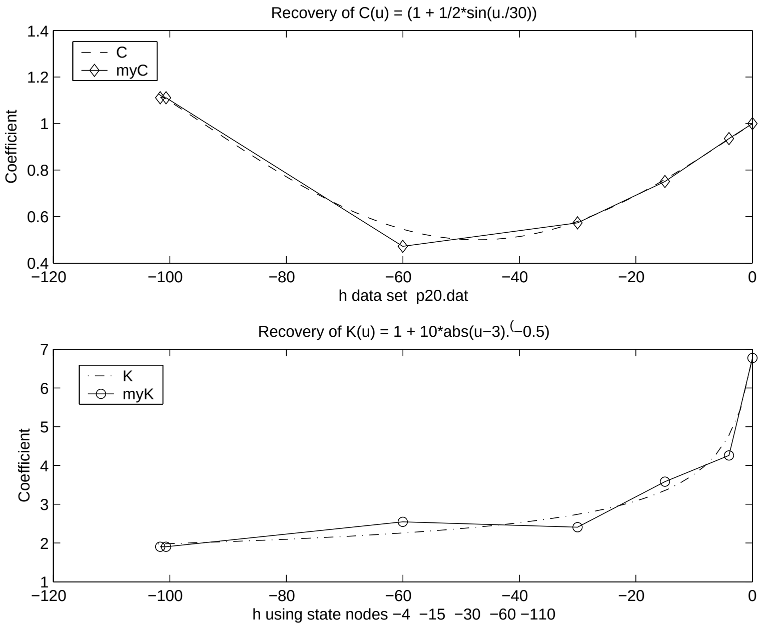


Figure 5.25: Sample recovery 1

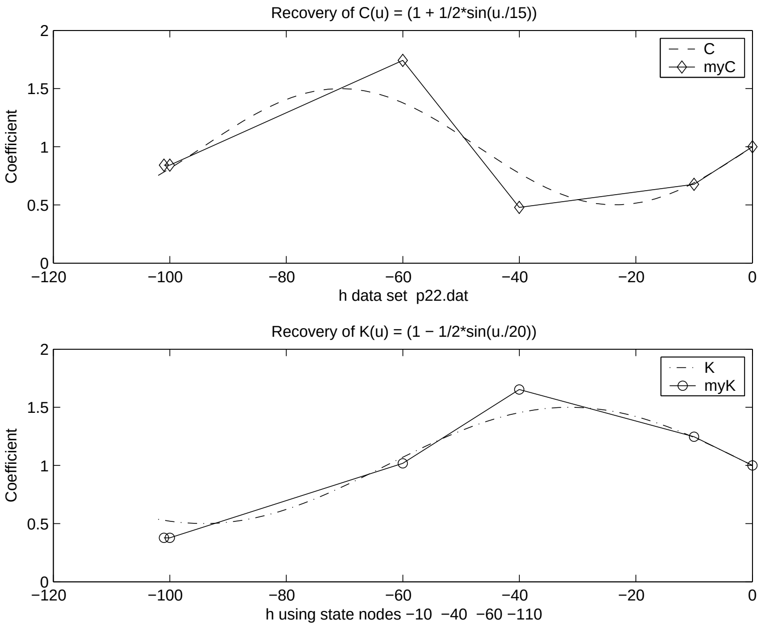


Figure 5.26: Sample recovery 2

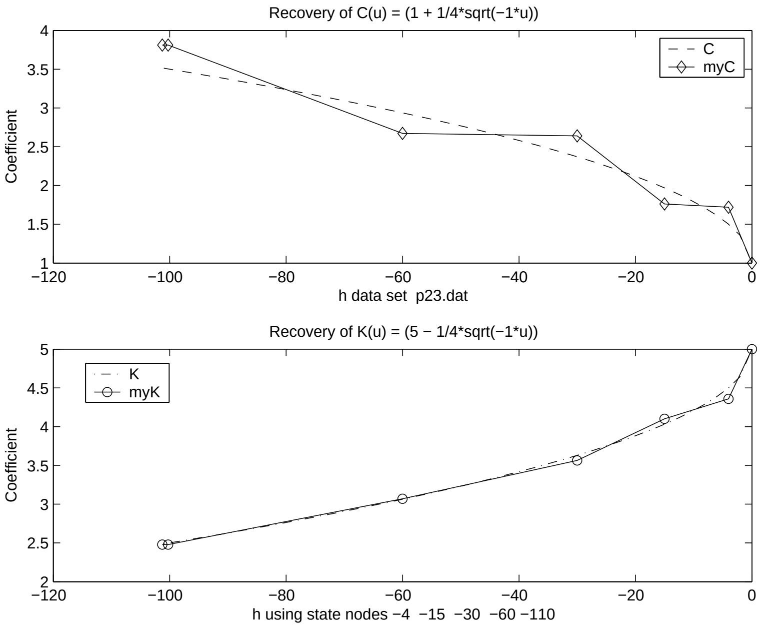


Figure 5.27: Sample recovery 3

has difficulty. These plots also suggest that recovery in regions which correspond to a increase of the function might prove more challenging than regions in which the function is decreasing.

5.9 Van Genuchten Family Recovery

Phase 2 numerical suction experiments were conducted over twelve soil texture classes. The system was initially taken to be at drainage equilibrium, and linear suction function $s(t) = t$ was applied to the bottom boundary for 100 units of time. The van Genuchten hydraulic functions $\Theta(h)$ and $K(h)$ were implemented, where

$$\Theta(h) = \Theta_r + \frac{\Theta_s - \Theta_r}{[1 + (\alpha h)^n]^{1/n-1}},$$

which can be rewritten to form the relative saturation S_e,

$$S_e(h) = \frac{\Theta(h) - \Theta_r}{\Theta_s - \Theta_r},$$

which is in turn used to construct the hydraulic conductivity function

$$K(h) = K_0 S_e(h)^\ell \left\{1 - [1 - S_e(h)^{n/(n-1)}]^{1-1/n}\right\}^2 .$$

We recall $C(h)$ to be the derivative of $\Theta(h)$. Both parameters α and n are used to control the shape of the curve $\Theta(h)$. K_0 is the matching point at saturation, and need not be equal to K_s, the saturated conductivity. Finally, ℓ is considered to be a measure of soil pore connectivity, and is normally taken to be 1/2.

The plots (5.28-5.31) demonstrate the recovery of several members the van Genuchten family. The success of the recovery is variable, ranging from a numerical failure in the simulation representing sand (5.30), to recovery of parameters associated with silt (5.31). We also comment that these solution are dependent on the nodal basis chosen. The appearance of the numerical instability of overshoot and undershoot was used to tune the nodal basis.

The failure of $K(h)$ in the Clay Loam problem is interesting. The recovery is visually quite accurate until the third node at $h = -30$, which corresponds to the 300^{th} time observation of 1000. The recovery in the final time strip corresponding to h in the interval $[-110 - 30]$ contains 700 observations.

The parameter values used in these experiments were taken from the documentation of Hydrus, and represent average values. The table entries for the twelve classes are provided in appendix E for reference.

5.10 Noisy data

Here the ability of the numerical method to perform recovery from noisy data is explored. Typically, successful parameter estimation becomes

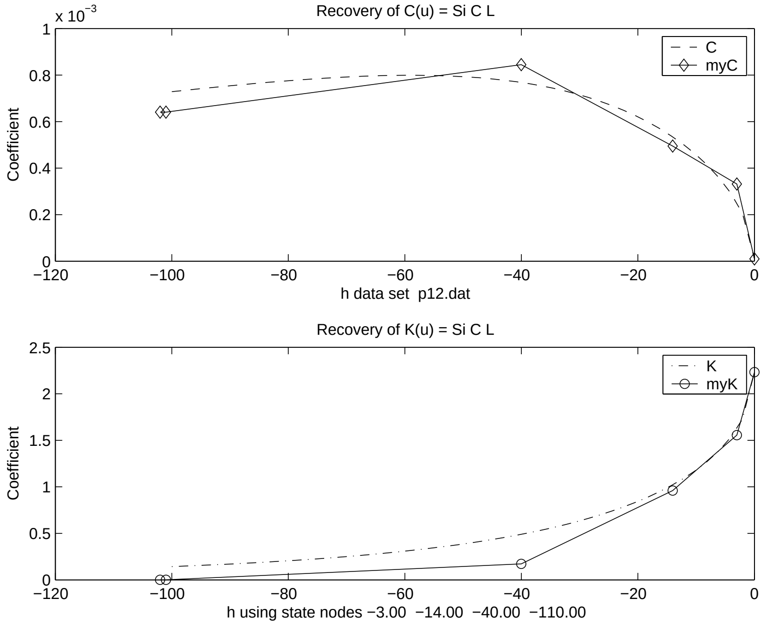


Figure 5.28: Recovery of coefficients associated with Si C L

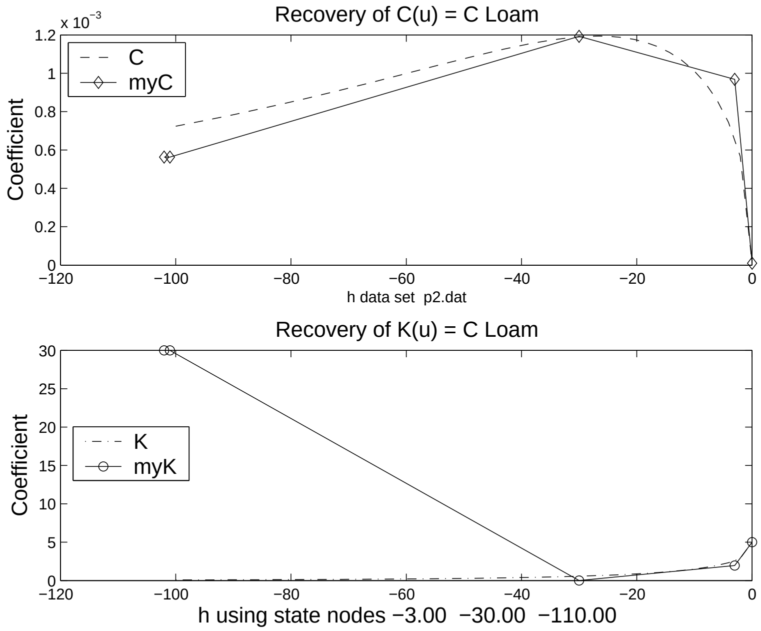


Figure 5.29: Recovery of coefficients associated with C Loam

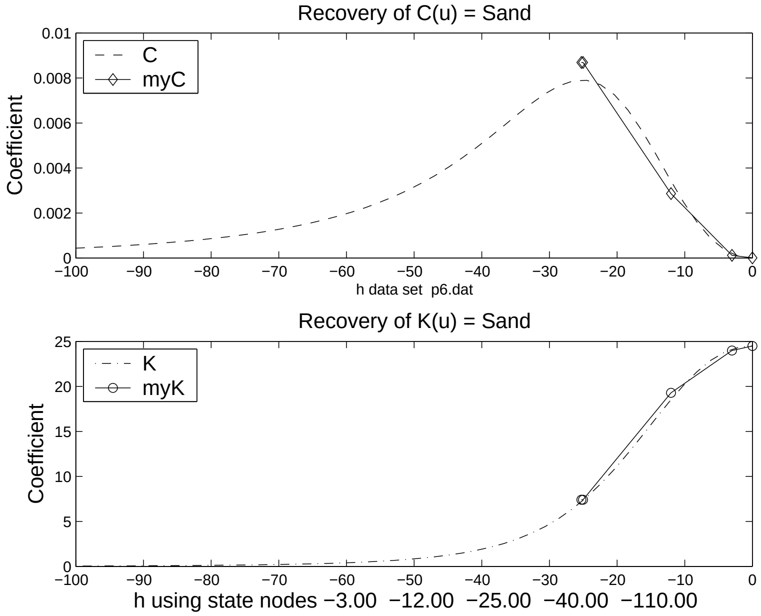


Figure 5.30: Recovery of coefficients associated with Sand

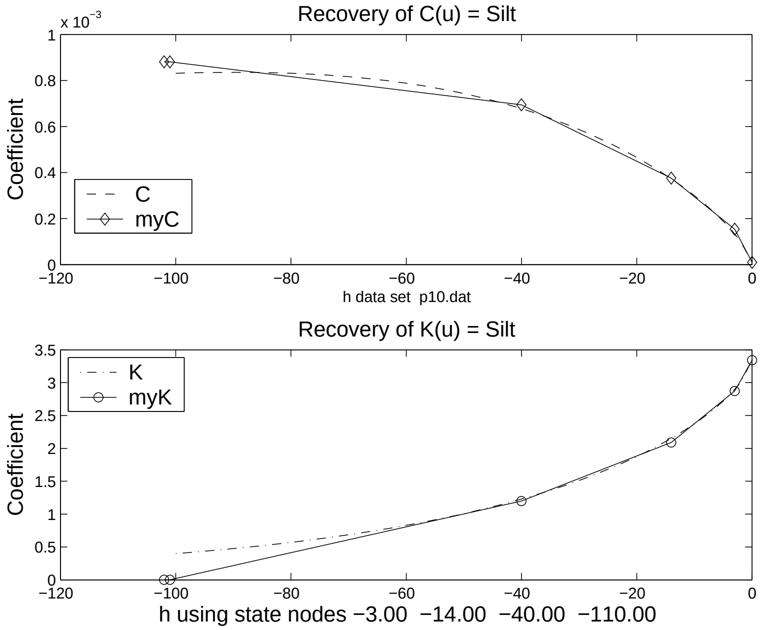


Figure 5.31: Recovery of coefficients associated with Silt

increasing difficult in the presence of noise. Recall the integral identities on which the identity method is based. Only an integrated quantity of the observed data used, which allows much of this noise to cancel. This is a significant observation, and a highly desirable feature.

Data was generated using the coefficients

$$C(h) = \frac{1}{2} - \frac{h}{25}\left(\frac{h}{100} + 1\right) \text{ and } K(h) = \frac{3}{2} + \frac{h}{25}\left(\frac{h}{100} + 1\right)$$

The system was simulated under gravity drainage for 1 unit of time, at which point suction was applied and the lower boundary pressure linearly drawn to $h = -100$ at $t = 100$. A uniform grid of 8 nodes was used for both C and K, making these experiments related to those concerning the dimension of the basis. The observation data was simulated numerically, and then relative uniform random noise ranging from 1% to 15% was added. This perturbed data was then submitted to the interactive recovery scheme. The figure 5.32 represents recovery under 10% noise, and suggests the robustness of the method.

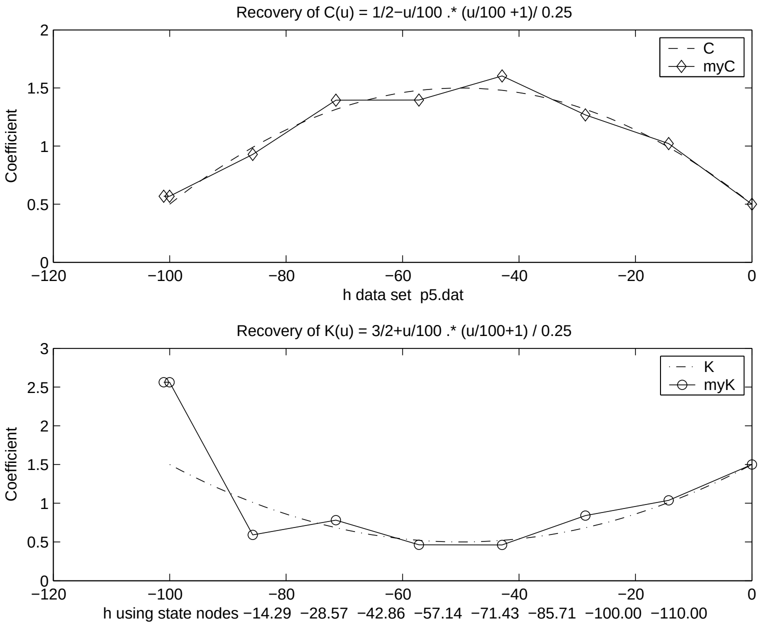


Figure 5.32: Recovery with 10% uniform noise

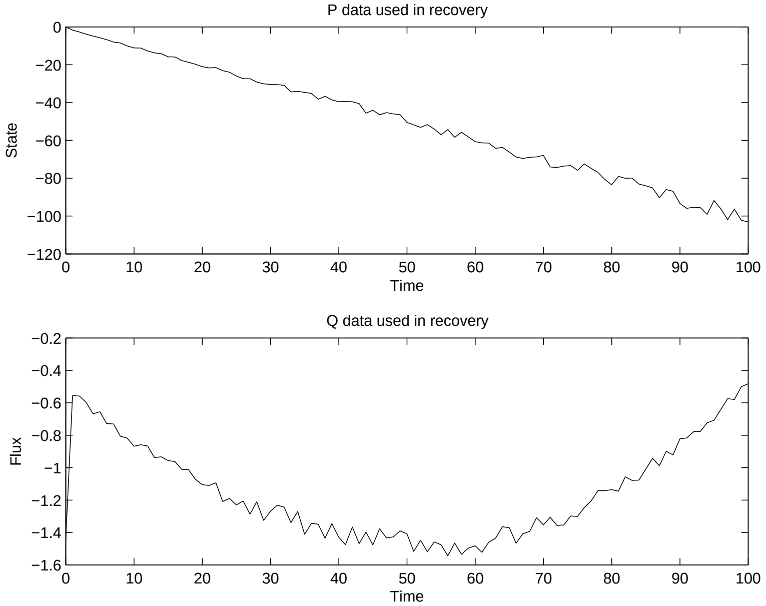


Figure 5.33: Data with 10% uniform noise

Chapter 6

CONCLUSIONS

Adjoint methods are used to construct explicit representations for the input to output mappings associated with the inverse problem of identifying unknown coefficients from over specified data measured on the boundary. The integral equations presented here provide a means for proving that the input output maps are explicitly invertible. A practical computational method has been developed and presented. This algorithm was then used to explore numerous features of the input to output mapping.

The method may be viewed as a tool to gather information about mapping. Numerical experiments involving the Richards equation suggest that unknown soil coefficients might be successfully identified via future refinements of the algorithm, although it might be quite difficult to compete with sophisticated output least squares methods. Additionally, the integral identity based algorithm appears to be robust under noise.

A `Matlab` implementation of the integral identity approach provides a great deal of explicit information about recovery of the unknown coefficients. This is viewed as the main contribution of this work. As a result, the analysis of the recovery process, both successful and unsuccessful, becomes possible since all components are transparent. Such information is not so readily obtained from an output least squares approach nor from equation error techniques.

The preliminary results of the implementation are highly encouraging. The adjoint techniques might provide a basis for the successful coefficient identification methods in the future.

We hope to extend this work in the following directions:

- Application to physical experimental data.
- Examine how the controlled quantities in the physical experiment impact recovery. It might be possible to determine if one experimental setting is better than another.
- Implement the alternative cumulative flux method.
- Consider linear and nonlinear scalings of the problem. This might allow recovery of coefficients exhibiting rapid change.
- Explore the coupling of the water content parameter $\Theta(t)$ and hydraulic conductivity K. This a current criticism of the van Genuchten and Brooks Corey methods. The integral identity method does not utilize a coupling.
- Develop a practical criteria for the adaptive selection of nodal basis. If possible, this might allow the method to be used as a robust identification tool, in addition to yielding information in the case of identification failure.
- Directly compare the integral identity solutions to OLS solutions, while considering the constraints that are commonly applied in the OLS setting.

Bibliography

[1] J. Cannon. *The One-Dimensional Heat Equation.* Addison - Wesley, Menlo Park, California, 1984.

[2] J. Cannon and P. DuChateau. Design of an experiment for the determination of a coefficient in a nonlinear diffusion equation. *Int. J. Eng. Sci.*, 25(8):1067–78, 1987.

[3] J. Cannon and P. DuChateau. Indirect determination of hydraulic properties of porous media. *International series of numerical mathematics*, 114, 1993.

[4] J. Cannon and P. DuChateau. Some asymptotic boundary behavior of solutions of nonlinear parabolic initial boundary value problems. *JMAA*, 68(2):536–547, 1997.

[5] A. Carasso. Determining surface temperatures from interior observations. *SIAM J. Appl. Math*, 42(3):558–574, June 1982.

[6] G. Chavent. On the theory and practice of non-linear least-squares. *Adv. Water Resources*, 14(2):55–63, 1991.

[7] G. Chavent and K. Kunisch. The output least squares identifiability of the diffusion coefficient from a h_1- observation in a 2-d elliptic equation. Technical Report 4067, Institut National de Recherche en Information et en Automatique (INRIA), 2000.

[8] G. Chavent and P. Lemonnier. Identification de la non-linearité d'une équation parabolique quasilineaire. *Applied math. & opt.*, 1(2):121–161, 1974.

[9] M. Spivack D. E. Reeve. Recovery of a variable coefficient in a coastal evolution equation. *Journal of Computational Physics*, 151:585–596, 1999.

[10] P. DuChateau. An inverse problem for the hydraulic properties of porous media. *SIAM J. Appl. Math*, 28(3):611–632, June 1997.

[11] L. Evans. *Partial Differential Equations.* American Mathematical Society, Providence, Rhode Island, 1991.

[12] B. H. Gilding. Qualitative mathematical analysis of the richards equation. *Transport in Porous Media*, 5:651–666, 1991.

[13] C. Groestch. *Inverse Problems in the Mathematical Sciences.* Vieweg, Braunschweig, Wiesbaden, 1993.

[14] Guenther and Lee. *Partial Differential Equations of Mathematical Physics and Integral Equations.* Prentice Hall, Mineola, NY, 1988.

[15] J. Jaffre H. B Ameur, G. Chavent. Refinement and coarsening of parameterization for the estimation of hydraulic transmissivity. In *Proceedings of Inverse Problems in Engineering*, 3, Port Ludlow, WA, 1999. 3rd Int'l Conf on Inverse Problems in Engineering.

[16] M. Hanke and O. Scherze. Error analysis of an equation error method for the identification of the diffusion coefficient in a quasilinear parabolic differential equation. *SIAM J. Appl. Math*, 59:1012–1027, 1999.

[17] D. Hinestroza and D. Murio. Identification of transmissivity coefficients by mollification techniques. part i: 1-dimensional elliptic and parabolic problems. *Comp. Math Applications*, 25:59–79, 1993.

[18] M. K. Hubbert. Darcy's law and the field equations of hte flow of underground fluids. Technical Report 5, l'Association Internationale d'hydrologie scientifique, 1957.

[19] V. Isakov. *Inverse Source Problems.* AMS, Providence, Rhode Island, 1980.

[20] P. Knabner and S. Bitterlich. An efficient method for solving an inverse problem for the richards equation. *Journal of Computational and Applied Mathematics*, 147(1):153–173, 2002.

[21] R. Nabakov. An inverse problem for porous medium equation. In J. Gottlieb and P. DuChateau, editors, *Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology*, pages 155–163. Kluwer, Dordrecht, Boston, London, 1996.

[22] V. A. Solonnikov O. A. Ladyzhenskaya and N. N. Uralceva. Linear and quasi-linear equations of parabolic type, 1969.

[23] G. R. Richter. Numerical identification of a spatially varying diffusion coefficient. *Math Comp*, 36:375–386, 1981.

[24] M. N. Ozisik Y. Jarny and J. P. Bardon. A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *Int. J. Heat Mass Transfer*, 34(11):2911–2918, 1991.

Appendix A

EXISTENCE UNIQUENESS FOR ONE PARAMETER

N.B. : The notation used in this section is local

In this section we present a typical existence uniqueness argument for solutions of the one parameter problem (2.1) under the assumption that $D : \mathbb{R} \to \mathbb{R}$ is strictly positive and bounded in $L^\infty(\mathbb{R})$.

It is more convenient to treat the equivalent transformed problem with homogeneous boundary conditions, and we do so. In the discussion that follows, $a(u)$ is related but not necessarily equal to $D(u)$, as are the Lipschitz constants.

We begin by considering the IVBP

$$\begin{aligned} \partial_t u - \partial_x(a(u)\partial_x u) &= f(x,t) && (x,t) \in \Omega \times (0,T) \\ u(x,0) &= u_0(x) && x \in \Omega \\ u(0,t) = 0 \quad u(1,t) &= 0 && 0 < t < T \end{aligned} \tag{A.1}$$

where Ω is the open region $(0,1) \in \mathbb{R}$. Assume that $a : \mathbb{R} \to \mathbb{R}$ is in $L^\infty(\mathbb{R})$ and that there exist constants C_0 and C_1 such that $0 < C_0 < a(x) \leq C_1$ for all $x \in \mathbb{R}$. We begin by establishing some notation: Define

$$V = H_0^1(\Omega) \subset H \subset V' = H^{-1}(\Omega)$$

and define the bilinear form

$$a(u,v) = \int_{\Omega} a(u)u_x \cdot v_x dx \qquad \text{for } u, v \in V$$

Now

$$|a(u,v)| \leq C_1 |\int_{\Omega} u_x \cdot v_x dx| \leq C_1 ||u||_V ||v||_V$$

This estimate implies the existence of a continuous nonlinear map from V to V', given by

$$\langle A(u), v \rangle_{V' \times V} = a(u,v) \qquad \text{for } u, v \in V,$$

where

$$||A(u)||_{V'} \leq C_1 ||u||_V$$

Now let $u \in L^2[0, T : V(\Omega)]$ be a weak solution if it satisfies the problem

$$\begin{aligned} (u'(t), v)_H + a(u(t), v) &= (f(t), v)_H \qquad \text{for all } v \in V \\ u(0) &= u_0 \end{aligned} \tag{A.2}$$

Equivalently, $u(t)$ must be a solution of

$$u'(t) + A(u(t)) = f(t), \qquad u(0) = u_0$$

If $f \in L^2[0, T : H(\Omega)]$ and $u_0 \in H$, it can then be shown that there exists a unique weak solution to the IVBP (A.2). This implies the existence of a subsequence of solutions converging to a limit which can then be shown to be a solution for the problem (A.2). Uniqueness follows from the assumption that $a(\cdot)$ has a strictly positive lower bound.

Existence: V is compactly embedded in H. There then exists a orthonormal basis of H which is also an orthogonal basis for V. Call this basis $\{w_k\}$. We can now define a solution

$$u_N(t) = \sum_{k=1}^{\infty} c_{k,N}(t) w_k \quad \text{for } N = 1, 2, 3, ...$$

which satisfies

$$(u_N'(t), w_j)_H + a(u_N(t), w_j) = (f_j(t), w_j)_H \qquad \text{for } j = 1, 2, ..., N \tag{A.3}$$
$$(u_N(0) - u_0, w_j)_H = 0 \qquad \text{for } j = 1, 2, ..., N.$$

Since the basis is orthonormal, this collapses to a system of odes in the coefficient $c_{j,N}(t)$,

$$c_{j,N}'(t) + \sum_{k=1}^{N} c_{k,N} \int_0^1 a(u_N) \partial_x w_k \cdot \partial_x w_j = f_j(t)$$
$$c_{j,N}(0) = -(u_0, w_j)_H.$$

For fixed N, this system has a unique solution over the time interval $[0, T_N]$, with $T_N \leq T$. Energy estimates provide the *a priori* bounds needed to complete the argument. All estimates follow from (A.3). Let $w_j = u_N$, which is possible since (A.3) holds for all $j \leq N$, and therefore for u_N as well. Then

$$\frac{d}{dt}||u_N(t)||_H^2 + C_0||u_N(t)||_V^2 \leq \frac{1}{C_0}||f(t)||_H^2. \tag{A.4}$$

Equation (A.4) is the basis for obtaining bounds of both $||u_N||_{L^\infty[0,T:H(\Omega)]}$ (discard $||u_N||_V$ term and integrate) and $||u_N||_{L^2[0,T:V(\Omega)]}$ (integrate then throw away $||u_N(T)||_H$ term).

$$||u_N||_{L^\infty[0,T:H]} \leq ||u_0||_H^2 + ||f(t)||_{L^2[0,T:H]}^2 = M_1 \tag{A.5}$$

$$||u_N||_{L^2[0,T:V]} \leq \left(\frac{1}{C_0}\right)||u_0||_H^2 + \left(\frac{1}{C_0}\right)^2 ||f||^2_{L^2[0,T:H]} = M_2 \qquad \text{(A.6)}$$

A bound for u'_N in $L^2[0,T,V'(\Omega)]$ is also needed. To obtain this bound, define a projection P_N from $V \to V_N$. The estimate is due to the fact that u_N is a solution to the weak problem, which implies

$$\langle u'_N(t) + A(u_N(t)) - g(t), P_N V\rangle_{V' \times V} = 0 \quad \text{for all} \quad v \in V,$$

which can be interpreted as

$$||u'_N(t)||_{L^2[0,T,V']} = ||P_N^T(A(u_N(t)) - f(t)||_{L^2[0,T:V']}$$

From this, the estimate

$$||u'_N||_{L^2[0,T:V']} \leq C_1||u_N||_{L^2[0,T:V]} + ||f||_{L^2[0,T:V']} \leq M_3 \qquad \text{(A.7)}$$

follows.

The estimates (A.5, A.6 and A.7) imply weak convergence of both u_N in $L^2[0,T:V(\Omega)]$ and u'_N in $L^2[0,T:V'(\Omega)]$. In fact, the compact embedding of V in H implies

$$u_N(t) \to u(t) \quad \text{strongly in} \quad L^2[0,T:H(\Omega)]$$

It remains only to show that $u = \lim_N u_N$ is a solution of the original problem. For this purpose, we define

$$b(u) = \int_0^u a(s)ds$$

Since a is bounded,

$$C_0|u| \leq |b(u)| \leq C_1|u| \quad \forall u \in \mathbb{R}$$

It follows from

$$\partial_x b(u) = a(u)\partial_x u$$

and (A.5) and (A.6) that

$$||b(u_N(\cdot))||_{L^2[0,T:V(\Omega)]} \leq C_1 M_1 \quad \forall N$$

This results implies that

$$b(u_N(\cdot)) \to B_2 \qquad \text{weakly in } L^2[0,T:V(\Omega)]$$

Since V is compactly embedded in H,

$$b(u_N(\cdot)) \to B_2 \qquad \text{strongly in } L^2[0,T:H(\Omega)]$$

Noticing that $b(\cdot)$ is continuous on $\mathbb{R}$ and that $u_N(\cdot)$ converges strongly to B_2 in $L^2[0,T:H(\Omega)]$, then $b(u) = B_2$. Now consider

$$\begin{aligned} a(u,v) &= \int_\Omega a(u)u_x \cdot v_x \\ &= \int_\Omega (b(u))_x \cdot v_x dx \\ &= (b(u),v)_V - (b(u),v)_H \end{aligned}$$

Recalling the definition of $a(u,v)$,

$$\int_0^T \langle A(u_N(t),v\rangle_{V'\times V} dt = \int_0^T \left[(b(u_N),v)_V - (b(u),v)_H\right] dt$$

and then passing to the limit in N

$$\begin{aligned} \int_0^T \langle B_1, v\rangle_{V'\times V} &= \int_0^T \left[(b(u),v)_V - (b(u),v)_H\right] dt \\ &= \int_0^T \langle A(u(t),v\rangle_{V'\times V} dt \end{aligned}$$

Therefore, $A(u(t)) = B_1$, where B_1 is the weak limit of $A(u_N)$. Finally, passing to the limit in the discretized pde (A.3), it follows that $u(t)$ is a weak solution of the partial differential equation. Now $u'(t)$ can be written $u'(t) = f(t) - A(u(t))$, and since $f(t)$ and $A(u(t))$ are both in $L^2[0, T : V'(\Omega)]$, it follows that $u(t)$ must be in $L^2[0, T : V(\Omega)] \cap C[0, T : H(\Omega)]$. Also $u(0) = u_0$, and so $u(t)$ is a solution to the initial value problem.

Uniqueness: Assume that h_1 and h_2 are two weak solutions of (A.1), then $w = h_1 - h_2$ is a solution of

$$\langle \partial_t w(t), v\rangle_{V' \times V} + \langle A(h_1) - A(h_2), v\rangle_{V'\times V} = 0 \quad \forall v \in V, \quad w(0) = 0 \quad \text{(A.8)}$$

Then

$$\begin{aligned} \langle A(h_1) - A(h_2), v\rangle_{V'\times V} &= \int_U (b(h_1) - b(h_2))_x \cdot v_x dx \\ &= (b(h_1) - b(h_2), v)_V - (b(h_1) - b(h_2), v)_H \end{aligned}$$

Since H is the pivot space between V and V', the inner product on H defines an isomorphism J that associates every $v \in V$ with unique element $Jv \in V'$. Choosing $v \in V$ so that $Jv = w$, then

$$(b(h_1) - b(h_2), v)_V = \langle b(h_1) - b(h_2), Jv\rangle_{V'\times V} = (b(h_1) - b(h_2), Jv)_H.$$

Therefore

$$(b(h_1) - b(h_2), v)_V = (b(h_1) - b(h_2), w)_H,$$

which is

$$\langle A(h_1) - A(h_2), v\rangle_{V'\times V} = (b(h_1) - b(h_2), w)_H - (b(h_1) - b(h_2), w)_H.$$

Then

$$\langle \partial_t w(t), v\rangle_{V'\times V} - (b(h_1) - b(h_2), w)_H = (b(h_1) - b(h_2), v)_H$$

and

$$\begin{aligned}\langle \partial_t w(t), J^{-1}w\rangle_{V'\times V} + C_0||w||_H^2 &\leq C_1 \int_\Omega |w(t)||J^{-1}w(x)|dx \\ &\leq \frac{1}{2}C_0||w||_H^2 + C_2||J^{-1}w(x)||_H^2.\end{aligned}$$

But

$$\langle \partial_t w(t), J^{-1}w\rangle_{V'\times V} = (\partial_t w(t), w)_V' = \frac{1}{2}\frac{d}{dt}||w(t)||_{V'}^2$$

and

$$||J^{-1}w(x)||_H^2 = (J^{-1}w(x), J^{-1}w(x))_H = (w, w)_{V'}.$$

Then

$$\frac{d}{dt}||w(t)_{V'}^2 \leq 2C_2||w(t)||_{V'}^2 \qquad w(0) = 0,$$

which implies that

$$||w(t)||_{V'} = 0,$$

and the solution is unique.

Therefore, if

$$\begin{aligned}&f \in \mathcal{C}[0, T : H^{-1}(\Omega)] \\ &0 < C_0 \leq a(u) \leq C_2 \text{ for all } u \in \mathbb{R} \\ &\text{ and } |a(h_1) - a(h_2)| \leq K|h_1 - h_2|,\end{aligned}$$

then the initial value problem (A.1) has a unique weak solution denoted by u with the following properties:

$$\begin{aligned}u(x,t) &\in L^2[0, T : H^1(\Omega)] \cap \mathcal{C}[0, T : L^2(\Omega)] \text{ and} \\ \partial_t u(x,t) &\in L^2[0, T : H^{-1}(\Omega)].\end{aligned}$$

Appendix B

DATA GENERATION

Originally, the PDE (4.2) was made discrete in space using a finite difference scheme and was integrated in time using explicit methods. This time integration was quickly switched to implicit methods, in which Matlab's ODE suite of solvers were used. The goal of this work was to explore the possibilities of an implementation, rather then focus on the numerical analysis aspects of the implementation. To this end, the numerical schemes were chosen for their simplicity to implement, while providing access to algorithmic detail.

Matlab 6.1 ODE suite allows a large number of time integrators to be called, all with very similar syntax. The suite includes the solvers: `ODE23, ODE113, ODE15S, ODE23S, ODE23T, ODE23TB , ODE45`. All were tested on a variety of problems, although the stiff solvers (recognized by the inclusion of an `S` in their name) seemed to outperform the others for the selected coefficients.

Matlab was used to numerically generate data sets for recovery. Matlab scripts were provided with known time dependent coefficient and boundary data. This information, as well of course the computed values of the corresponding observations in time, were written to a file. The number of data points were allowed to vary, as was number of spatial and temporal grid points.

An small sample data set is provided below.

```
% Generated on 26-Mar-2004
% using table entry 16

% First row contains D0
%   t        f        g        h
   1.625      NaN      NaN      NaN
0.000000 0.000000 -0.000000 0.000000
0.250000 0.250000 0.623279 0.050643
0.500000 0.500000 0.663078 0.202868
0.750000 0.750000 0.555244 0.366781
1.000000 1.000000 0.539687 0.496226


% D{2} = 1-.5*(atan(6*(x-0.5)))
% full table entry = {2,1,ode15s,5,10,[0,0,0,0,0,0]}
```

Notice that the initial value of the coefficient is provided in the $(1, 1)$ entry, as is $f(t)$, the forcing at $x = 1$, and g and h. Also the actual coefficient $D2 = 1 - .5 * (\arctan(6 * (x - 0.5)))$ used in the forward experiment is included with the data, but the inclusion of the % indicated that this is a comment, and will invisible in the recovery phase. A call-out table was used to generate as large number of such data sets, and the actual table entry is also included and commented.

This method allowed the precision of the data to easily controlled. This example recorded six significant figures, while other data sets registered more digits and other registered fewer.

The individual entries in the table entry,

```
% full table entry = {2,1,ode15s,5,10,[0,0,0,0,0,0]}
```

represent, respectively; the coefficient to be used, the forcing function, the time integration method, the number of time nodes, number of space nodes, and an error vector.

The error vector allowed control of perturbation in $f(t), g(t)$ and $h(t)$, and is grouped in pairs, which are passed as inputs into the following script,

```
function px = perturb(x,err);

M = length(x);

% err(1) is the random (uniform) error
% err(2) is systematic error in measure

px =  x.*( 1 + (rand(M,1)-.5+err(2))*err(1) );
```

which introduces perturbation.

Appendix C

MATLAB PSEUDO CODE

Pseudo code is provided for the 2 parameter identification algorithm.

```
function [C,K] = Coeff_Inverse();

% Try to recover the coefficient C(h)
% in the model C(h) h_t = (K(h)h_z)_z
% given experimental output

[T,P,Q,C0,K0] = load Data.file;

%  T = time
%  P = h(0,t)           ; state at z = 0
%  Q = D(h)(h_z-1) @   z=L ; ie flux  at z=L
% C0 = C(h(0,0))        ; initial coefficient
% K0 = K(h(0,0))        ; initial coefficient

% Begin Recovery

C = C0;       % Initial approximation
K = K0;       % Initial approximation
level = 1;    % Initial time level

while (max(t) < Tmax)

   % Initialize and set problem specs
   % includes problem, ic's, bc's, methods
   forward.info = ...

   for strip = level-1:level
      [t,h,h_z,h_t,p,q] = solve_forward(forward.info,C,K);
```

```
end

deltap = P-p;
deltaq = Q-q;

% Assign theta (dual data)
theta.one = ???;
theta.two = ???;

% Assign Dual operators
Dual.C = ???;
Dual.K = ???;

% Set problem specs
% includes problem, ic’s, bc’s, methods
dual.info = ...

for strip = level-1:level
   [phi,phi_z] = solve_dual(dual.info,Dual);
   [psi,psi_z] = solve_dual(dual.info,Dual);
end

for strip = level-1:level
   % Construct a 2D region and basis element
   [Omega,Lambda] = Active_region(h);

   % Construct Matrix M
   M(1,1) = Integrate((h_z-1).*phi_z.*Lambda, Omega );
   M(1,2) = Integrate(h_t.*phi.*Lambda, Omega );
   M(2,1) = Integrate((h_z-1).*psi_z.*Lambda, Omega );
   M(2,2) = Integrate(h_t.*psi*Lambda, Omega );

   % Construct vector b
   b(1) = Integrate(deltap.*theta.one, t );
   b(2) = Integrate(deltaq.*theta.two, t );
end

% Compute update
delta = M \ b;
deltaK = delta(1);
deltaC = delta(2);
```

```
   % Modify coefficients C and K
   C = C + deltaC;
   K = K + deltaK;

   % Iteration is possible
   if (some condition is met)
       level = level + 1;
   end

end;
```

Appendix D

MAPLE CODE

Maple was used to compute fourier solutions to the forward problem in order to verify numerical methods in Matlab.

```
#Fourier Series solution to heat equation
# with temp and flux boundary conditiond
#
#u_t   =  nu  u_xx                     0<x<L, t> 0:
#u(x,0) = u0(x)                  0<x<L:
#u_x(0,t) = theta1(t)     u (L,t) = theta2(t)      t>0:

restart:
L := 3:
nu := 2:
u0 := x -> 0:

# flux at top    (x = 0):
theta1 := t -> 1;

# state at bottom (x = L):
theta2 := t -> 0;

# Eigenvalues
mu := n -> (2*n-1)*Pi/(2*L):

# Shift function to generate homogenous BCs
S := (x,t) -> theta1(t)*(x-L) + theta2(t):
Sp := (x,t) -> diff(S(x,t),t):

Smode := (t,n) -> 2/L*int(S(x,t)*cos(mu(n)*x),x = 0..L):
```

```
Spmode := (t,n) -> 2/L*int(Sp(x,t)*cos(mu(n)*x),x = 0..L):
w0 := n -> 2/L*int((u0(x)-S(x,0))*cos(mu(n)*x),x = 0..L):
solh := (x,t,N) ->
         sum(cos(mu(n)*x)*w0(n)*exp(-(nu*mu(n)^2)*t),n=1..N):

solp := (x,t,N) ->
           sum(
              cos(mu(n)*x) *
              int(exp(-(nu*mu(n)^2)*(t-tau))*
              Spmode(tau,n),tau = 0 ..t
              ),n=1..N):

u := (x,t,N) -> solh(x,t,N) + solp(x,t,N) + S(x,t):

ux := (x,t,N) -> diff(u(x,t,N),x):

# Flux at x = L:
g := (t,N) -> nu*(subs(x=L,ux(x,t,N)) - 1):

# state at x = 0:
h := (t,N) -> subs(x=0,u(x,t,N)):

ut := (x,t,N) -> diff(u(x,t,N),t):

# Now make  phi dual solution by  t ->  T- t:

phi := (x,t,N,i) -> u(x,2^(-i)-t,N,i):
phix := (x,t,N,i) ->  diff(phi(x,t,N,i),x):

# Now solve psi dual problem:

# Functions for psi dual problem:
psi0 := x -> 0:
t1 := t -> 0:  # flux at top    (x = 0):
t2 := t -> -t: # state at bottom (x = L):

# Shift function
S1 := (x,t) -> t2(t):
S1p := (x,t) -> diff(S1(x,t),t):
S1mode := (t,n) -> t2(t)*2/L*int(cos(mu(n)*x),x = 0..L):
S1pmode := (t,n) -> diff(t2(t),t) *
                         2/L*int(cos(mu(n)*x),x = 0..L):
```

```
solp1 := (x,t,N) -> sum(
                     cos(mu(n)*x) *
                     int(exp(-(nu*mu(n)^2)*(t-tau)) *
                     S1pmode(tau,n),tau = 0 ..t
                     ),n=1..N):

psi := (x,t,N) -> solp1(x,t,N) + S1(x,t):
psix := (x,t,N) -> diff(psi(x,t,N),x):

#Now compute rates for enties in matrix
m11 := (i,N) -> (Int(
                     Int(
                        (ux(x,t,N)-1)*phix(x,t,N,i)
                     ,x=0..L)
                 ,t=0..2^(-i))):

m21 := (i,N) -> (Int(
                     Int(
                        (ux(x,t,N)-1)*psix(x,2^(-i)-t,N)
                     ,t = 0..2^(-i))
                 ,x=0..L)):

m12 := (i,N) -> (Int(
                      Int(
                         ut(x,t,N)*phi(x,t,N,i)
                      ,x=0..L)
                 ,t=0..2^(-i))):

m22 := (i,N) -> (Int(
                     Int(
                         ut(x,t,N)*psi(x,2^(-i)-t,N)
                     ,x=0..L)
                 ,t=0..2^(-i))):

b1 := (i,N) -> Int(
                   g(t,N)*(t2(2^(-i)-t))
               ,t=0..2^(-i)):
b2 := (i,N) -> Int(
                   h(t,N)*(theta1(2^(-i)-t))
               ,t=0..2^(-i)):

N := 10:
```

```
M := 5;

for i from 1 to M do:
 m1[i] := evalf(m11(i,N)):
 m2[i] := evalf(m12(i,N)):
 m3[i] := evalf(m21(i,N)):
 m4[i] := evalf(m22(i,N)):
 B1[i] := evalf(b1(i,N)):
 B2[i] := evalf(b2(i,N)):
 detM[i] := m1[i]*m4[i]-m2[i]*m3[i]:
 TOP1[i] := B1[i]*m4[i] - B2[i]*m2[i]:
 TOP2[i]:= m1[i]*B2[i] - m3[i]*B1[i]:
 DK[i] := TOP1[i]/detM[i]:
 DC[i] := TOP2[i]/detM[i]:
 print(m3):
 save m1,m2,m3,m4,B1,B2,detM,TOP1,TOP2,DK,DC, "data_maple.m":
od:

m1list := convert([seq(eval(m1[i]),i=1..M)],string):
m2list := convert([seq(eval(m2[i]),i=1..M)],string):
m3list := convert([seq(eval(m3[i]),i=1..M)],string):
m4list := convert([seq(eval(m4[i]),i=1..M)],string):

b1list := convert([seq(eval(B1[i]),i=1..M)],string):
b2list := convert([seq(eval(B2[i]),i=1..M)],string):

detMlist := convert([seq(eval(detM[i]),i=1..M)],string):

T1list := convert([seq(eval(TOP1[i]),i=1..M)],string):
T2list := convert([seq(eval(TOP2[i]),i=1..M)],string):

dClist := convert([seq(eval(DC[i]),i=1..M)],string):
dKlist := convert([seq(eval(DK[i]),i=1..M)],string):

fid:= fopen("jimmy_data_maple.m",WRITE):
fprintf(fid,"%% Generated by MAPLE \n"):
fprintf(fid,"%% computes sequence of integrals \n"):
fprintf(fid,"%% for two parameter estimation \n"):
fprintf(fid,"%% iint = int_0^L int_0^t \n"):
fprintf(fid,"%% int = int_0^t \n"):
fprintf(fid,"%% m11 = iint (h_x-1) phi_x \n"):
fprintf(fid,"%% m21 = iint (h_x-1) psi_x \n"):
```

```
fprintf(fid,"%% m12 = iint h_t phi \n"):
fprintf(fid,"%% m22 = iint h_t psi \n"):
fprintf(fid,"%% b1 = int g t2(t-tau) dtau \n"):
fprintf(fid,"%% b2 = int h theta1(t-tau) dtau \n"):
fprintf(fid,"%% with t = 2^(-i) with i the index \n"):

fprintf(fid," \n"):
fprintf(fid," \n"):

fprintf(fid," m11 = %s ; \n",m1list):
fprintf(fid," m12 = %s ; \n",m2list):
fprintf(fid," m21 = %s ; \n",m3list):
fprintf(fid," m22 = %s ; \n",m4list):

fprintf(fid," \n"):

fprintf(fid," b1 = %s ; \n",b1list):
fprintf(fid," b2 = %s ; \n",b2list):

fprintf(fid," \n"):

fprintf(fid," top1 = %s ; \n",T1list):
fprintf(fid," top2 = %s ; \n",T2list):

fprintf(fid," \n"):

fprintf(fid," detM = %s ; \n",detMlist):

fprintf(fid," \n"):

fprintf(fid," dK = %s ; \n",dKlist):
fprintf(fid," dC = %s ; \n",dClist):

fclose(fid):
```

Appendix E

VAN GENUCHTEN PARAMETERS

In this section, a table of van Genuchten parameters is provided for reference.

The van Genuchten water content function $\Theta(h)$ is given by

$$\Theta(h) = \Theta_r + \frac{\Theta_s - \Theta_r}{[1 + (\alpha h)^n]^{1/n-1}},$$

which can be rewritten to represent the relative saturation S_e,

$$S_e(h) = \frac{\Theta(h) - \Theta_r}{\Theta_s - \Theta_r},$$

which is in turn used to construct the hydraulic conductivity function

$$K(h) = K_0 S_e(h)^\ell \left\{1 - [1 - S_e(h)^{n/(n-1)}]^{1-1/n}\right\}^2.$$

We recall $C(h)$ to be the derivative of $\Theta(h)$. We note that the van Genuchten formulation couples Θ with K. Here Θ_r and Θ_s are the residual and saturated water contents, respectively. Both parameters α and n are used to control the shape of the curve $\Theta(h)$. K_0 is the matching point at saturation, and need not be equal to K_s, the saturated conductivity. Finally, ℓ is considered to be a measure of soil pore connectivity, and is normally taken to be 1/2. Hydrus allows ℓ to assume negative, unphysical values, as this practice has been shown to lead to better results.

Average Class values for van Genuchten parameters

Class	N	Θ_r	Θ_s	α	n	K_s	K_0	ℓ
		cm^3/cm^3	cm^3/cm^3	$1/cm$		cm/day	cm/day	
Clay	84	0.098	0.459	0.015	1.253	14.757	2.965	-1.561
C Loam	140	0.079	0.442	0.016	1.416	8.185	5.000	-0.763
Loam	242	0.061	0.399	0.011	1.472	12.050	3.698	-0.371
L Sand	201	0.049	0.390	0.035	1.746	105.196	24.322	-0.874
Sand	308	0.053	0.375	0.035	3.177	642.688	24.491	-0.930
S Clay	11	0.117	0.385	0.033	1.208	11.350	4.335	-3.665
S C L	87	0.063	0.384	0.021	1.330	13.183	6.934	-1.280
S Loam	476	0.039	0.387	0.027	1.449	38.282	15.488	-0.861
Silt	6	0.050	0.489	0.007	1.679	43.752	3.342	0.624
Si Clay	28	0.111	0.481	0.016	1.321	9.616	3.170	-1.287
Si C L	172	0.090	0.482	0.008	1.521	11.117	2.234	-0.156
Si Loam	330	0.065	0.439	0.005	1.663	18.239	1.750	0.365