

ABSTRACT

GIFFEN, DEENA HANNOUN. Simulating Non-Dilute Transport in Porous Media Using a Thermodynamically Constrained Averaging Theory-Based Model. (Under the direction of Ralph C. Smith.)

In this dissertation, we model non-dilute transport in porous media using a thermodynamically constrained averaging theory-based model. We employ the equations from [23], reformulate them in a robust form, and add closure relations and boundary and initial data so that the model can be simulated numerically. We then perform frequentist parameter estimation using nonlinear least squares to calibrate the model to experimental data. We construct a cubic spline based surrogate model, which drastically reduces simulation times and allows us to perform Bayesian parameter estimation. Finally, we use a simplified version of the model, which is independent of initial concentration, to perform mixed-effects simulations from both frequentist and Bayesian perspectives.

© Copyright 2017 by Deena Hannoun Giffen

All Rights Reserved

Simulating Non-Dilute Transport in Porous Media Using a Thermodynamically
Constrained Averaging Theory-Based Model

by
Deena Hannoun Giffen

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2017

APPROVED BY:

Alina Chertock

Brian Reich

Pierre Gremaud

Ralph C. Smith
Chair of Advisory Committee

DEDICATION

To Rory. I hope your love of math and science takes you far!

BIOGRAPHY

Deena Hannoun Giffen (née Deena Hannoun) was born in Pasadena, California and grew up in San Diego, California and McGaheysville, Virginia. She later studied mathematics and English literature at James Madison University. Deena worked as a Software Analyst for one year before deciding to continue her mathematical studies at North Carolina State University. She is currently employed as the Limnology Modeling Project Manager at the Southern Nevada Water Authority, where she manages a 3-D hydrodynamic and water quality code to assess the impact of the ongoing drought on the Lower Colorado River Basin.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my adviser, Dr. Ralph Smith, for his guidance. I would not be the mathematician I am today without your influence.

I would also like to acknowledge NSF and the Army Research Office. Thank you for providing financial support for this dissertation.

I would also like to thank my husband, Nicholas Giffen. Your unwavering support and encouragement has been invaluable as I go through this process.

My lifelong friend Caroline Kigans. Thanks for keeping me sane and making me laugh during this whole process!

My parents, Imad and Betty Hannoun, and my brother Kareem. You are the best family anyone could ever hope for.

My brother in laws Alex, Matthew, and Jake and sister in law Kelsey. I love how close we've all become, and I wouldn't trade y'all in for anything.

Alina, Brian, and Pierre. Thank you for serving on my committee and for providing feedback to make this dissertation better than I thought it could be.

My officemates Anna Fregosi and Anne Costolanski. Thanks for always being there for me.

My friends, especially my running buddy Jessica Tisdale Knouse (and new running buddy Ellis Knouse), Nick's lifelong best friend Bill Knouse, and Steve, Trish, little Steven, and Caiden Turner. We always have way too much fun together.

Todd, my boss at the Water Authority. Thank you for encouraging me to finish even after I started working.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
Chapter 1 Introduction	1
1.1 Contributions	2
1.2 Problem Description	2
1.2.1 Porous Media Flow and Species Transport	3
1.2.2 Prior Work	5
1.2.3 Deficiencies in the Traditional Model and Open Questions	8
1.3 Motivation for TCAT	8
1.4 Goals and Objectives	9
Chapter 2 Derivation of the TCAT Model	10
2.1 The Representative Elementary Volume	10
2.2 Microscale Conservation Equations	11
2.3 Averaging	13
2.3.1 Averaging Benefits and Basics	13
2.3.2 Averaging from the Microscale to the Macroscale	14
2.4 The TCAT Model	15
2.5 Determination of Closure Relations	20
2.6 Initial and Boundary Conditions	20
2.7 Linearized Version of the TCAT Model	22
2.7.1 Simplifying Assumptions	22
2.7.2 Applying Simplifying Assumptions to TCAT	24
2.7.3 Analytic Solution	26
2.8 Mathematical Challenges	28
Chapter 3 Mathematical Simulation of the TCAT Model	30
3.1 The TCAT Model	30
3.2 Differential-Algebraic Equations	33
3.3 Transforming the TCAT Model	33
3.4 The Method of Lines	34
3.4.1 Finite Difference Schemes	35
3.4.2 Conservation Form	36
3.5 Time Integration in MATLAB	36
3.5.1 Numerical Verification of Index	37
3.5.2 The Methodology of ode15s	37
3.5.3 Differential-Algebraic Equation Consistency	40

3.6	Sparse Jacobian Pattern	40
3.7	Solver Settings	42
3.8	Modifications to ode15s	43
3.9	Adaptive Gridding Technique and Grid Convergence	44
3.9.1	Spatially Adaptive Method of Lines	44
3.9.2	Grid Refinement Study	45
Chapter 4	Parameter Estimation	50
4.1	The Levenberg-Marquardt Algorithm	50
4.2	Tracer Parameter Estimation	52
4.3	High Concentration Parameter Estimation	53
4.3.1	Global Estimation for $\log(c_1)$	54
4.3.2	Forward Simulations for Global $\log(c_1)$	55
4.3.3	Individual Parameter Estimation	57
4.4	Conclusions	59
Chapter 5	Surrogate Modeling	61
5.1	Justification for Constructing a Surrogate Model	61
5.2	Surrogate Model Classification	63
5.3	Regression-Based Model Construction	64
5.4	Surrogate Example	65
5.5	Constructing a Surrogate Model for TCAT	67
5.6	Evaluating the Surrogate Model Over Time	70
5.7	Surrogate Models for Additional Experiments	71
5.8	Surrogate Model and Bayesian Inference	73
Chapter 6	Bayesian Inference	74
6.1	Statistical Modeling	74
6.2	Frequentist and Bayesian Approaches	75
6.3	Bayes' Relation	76
6.4	Markov Chain Monte Carlo (MCMC) and the Metropolis Algorithm	77
6.5	Convergence	80
6.6	Implementation: DRAM	82
6.7	DRAM Applied to the TCAT Model	85
Chapter 7	Mixed-Effects Modeling	89
7.1	Motivation	89
7.2	Linear Mixed-Effects Models	91
7.3	Mixed-Effects Applied to TCAT	95
7.4	Frequentist Mixed-Effects Parameter Estimation	96
7.5	Bayesian Parameter Estimation: DRAM	98
7.6	Comparison of Bayesian and Frequentist Parameter Estimation Results	102

7.7 Comparison of Mixed-Effects and Simplified TCAT Models	103
Chapter 8 Conclusions and Future Work	105
BIBLIOGRAPHY	107

LIST OF TABLES

Table 3.1	Grid refinement study results from self-estimation.	47
Table 3.2	Grid refinement study results from the reference solution.	48
Table 4.1	Effective dispersion estimations for the three different tracers.	53
Table 4.2	Experiments considered for the global parameter estimation.	54
Table 4.3	Selected experiments for the forward simulations.	55
Table 4.4	Optimized parameter values for selected experiments.	58
Table 5.1	Information criterion comparison for the (5.6).	66
Table 5.2	Coefficients of the second-order response surface model for (5.6).	67
Table 5.3	Information criterion comparison for the TCAT model.	68
Table 5.4	Coefficients of the second-order response surface model for TCAT.	69
Table 5.5	Root mean square errors for different time	71
Table 5.6	Errors from surrogate modeling.	72
Table 6.1	Experiments considered for DRAM simulations.	86
Table 6.2	Summary statistics from DRAM for the four considered experiments.	88
Table 7.1	Experiments considered for mixed-effects modeling.	96
Table 7.2	Random effects from <code>nlmefit</code>	97
Table 7.3	The residual norms from <code>nlmefit</code>	99
Table 7.4	The random effects from DRAM.	99
Table 7.5	The summary statistics from DRAM applied to the hyperparameters.	100
Table 7.6	The residual norms from mixed-effects DRAM.	101
Table 7.7	Estimated parameter values for TCAT from <code>nlmefit</code> and DRAM.	103
Table 7.8	Comparison of mixed-effects model to the simplified TCAT model.	104

LIST OF FIGURES

Figure 1.1	Experimental setup for the porous media system.	4
Figure 1.2	Comparison of high and low concentration breakthrough curves.	5
Figure 3.1	Results of the self-estimation grid study in three different norms.	48
Figure 3.2	Results of the reference solution grid study in three different norms. . . .	48
Figure 4.1	Model and experimental breakthrough curves for a tracer experiment. . .	53
Figure 4.2	Breakthrough curves for the three selected experiments, B11, B7, and B14.	55
Figure 4.3	Forward simulations of breakthrough curves for Case One, where there is fresh water in the column at the start of the experiment.	56
Figure 4.4	Forward simulations of breakthrough curves for Case Two, where there is a resident brine in the column at the start of the experiment.	57
Figure 4.5	Individually estimated breakthrough curves for Case One.	59
Figure 4.6	Individually estimated breakthrough curves for Case Two.	60
Figure 5.1	Response surface model and test points for (5.6).	67
Figure 5.2	Polynomial response surface model and test points for TCAT.	69
Figure 5.3	Cubic spline response surface model and test points for TCAT.	70
Figure 5.4	Surrogate breakthrough curves for varying $\log(c_1)$	71
Figure 5.5	Surrogate models compared to test points at breakthrough.	72
Figure 6.1	Error distributions for the four considered experiments.	75
Figure 6.2	(a) Burn-in period for the chain and (b) chain that has converged.	81
Figure 6.3	MCMC chains for the four considered experiments.	87
Figure 6.4	Kernel density estimates for the four considered experiments.	88
Figure 7.1	Measurement heights (cm) for 26 boys on 9 occasions [20].	90
Figure 7.2	Random effects plotted by subject.	94
Figure 7.3	Model fit and residuals for the Oxford boys growth model.	94
Figure 7.4	Mixed-effects model breakthrough curves.	97
Figure 7.5	Mixed-effects responses compared to data.	98
Figure 7.6	DRAM results for hyperparameters.	100
Figure 7.7	DRAM results for hyperparameters, continued.	101
Figure 7.8	Mixed-effects responses compared to data.	102

LIST OF SYMBOLS

$\hat{\gamma}_{i\alpha}$	Macroscale activity coefficient
$\hat{\mu}^a$	Viscosity
$\theta^{\bar{a}}$	Entropy weighted macroscale temperature
ρ_α	Microscale density
ρ^a	Macroscale density
$\omega_{i\alpha}$	Microscale mass fraction
$\omega^{i\bar{a}}$	Macroscale mass fraction
A, B	Species qualifiers
\mathbf{d}_α	Microscale rate of strain tensor
$\mathbf{d}^{\bar{a}}$	Macroscale rate of strain tensor
$\hat{\mathbf{D}}$	Effective dispersion tensor
\mathbf{g}_α	Microscale gravity vector
$\mathbf{g}^{\bar{a}}$	Macroscale gravity vector
i	General index for a species
\mathbf{I}	Identity tensor
J	Dispersive mass flux
$M W_i$	Molecular weight of species i
$\epsilon^{\bar{w}}$	Volume fraction of the wetted phase
$\epsilon^{\bar{w}} \nu^{\bar{w}}$	Darcy velocity
$r_{i\alpha}$	Microscale reaction term
$\hat{\mathbf{R}}$	Tensor closure coefficients
\mathbf{t}_α	Microscale stress tensor
$\mathbf{t}^{\bar{a}}$	Macroscale stress tensor
$\mathbf{T}_{\alpha \rightarrow \beta}$	Microscale transfer of momentum
$\mathbf{T}^{\alpha \rightarrow \beta}$	Macroscale transfer of momentum
$\mathbf{u}_{i\alpha}$	Microscale dispersion velocity
$\mathbf{u}^{\bar{i}\bar{a}}$	Macroscale dispersion velocity

\mathbf{v}_α	Microscale velocity
$\mathbf{v}^{\bar{\alpha}}$	Macroscale velocity
\bar{V}	Partial mass volume
w	Entity index for wetted phase
$x_{i\alpha}$	Microscale mole fraction
$x^{\bar{\bar{i}\alpha}}$	Macroscale mole fraction
\mathbf{X}_{is}	Microscale partial mass tensor
$\mathbf{X}^{\bar{\bar{i}s}}$	Macroscale pressure weighted partial mass tensor
\bar{w}	One line above a superscript denotes a density weighted macroscale average
$\bar{\bar{w}}$	Two lines above a superscript denotes a uniquely weighted macroscale average

CHAPTER

1

INTRODUCTION

In this dissertation, we consider single-fluid phase flow in porous media. While the traditional model is well-established [6, 22], it has some deficiencies that warrant the development of alternative formulations. In 2009, Miller and Gray published a paper [23] outlining a new model based on thermodynamically constrained averaging theory (TCAT). In this dissertation, we will reformulate the equations presented in Miller and Gray's 2009 paper, and perform parameter estimation using experimental data and a nonlinear least squares algorithm. Next, because of prohibitively long run times, we construct a regression-based surrogate model for TCAT using a cubic spline. We will then utilize the surrogate model to perform parameter estimation from a Bayesian perspective, which will allow us to determine a distribution for our parameter of interest. Finally, we will present a mixed-effects model from both a frequentist and Bayesian perspective with a simplified form of the TCAT model.

1.1 Contributions

- We reformulated the TCAT model as a partial differential algebraic equation (PDAE). This yields an index-1 PDAE consisting of one PDE that describes flow and one algebraic constraint. After we apply the method of lines to the PDAE, we arrive at an ordinary DAE. Our robust form allows us to use an available DAE solver to perform fully-coupled model simulations.
- We used our code to investigate closure relations in the model and determined that there should be a single unknown parameter c_1 in the high concentration model. This is the first time this model has successfully been simulated in 1-D at high concentrations.
- We performed parameter estimation for the single parameter model with MATLAB's Levenberg-Marquardt code `lsqnonlin`. Whereas we were able to simulate a wide range of incoming concentrations, we determined that the lower concentration experiments provided a better fit to data.
- We constructed a regression-based surrogate model for the TCAT model. The surrogate model uses cubic splines to emulate the behavior of the TCAT model. The surrogate reduces run times by 4-5 orders of magnitude, making the computational demands of Bayesian inference possible for TCAT.
- We used Bayesian inference to further investigate the TCAT model and determine distributions for our parameter.
- We performed mixed-effects simulations on the simplified TCAT model from both frequentist and Bayesian perspectives. This allowed us to quantify the fixed and random effects of the model.

1.2 Problem Description

In this section, we will detail the problem under investigation and provide some background on single-phase flow in porous media systems.

A porous media system is a domain that is occupied by a persistent solid phase, also called a solid matrix. The part of the domain that is not occupied by the solid phase is

called the pore space. The pore space can be occupied by a single-fluid phase or by several immiscible fluids [40]. Many porous media systems occur naturally. Some examples of porous media are soil, sand, bread, kidneys, and lungs [5]. In this work, we consider a single-fluid phase flow problem, where one fluid displaces another within the porous media system.

We consider the brine invasion problem, where a calcium bromide solution of known concentration is pumped into the bottom of a vertical column of porous media—in this case, sand, and freshwater. The solution is pumped into the column at a controlled inflow rate and is allowed to move through the column. At the top of the column, one measures the relative concentration of the exiting brine and records the time. When we consider concentration, we must also consider the concept of the mass fraction of a solution. The mass fraction of a solution is the fractional proportion of the mass of a substance, in this case, calcium bromide. Mass fraction ranges between 0 for fresh water and 1 for a completely saturated solution. Density times mass fraction yields concentration [5].

Experiments were conducted at the University of North Carolina with different incoming mass fractions ranging from 10^{-6} to 0.4. A second set of experiments was also generated where there is brine in the column at time $t = 0$ and a higher concentration brine is pumped into the bottom of the column.

Figure 1.1 illustrates the experimental setup in the lab.

1.2.1 Porous Media Flow and Species Transport

Here, we discuss and model species transport in porous media. We will investigate the mass fraction distribution of a chemical species α within a macroscopic porous media system. In this system, the chemical species is affected by both advection and diffusion. Advection describes the movement of a species in a system. Diffusion describes the tendency of chemical species to move from higher concentration areas to lower concentration areas [34].

Numerous research has been conducted on the topic of species transport in porous media. In this dissertation, we consider stable displacement, where a variable density brine solution is pumped into the bottom of a column packed with porous media [80]. Many researchers have conducted such experiments, including [32, 67, 76, 77].

We will determine the differences between high concentration species transport and low concentration species transport. One notable difference between the two types of transport

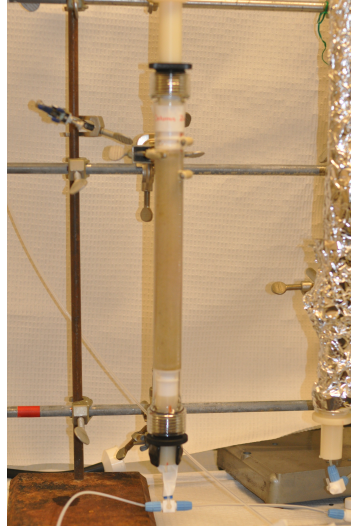


Figure 1.1: Experimental setup for the porous media system.

is hydrodynamic dispersion. Hydrodynamic dispersion is a type of diffusion and is defined as “the tendency of a fluid to spread out from the path that it would be expected to follow according to the advective hydraulics of a flow system” [61].

Schotting, Moser, and Hassanizadeh [67] performed high concentration intrusion experiments and determined that hydrodynamic dispersion is not constant, and instead depends on the discharge rate of fluid from the column. Later, Watson and Barry [76] determined that there is a physical difference between hydrodynamic dispersion for low concentration versus high concentration solutions. They found that there is less dispersion in high concentration displacement than in low concentration displacement, regardless of flow rate.

Watson and Barry’s findings [76] show that the physics of the low concentration displacement problem are different than the physics in a higher concentration system. The experimental data used in this dissertation also verifies this conclusion. Both high and low concentration stable displacement experiments were conducted, and the data collected in the lab includes time and relative concentration at the top of the column. With this data, we can create a breakthrough curve, which shows the change in relative concentration as the brine exits the column. Figure 1.2 shows two example breakthrough curves from data collected in the lab. The plot is fixed in space at the top of the column. The x -axis is time, and the y -axis is relative concentration. We first note a relative concentration of zero as the

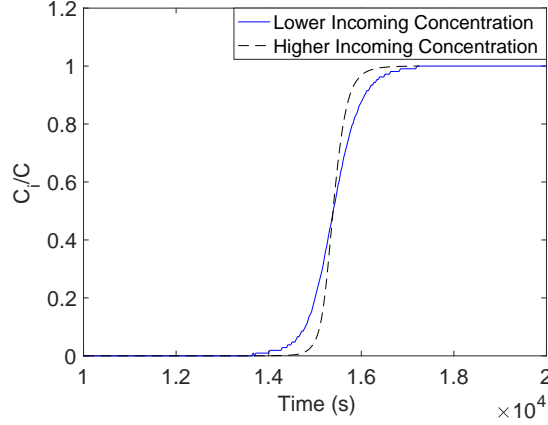


Figure 1.2: Comparison of high and low concentration breakthrough curves.

brine travels to the top of the column. Next, we see a breakthrough curve emerge as the brine displaces the fluid in the column and reaches the top of the column. Finally, we see steady state flow with a relative concentration of 1 as the front continues to pass through the column and the initial fluid has been completely flushed out of the column.

The solid blue breakthrough curve is data collected from a lower concentration experiment, and the black dashed breakthrough curve is data collected from a much higher concentration experiment. Both experiments were performed on identical columns, and we note that there is a stark difference in the pitch of the curves. The high concentration breakthrough curve is much steeper than its lower concentration counterpart. These observations lead us to conclude that the physics of high and low concentration transport are different.

1.2.2 Prior Work

Here, we provide an overview of models that have attempted to solve the species transport problem. The macroscale partial differential equations for 1-D brine flow are formulated as [77]

$$\phi \frac{\partial}{\partial t} \rho = -\frac{\partial}{\partial z} (\rho q) \quad (1.1)$$

$$\phi \frac{\partial}{\partial t} (\rho \omega) = -\frac{\partial}{\partial z} (\rho \omega q) - \frac{\partial J}{\partial z} \quad (1.2)$$

$$q = \frac{-k}{\mu} \left(\frac{\partial p}{\partial z} + \rho g \right). \quad (1.3)$$

Here ρ is the density, ω is the mass fraction, ϕ is the porosity, q is the Darcy velocity, and J is the dispersive mass flux. We have used simplified notation because these quantities are not averaged from the microscale as in the TCAT approach.

The traditional way to model density-dependent transport is Fick's Law. Fick's Law is a linear closure for the dispersive mass flux, where

$$J = -\rho \phi D \frac{\partial}{\partial z} \omega. \quad (1.4)$$

It is well established that Fick's Law is inadequate to describe both low and high concentration displacement [23, 27, 32, 67, 76, 77]. This motivates the development of alternative models.

In 1988, Hassanizadeh and Leijnse [31] introduced extended forms of Darcy's Law and Fick's Law in an attempt to remedy deficiencies in the model. The updated forms were given by

$$q = \frac{-k}{\mu} \left(\frac{\partial p}{\partial z} + \rho g \right) - D^f \frac{\partial \omega}{\partial z} \quad (1.5)$$

$$J = -\rho \phi D \frac{\partial}{\partial z} \omega + \omega K^s \left(\frac{\partial p}{\partial z} - \rho g \right), \quad (1.6)$$

where D^f and K^s are coefficients. However, when these new equations were used to simulate high concentration experimental data, Hassanizadeh [30] found that they did not adequately describe the experimental results as varying values of the coefficients were needed for different incoming concentrations and flow rates.

Later, Hassanizadeh and Leijnse [32] published several alternative formulations of Fick's Law. They determined that the nonlinear closure for dispersion,

$$(1 + \beta |J|) J = -\rho \phi D \frac{\partial}{\partial z} \omega, \quad (1.7)$$

more accurately described their experimental data. Here, β is a coefficient that accounts for the nonlinear effects seen in the lab [32]. When $\beta = 0$, (1.7) simplifies to Fick's Law.

Whereas they found that (1.7) is superior to Fick's Law for high concentration displacement experiments, Hassanizadeh and Leijnse acknowledged that more general validation of their model needed to be conducted [32]. Further, their fitted value of β was much smaller than the fitted β values for a similar high concentration data set collected by Moser [76].

Watson and Barry [76] generated high quality data to investigate the findings in [32]. They also concluded that (1.7) is a better approximation for diffusion than Fick's Law. In 2002, Watson, Barry, Schotting, and Hassanizadeh [77] performed more experiments to test the validity of Darcy's Law and Fick's Law. While they again found that (1.7) describes high and low concentration displacements better than Fick's Law, they did report some variation in β under different experimental conditions, specifically flow rate. Hassanizadeh and Leijnse [32] claim that one can determine a value of β that provides a good fit to data based on flow rate for a specific medium. However, other researchers disagree and point out that experiments of different densities cannot be modeled with a unique value of β [43]. This means that one would have to fit each experiment separately, and therefore, the closures for diffusion presented in [32] are not a great improvement over Fick's Law.

Because of the failure of the model presented in [32], researchers continue to explore modeling species transport in porous media. One alternative approach, published in 2007 [43], involves using a homogenization method. The homogenization method is an upscaling technique. Egorov's model assumes that the porous media is heterogeneous at the Darcy scale. The alternate approach, proposed by Demidov, starts at the pore scale. Both methods yield a macroscale balance equation, where dispersion is a function of the density gradient [43].

While the homogenization approach provides additional insight into the modeling process, it does have some drawbacks. Egorov's model was found to yield good results for nearly homogeneous porous media. Demidov's model was effective for some heterogeneous porous media but, as the grain sizes became more widely varied, additional parameters were needed to fit the model to data [43]. Therefore, there is not a single set of parameters for either of these models that can describe a wide range of experimental data.

These shortcomings lead us to continue looking for a robust model that can describe a range of experimental data with a single set of parameters.

1.2.3 Deficiencies in the Traditional Model and Open Questions

In the previous section, we pointed out a number of examples where the traditional model fails to match lab data. In addition to these modeling issues, we must also consider how the traditional model fails to provide consistent physics. Whereas the traditional model for flow and transport in porous media is widely accepted, it has some physical inconsistencies that lead us to look at alternative models.

The TCAT model employs macroscale closure relations for advection and diffusion at all entities, which other models fail to do. The lack of these equations yields a dearth of rigorous, physics-based closure relations [24, p. 202].

One other issue is that the length scales for different elements within the system can vary greatly. For example, the grain size of the porous media, which we consider to be at the microscale, is often much smaller than the length scale of the entire porous media system, which we would typically model at the macroscale. The traditional model does not take these discrepancies into account; however, the TCAT model provides a consistent method for upscaling microscale equations to the macroscale [24, p. 9].

Further, some models neglect thermodynamics. As Gray and Miller point out, this is detrimental because one cannot then use thermodynamics to inform closure relations. Other methods use microscale thermodynamics, but this yields physical quantities that are not rigorously defined at the macroscale [24]. TCAT eliminates these issues by averaging microscale thermodynamic equations to the macroscale. Consequently, all physical quantities are well-defined at the macroscale [24, p. 21].

1.3 Motivation for TCAT

Thermodynamically constrained averaging theory (TCAT)-based modeling is a new class of models that seeks to address and solve the issues associated with the traditional model. One novel feature of the TCAT approach is that it takes pore-scale physics into account. However, since the system of interest is at the macroscale, we need a rigorous method for upscaling. TCAT provides upscaling by using averaging to transition between scales. Therefore, the result is a macroscale model that includes information from the microscale [22].

With averaging, TCAT uses information from microscale thermodynamics to close the system. The traditional model only considers diffusion due to concentration, but TCAT

takes diffusion from three quantities into account—concentration, pressure, and activity [23].

Previous research established that there is different physics in high and low concentration species transport in porous media [32, 77]. This is a very active field of research with many open questions. For example, is there a single model that can accurately describe both low and high concentration species transport? Further, how can we construct a model that describes a macroscale system but takes the physical phenomena at the pore scale into account? It is these open questions that motivate continued research on species transport.

1.4 Goals and Objectives

The goal of this research is to investigate a new class of models. Here, we will focus on thermodynamically constrained averaging theory (TCAT)-based models. The goal of this class of models is to address some of the open questions in porous media research.

In this dissertation, we begin with the TCAT model from [23]. This is a macroscale model for species transport in porous media that differs from the traditional model in several notable ways. We discuss the specific details of the TCAT model in Chapter 2. While the formulation of the TCAT model was developed by Miller and Gray [23] and is outside the scope of this dissertation, we will validate and approximate their model. Our main objective is to find a single species transport model with a unique parameter that can characterize a wide range of concentrations. While the TCAT model fails to provide an adequate fit to data in the highest concentration cases, we were able to find a single parameter that can describe experiments with a mass fraction less than 0.15. We then went beyond the traditional frequentist parameter fitting and constructed a regression-based surrogate model for TCAT that allowed us to estimate parameters from a Bayesian perspective. Finally, we applied simplifying assumptions to the TCAT model to obtain a design-independent model that we employ to quantify fixed and random effects for a set of experimental data.

CHAPTER

2

DERIVATION OF THE TCAT MODEL

In the previous chapter, we summarized the physics of how species are transported through porous media systems. In this chapter, we will derive the microscale conservation equation and average it to the macroscale to provide a basis for the TCAT model. We then complete the derivation of the TCAT model and discuss closure relations and initial and boundary conditions. Finally, we present a linearized version of the TCAT model, which we will later use when we perform our mixed-effects simulations.

2.1 The Representative Elementary Volume

In a porous media system, we use the term entity to describe the different aspects of the system. An entity can be a phase (water, solid, gas), an interface, or a common curve [18].

We also must consider the concept of a representative elementary volume. A representative elementary volume (REV) is a way to define the length scale at which physical properties are unchanging [79]. Different length scales have different REVs [24].

Definition 2.1.1. *A Representative Elementary Volume (REV), denoted by Ω , is the volume at which a physical property of a sample becomes independent of the size of the sample [2].*

Depending on the length scale under consideration, we will consider different REV's [24]. As the REV definition implies, when considering a porous media system, the length scales of different elements within the system are important. Miller and Gray [24] define five different length scales that can describe a range of natural phenomena. While the actual measurements for these scales vary depending upon the application one considers, the length scales are discrete and do not overlap. The five distinct scales, in order from smallest to largest, are the molecular scale, microscale, resolution scale, macroscale, and megascale.

For purposes of this work, we will focus on the microscale and the macroscale. The microscale is often referred to as the pore scale. Its REV is the mean distance between particle collisions. At the microscale, one would use the Navier-Stokes equations to model the system. Whereas considering the microscale is valuable when modeling species transport in a porous media system, we ultimately need a larger scale so that we have a solvable system [24].

At the macroscale, we can differentiate between the wetted and solid phases. This means that when we consider a single point within the system, we can determine which phase it is in. While we do lose detail when we go from the microscale to the macroscale, the end result is a solvable model for a large porous media system. In this context, we refer to a solvable system as one that uses the correct length scale to describe the phenomena of interest that has well-defined closure relations while being feasible to simulate numerically [24].

The identification of these length scales is important for the derivation of the TCAT model. The TCAT model provides a rigorous method for upscaling from the microscale to the macroscale. Therefore, the end result is a model that takes pore scale physics into account but is solvable [24].

2.2 Microscale Conservation Equations

Here, we construct a microscale equation for a general extensive property. In the following section, we will average this equation to the macroscale.

An extensive property is one whose value depends on the amount of matter being considered. Alternatively, an intensive property is one that is independent of the amount

of matter present. For example, mass is an extensive property, but fluid temperature and density are intensive properties [33].

The goal of a conservation equation is to study how extensive properties change in time. The conceptual conservation equation is [24]

$$\begin{aligned} &[\text{Rate of Accumulations}] + [\text{Net Outward Advective Flux}] - [\text{Body Sources}] \\ &- [\text{Non-Advective Boundary Sources}] - [\text{Rate of Generation}] = 0. \end{aligned} \quad (2.1)$$

For more details, we direct the reader to [24, 37, 63, 79]. We provide here a very brief overview of a microscale conservation equation applied to a general extensive property.

We begin at the microscale with the general point form of a species conservation equation,

$$\frac{\partial(F_{ia})}{\partial t} + \nabla \cdot (F_{ia} \mathbf{v}_{ia}) - \nabla \cdot (\mathbf{F}_{Bia}) - F_{Nia} - F_{Gia} = 0. \quad (2.2)$$

Here F_{ia} is a placeholder for the microscale quantity of interest and \mathbf{v}_{ia} denotes the microscale species velocity [24]. The terms from (2.1) and (2.2) have the following correspondence:

$$\begin{aligned} \frac{\partial(F_{ia})}{\partial t} &= \text{Rate of Accumulations} \\ \nabla \cdot (F_{ia} \mathbf{v}_{ia}) &= \text{Net Outward Advective Flux} \\ \nabla \cdot (\mathbf{F}_{Bia}) &= \text{Body Sources} \\ F_{Nia} &= \text{Non-Advective Boundary Sources} \\ F_{Gia} &= \text{Rate of Generation.} \end{aligned}$$

Equation (2.2) states that the rate of accumulation of our extensive property, such as mass or momentum, comes from five different sources. We must consider advective transport, nonadvective transport or diffusion, mass transfer, reactions, and sources or sinks [63].

We now format the equation for microscale species conservation, neglecting body sources and generation. We substitute $\rho_a \omega_{ia}$ for our placeholder F_{ia} . The resulting equation is

$$\frac{\partial \rho_a \omega_{ia}}{\partial t} = -\nabla \cdot (\rho_a \omega_{ia} \mathbf{v}_{ia}) + r_{ia}. \quad (2.3)$$

In (2.3), ρ_α is the phase density, $\omega_{i\alpha}$ is the species mass fraction, and $r_{i\alpha}$ is the reaction term.

2.3 Averaging

We discuss here how we average the TCAT model from the microscale to the macroscale.

2.3.1 Averaging Benefits and Basics

As we detailed in Chapter 1, one problem with the traditional model is that the length scales for different entities within the system can vary drastically. Unfortunately, the traditional model fails to take these length scale discrepancies into account. The TCAT model improves upon the traditional model by providing a consistent method for upscaling microscale equations to the macroscale. We achieve this upscaling from the microscale to the macroscale by using the averaging operator [24].

Consequently, our goal is to take (2.3) and average it to the macroscale. To accomplish this, we will introduce the averaging operator,

$$\langle f \rangle_{\Omega_j, \Omega_k, W} = \frac{\int_{\Omega_j} W f \, d\mathbf{r}}{\int_{\Omega_k} W \, d\mathbf{r}}. \quad (2.4)$$

Here f is the quantity we are averaging and W is the weight. Domains are represented by Ω_j and Ω_k . In many cases, $\Omega_k = \Omega$, and the quantity is scaled by the REV.

We will now summarize three common ways in which we will use the averaging operator to transform (2.3) to the macroscale. First, we have the intrinsic phase average for a general quantity f_α , which is given by

$$\langle f_\alpha \rangle_{\Omega_\alpha, \Omega_\alpha} = f^\alpha. \quad (2.5)$$

Note that in (2.5), we assume $W = 1$ (no weighting), and the third term in the subscript is consequently omitted. In (2.5), f^α is a macroscale intrinsic average for a phase α . Next, we introduce the general form of an averaging operator

$$\langle f_\alpha \rangle_{\Omega_\alpha, \Omega} = \langle f_\alpha \rangle_{\Omega_\alpha, \Omega_\alpha} \langle 1 \rangle_{\Omega_\alpha, \Omega} = \epsilon^{\bar{\alpha}} f^\alpha, \quad (2.6)$$

when it is scaled by the REV. Here $\epsilon^{\bar{\bar{a}}}$ denotes the volume fraction of the phase α . Finally, we have the mass average

$$\langle f_\alpha \rangle_{\Omega_\alpha, \Omega_\alpha, \rho_\alpha} = f^{\bar{\bar{a}}}. \quad (2.7)$$

We will now use these properties to average the microscale conservation equation, given by (2.3), to the macroscale [24].

2.3.2 Averaging from the Microscale to the Macroscale

We begin by considering an average of the microscale equation scaled by the REV,

$$\left\langle \frac{\partial \rho_\alpha \omega_{i\alpha}}{\partial t} \right\rangle_{\Omega_\alpha, \Omega} = \left\langle -\nabla \cdot (\rho_\alpha \omega_{i\alpha} \mathbf{v}_{i\alpha}) \right\rangle_{\Omega_\alpha, \Omega}. \quad (2.8)$$

For purposes of this work, we have dropped the reaction term as no chemical species are being created or destroyed. We are able to use transport and divergence theorems to move the derivatives outside of the averaging operators [24]. When we apply the averaging operator to the term on the left-hand side of (2.8), we obtain

$$\langle \rho_\alpha \omega_{i\alpha} \rangle_{\Omega_\alpha, \Omega} = \langle \omega_{i\alpha} \rangle_{\Omega_\alpha, \Omega_\alpha, \rho_\alpha} \langle \rho_\alpha \rangle_{\Omega_\alpha, \Omega_\alpha} \langle 1 \rangle_{\Omega_\alpha, \Omega} = \epsilon^{\bar{\bar{a}}} \rho^\alpha \omega^{i\bar{\bar{a}}}. \quad (2.9)$$

Application of the averaging operator to the term on the right-hand side of (2.8) yields

$$\langle \rho_\alpha \omega_{i\alpha} \mathbf{v}_{i\alpha} \rangle_{\Omega_\alpha, \Omega} = \langle \mathbf{v}_{i\alpha} \rangle_{\Omega_\alpha, \Omega_\alpha, \rho_\alpha \omega_{i\alpha}} \langle \omega_{i\alpha} \rangle_{\Omega_\alpha, \Omega_\alpha, \rho_\alpha \omega_{i\alpha}} \langle \rho_\alpha \rangle_{\Omega_\alpha, \Omega_\alpha} \langle 1 \rangle_{\Omega_\alpha, \Omega} = \epsilon^{\bar{\bar{a}}} \rho^\alpha \omega^{i\bar{\bar{a}}} \mathbf{v}^{i\bar{\bar{a}}}. \quad (2.10)$$

Finally, we substitute (2.9) and (2.10) into (2.8) to obtain the macroscale conservation equation

$$\frac{\partial (\epsilon^{\bar{\bar{a}}} \rho^\alpha \omega^{i\bar{\bar{a}}})}{\partial t} = -\nabla \cdot (\epsilon^{\bar{\bar{a}}} \rho^\alpha \omega^{i\bar{\bar{a}}} \mathbf{v}^{i\bar{\bar{a}}}). \quad (2.11)$$

In practice, it can be difficult to measure $\mathbf{v}^{i\bar{\bar{a}}}$, so one typically closes (2.11) by splitting $\mathbf{v}^{i\bar{\bar{a}}}$ into a deviation velocity and a macroscale system velocity as detailed in [24].

We now have an averaged macroscale conservation equation, which will serve as the basis for our TCAT model.

2.4 The TCAT Model

The traditional model, as well as recent improvements to the traditional model, have some deficiencies that lead us to consider an alternative formulation. Gray and Miller [23] outlined the TCAT model, which solves many of these problems. The 1-D formulation of the model is given by

$$\epsilon^{\bar{w}} \frac{\partial \rho^w}{\partial t} = - \frac{\partial (\rho^w \epsilon^{\bar{w}} v^{\bar{w}})}{\partial z} \quad (2.12)$$

$$\epsilon^{\bar{w}} \frac{\partial (\rho \omega^{A\bar{w}})}{\partial t} = - \frac{\partial (\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} v^{\bar{w}})}{\partial z} + P_z + \Omega_z + \Gamma_z \quad (2.13)$$

$$\epsilon^{\bar{w}} v^{\bar{w}} = \frac{-1}{(\hat{R}^w - \hat{R}^d)} \left(\frac{\partial p^w}{\partial z} - \rho^w g^{\bar{w}} \right). \quad (2.14)$$

Equation (2.12) is a flow equation, which we obtain by summing our averaged macroscale conservation equation over all species. Equation (2.13) is a species conservation of mass equation. In (2.13), we split $\mathbf{v}^{\bar{w}}$ to obtain different terms for advection and diffusion. In the TCAT model, we consider diffusion due to pressure P_z , activity Γ_z , and concentration Ω_z . Equation (2.14) is a modified Darcy velocity. Note that instead of using the traditional q , we denote the Darcy velocity by $\epsilon^{\bar{w}} v^{\bar{w}}$.

The main reference that we will focus on is Miller and Gray's 2009 publication, [23]. This paper presents a TCAT-based model for single-phase species transport in porous media. When [23] was published, several closure relations, such as a closure for \hat{R}^d , had not been determined. Further, the model had not been solved or simulated numerically.

We will now present an overview of how one derives a TCAT-based model. There are eight basic steps [22].

1. Formulate an entropy inequality for entire system.
2. Formulate mass, momentum, and energy equations based on pore scale (microscale) quantities.
3. Average pore scale thermodynamics to desired scale.
4. Augment entropy inequality using Lagrange multipliers.
5. Use Lagrange multipliers to produce constrained entropy inequality.

6. Eliminate time derivatives in constrained entropy inequality using geometric identities to write a simplified entropy inequality.
7. Employ closure relations consistent with the second law of thermodynamics.
8. Construct closure relations that are constantly evolving.

Since Gray and Miller have provided a rigorous derivation of the TCAT model, we will omit the first six steps and instead direct the reader to [23]. We summarize the nomenclature for the TCAT model in the front matter on pages x and xi. We begin the derivation at step 7, where we provide a brief outline of the model closure and finally provide the equations we are solving.

We begin by considering the effect of restrictions and approximations on a species conservation of momentum equation. The result is equation (152) in [23], which is given by

$$\frac{D^{\bar{w}}(\epsilon^{\bar{w}} \rho^w \mathbf{v}^{\bar{w}})}{Dt} + \epsilon^{\bar{w}} \rho^w \mathbf{v}^{\bar{w}} \mathbf{I} : \mathbf{d}^{\bar{w}} - \nabla \cdot (\epsilon^{\bar{w}} \mathbf{t}^{\bar{w}}) - \sum_{i \in \mathcal{J}_s} \epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}} \mathbf{g}^{\bar{\alpha}} - \mathbf{T}^{ws \rightarrow w} = 0. \quad (2.15)$$

To simplify (2.15), Gray and Miller use the simplified entropy inequality (SEI) to guide closure relations. The SEI is the result of step 6 in the TCAT derivation. From these manipulations, we will have a modified Darcy velocity, $\epsilon^{\bar{w}} \nu^{\bar{w}}$, which describes the velocity field for the whole system.

Using knowledge about the system, the SEI, and first-order closures, we arrive at a form

$$\nabla p^w - \rho^w \mathbf{g}^{\bar{w}} + \epsilon^{\bar{w}} (\hat{\mathbf{R}}^w - \hat{\mathbf{R}}^d) \mathbf{v}^{\bar{w}} + \hat{\mathbf{R}}^p \nabla \epsilon^{\bar{w}}, \quad (2.16)$$

of the momentum equation, which is equation (166) in [23].

We now make the following assumptions to further reduce the model:

1. The system of interest is 1-D, with flow in the vertical (z) direction.
2. The volume fraction of the wetted phase $\epsilon^{\bar{w}}$ is spatially invariant ($\nabla \epsilon^{\bar{w}} = 0$).

With these assumptions, we now reformulate (2.16) as

$$\frac{\partial p^w}{\partial z} + \rho^w g^{\bar{w}} + \epsilon^{\bar{w}} (\hat{R}^w - \hat{R}^d) \nu^{\bar{w}} = 0. \quad (2.17)$$

Isolating $\epsilon^{\bar{w}} \nu^{\bar{w}}$ yields

$$\epsilon^{\bar{w}} \nu^{\bar{w}} = \frac{-1}{(\hat{R}^w - \hat{R}^d)} \left(\frac{\partial p^w}{\partial z} - \rho^w g^{\bar{w}} \right). \quad (2.18)$$

This is a modified form of Darcy's Law, which we use to describe our system flow.

Next, we formulate a species transport equation from our restrictions and approximations. This is equation (151) in [23] and is given by

$$\frac{D^{\bar{w}}(\epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}})}{Dt} + \epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}} \mathbf{I} : \mathbf{d}^{\bar{w}} + \nabla \cdot (\epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}} \mathbf{u}^{i\bar{w}}) = 0. \quad (2.19)$$

We simplify (2.19), which will result in both a 1-D flow equation and a 1-D species conservation of mass equation. To write our 1-D flow equation, we sum (2.19) over all species to obtain

$$\frac{D^{\bar{w}} \epsilon^{\bar{w}} \rho^w}{Dt} = 0. \quad (2.20)$$

Using the same assumptions we applied to (2.16), we can rewrite the material derivative as

$$\epsilon^{\bar{w}} \frac{\partial \rho^w}{\partial t} = - \frac{\partial (\rho^w \epsilon^{\bar{w}} \nu^{\bar{w}})}{\partial z}, \quad (2.21)$$

which is our 1-D flow equation. Next, we use (2.19) to write our 1-D species transport equation.

We simplify the material derivative in the species transport equation (2.19) to obtain

$$\frac{\partial (\epsilon^{\bar{w}} \rho^w \omega^{A\bar{w}})}{\partial t} + \nabla \cdot (\epsilon^{\bar{w}} \rho^w \omega^{A\bar{w}} \mathbf{v}^{\bar{w}}) + \nabla \cdot (\epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}} \mathbf{u}^{i\bar{w}}) = 0. \quad (2.22)$$

Gray and Miller again use the SEI as well as other physics results to develop a closure for $\nabla \cdot (\epsilon^{\bar{w}} \rho^w \omega^{i\bar{w}} \mathbf{u}^{i\bar{w}})$. Thus, (2.22) becomes

$$\begin{aligned} & (\epsilon^{\bar{w}} \rho^w \omega^{A\bar{w}})_t + \nabla \cdot (\epsilon^{\bar{w}} \rho^w \omega^{A\bar{w}} \mathbf{v}^{\bar{w}}) \\ & + \nabla \cdot \left\{ -\epsilon^{\bar{w}} \left(\frac{x^{\bar{A}\bar{w}} x^{\bar{B}\bar{w}}}{\omega^{\bar{A}\bar{w}} \omega^{\bar{B}\bar{w}}} \right) \left[(\mathbf{X}^{\bar{A}\bar{w}} - \omega^{\bar{A}\bar{w}}) \mathbf{D}^{ABwe} \cdot \nabla p^w + \rho^w \frac{R\theta^{\bar{w}}}{MW_w} \mathbf{D}^{ABwe} \cdot \nabla x^{\bar{A}\bar{w}} \right. \right. \\ & \left. \left. + \rho^w \frac{R\theta^{\bar{w}}}{MW_w \gamma^{\bar{A}\bar{w}}} x^{\bar{A}\bar{w}} \mathbf{D}^{ABwe} \cdot \nabla \gamma^{\bar{A}\bar{w}} + R^p (X^{\bar{A}\bar{w}} - \omega^{\bar{A}\bar{w}}) \mathbf{D}^{ABwe} \cdot \nabla \epsilon^{\bar{w}} \right] \right\} = 0. \end{aligned} \quad (2.23)$$

We now apply the following assumptions to (2.23):

1. The system of interest is 1-D, with flow in the vertical (z) direction.
2. The volume fraction of the wetted phase $\epsilon^{\bar{\bar{w}}}$ is spatially invariant ($\nabla \epsilon^{\bar{\bar{w}}} = 0$).
3. Because the system is binary, $x^{\bar{\bar{Bw}}} = 1 - x^{\bar{\bar{Aw}}}$ and $\omega^{\bar{\bar{Bw}}} = 1 - \omega^{\bar{\bar{Aw}}}$.

Consequently, (2.23) becomes

$$\begin{aligned} & \epsilon^{\bar{\bar{w}}} \frac{\partial(\rho^w \omega^{\bar{\bar{Aw}}})}{\partial t} + \frac{\partial(\rho^w \omega^{\bar{\bar{Aw}}} \epsilon^{\bar{\bar{w}}} \nu^{\bar{\bar{w}}})}{\partial z} \\ &= \epsilon^{\bar{\bar{w}}} \frac{\partial}{\partial z} \left[\left(\frac{x^{\bar{\bar{Aw}}}(1 - x^{\bar{\bar{Aw}}})}{\omega^{\bar{\bar{Aw}}}(1 - \omega^{\bar{\bar{Aw}}})} \right) D^{ABwe} \left((X - \omega^{\bar{\bar{Aw}}}) \frac{\partial p}{\partial z} + \frac{\rho^w R \theta^{\bar{\bar{w}}}}{M W_w} \frac{\partial x^{\bar{\bar{Aw}}}}{\partial z} + \frac{\rho^w R \theta^{\bar{\bar{w}}} x^{\bar{\bar{Aw}}}}{M W_w \hat{\gamma} i w} \frac{\partial \hat{\gamma} i w}{\partial z} \right) \right] \end{aligned} \quad (2.24)$$

We will now make use of two closure relations

$$x^{\bar{\bar{Aw}}} = \omega^{\bar{\bar{Aw}}} \left(\frac{M W_w}{M W_A} \right) \quad (2.25)$$

$$\frac{\partial x^{\bar{\bar{Aw}}}}{\partial z} = \frac{\partial \omega^{\bar{\bar{Aw}}}}{\partial z} \left(\frac{M W_A M W_B}{M W_w} \right), \quad (2.26)$$

for $x^{\bar{\bar{Aw}}}$ and $\omega^{\bar{\bar{Aw}}}$, as well as the fact that

$$X = \rho^w \bar{V} \omega^{\bar{\bar{Aw}}} \quad (2.27)$$

to simplify the right-hand side of (2.24).

We begin by examining the pressure gradient. We note that

$$\begin{aligned} & \epsilon^{\bar{\bar{w}}} \frac{\partial}{\partial z} \left(\left(\frac{x^{\bar{\bar{Aw}}}(1 - x^{\bar{\bar{Aw}}})}{\omega^{\bar{\bar{Aw}}}(1 - \omega^{\bar{\bar{Aw}}})} \right) (X - \omega^{\bar{\bar{Aw}}}) \hat{D}^{ABwe} \frac{\partial p}{\partial z} \right) \\ &= \epsilon^{\bar{\bar{w}}} \frac{\partial}{\partial z} \left(\left(\frac{1 - x^{\bar{\bar{Aw}}}}{1 - \omega^{\bar{\bar{Aw}}}} \right) \left(\frac{M W_w}{M W_A} \right) (\rho^w \bar{V} - 1) \omega^{\bar{\bar{Aw}}} \hat{D}^{ABwe} \frac{\partial p}{\partial z} \right) = P_z. \end{aligned} \quad (2.28)$$

Similarly, for the gradient with respect to mole fraction, we have

$$\begin{aligned}
& \epsilon^{\bar{w}} \frac{\partial}{\partial z} \left(\left(\frac{x^{\bar{A}\bar{w}}(1-x^{\bar{A}\bar{w}})}{\omega^{A\bar{w}}(1-\omega^{A\bar{w}})} \right) \frac{\rho R \theta^{\bar{w}}}{M W_w} \hat{D}^{ABwe} \frac{\partial x^{\bar{A}\bar{w}}}{\partial z} \right) \\
&= \epsilon^{\bar{w}} \frac{\partial}{\partial z} \left(\left(\frac{1-x^{\bar{A}\bar{w}}}{1-\omega^{A\bar{w}}} \right) \left(\frac{M W_W^2}{M W_A^2 M W_B} \right) \rho^w R \theta^{\bar{w}} \hat{D}^{ABwe} \frac{\partial \omega^{A\bar{w}}}{\partial z} \right) = \Omega_z.
\end{aligned} \tag{2.29}$$

And finally, for the activity gradient, we find that

$$\begin{aligned}
& \epsilon^{\bar{w}} \frac{\partial}{\partial z} \left(\left(\frac{x^{\bar{A}\bar{w}}(1-x^{\bar{A}\bar{w}})}{\omega^{A\bar{w}}(1-\omega^{A\bar{w}})} \right) \frac{\rho^w R \theta^{\bar{w}} x^{\bar{A}\bar{w}}}{M W_w \hat{\gamma} i w} \hat{D}^{ABwe} \frac{\partial \hat{\gamma} i w}{\partial z} \right) \\
&= \epsilon^{\bar{w}} \frac{\partial}{\partial z} \left(\left(\frac{1-x^{\bar{A}\bar{w}}}{1-\omega^{A\bar{w}}} \right) \left(\frac{M W_W}{M W_A^2} \right) \left(\frac{\rho^w R \theta^{\bar{w}}}{\hat{\gamma} i w} \right) \omega^{A\bar{w}} \hat{D}^{ABwe} \frac{\partial \hat{\gamma} i w}{\partial z} \right) = \Gamma_z.
\end{aligned} \tag{2.30}$$

We can now substitute (2.28), (2.29), and (2.30) into (2.24) to obtain

$$\epsilon^{\bar{w}} \frac{\partial (\rho^w \omega^{A\bar{w}})}{\partial t} = - \frac{\partial (\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} \nu^{\bar{w}})}{\partial z} + P_z + \Omega_z + \Gamma_z. \tag{2.31}$$

This yields the model

$$\epsilon^{\bar{w}} \frac{\partial \rho^w}{\partial t} = - \frac{\partial (\rho^w \epsilon^{\bar{w}} \nu^{\bar{w}})}{\partial z} \tag{2.32}$$

$$\epsilon^{\bar{w}} \frac{\partial (\rho^w \omega^{A\bar{w}})}{\partial t} = - \frac{\partial (\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} \nu^{\bar{w}})}{\partial z} + P_z + \Omega_z + \Gamma_z \tag{2.33}$$

$$\epsilon^{\bar{w}} \nu^{\bar{w}} = \frac{\frac{\partial \rho^w}{\partial z} - \rho^w g^{\bar{w}}}{-(\hat{R}^w - \hat{R}^d)}. \tag{2.34}$$

Equations (2.32), (2.33), and (2.34) represent the model as formulated in [23]. When [23] was published, several closure relations, such as a closure for \hat{R}^d , had not been determined. Further, the model had not been simulated and there was no numerical basis for a solution.

Next, we outline the additional mathematical relations so that we have a solvable system. In Chapter 3, we will discuss our reformulation of the equations and our numerical solution method.

2.5 Determination of Closure Relations

It was stated in [23] that some of the terms do not have closed forms, meaning that the exact formulations of the closure relations had not been determined at the time of publication. We remedy this by using model simulations compared to experimental data to inform these closure relations.

We began by considering our formulation of the closure relation \hat{R}^d , which appears in both our Darcy velocity and our equation for spatially varying diffusion. We found that when we set $\hat{R}^d = c \frac{\partial \rho^w}{\partial z}$, the Darcy velocity became extremely large. Therefore, we split \hat{R}^d into two different forms. We created $\hat{R}_1^d = c_1 \frac{\partial \rho^w}{\partial z}$, which appears in the Darcy velocity, and $\hat{R}_2^d = c_2 \frac{\partial \rho^w}{\partial z}$, which appears in our closure for spatially varying diffusion.

Next, we performed parameter fits for c_1 and c_2 using the model output compared to data generated at the University of North Carolina. We had initially determined that physically, we needed $c_1 \geq 0$ and $c_2 \leq 0$. However, when we ran our code with $c_2 < 0$, our spatially varying diffusion calculation became unstable and caused solver failure. We changed $c_2 \geq 0$. We subsequently had implemented a robust model and could compare the model to data.

We determined that $c_2 \gg c_1$, so we modified our code to estimate c_1 and $\log(c_2)$. However, when we performed this parameter estimation, we found that c_1 has little effect on the output and was consequently not identifiable. We then set c_1 , the deviation term in the Darcy velocity, equal to zero. This left only one parameter to be fit in the diffusion closure \hat{R}_2^d . Therefore, we proceeded with the relations

$$\hat{R}_1^d = 0, \quad (2.35)$$

$$\hat{R}_2^d = c_1 \frac{\partial \rho^w}{\partial z}. \quad (2.36)$$

Note that we have changed the notation of the parameter in \hat{R}_2^d from c_2 to c_1 for clarity.

2.6 Initial and Boundary Conditions

To have a well-posed problem, we need initial and boundary data for our primary computational variables. For mass fraction $\omega^{A\overline{w}}$, we set

$$\omega^{A\bar{w}}(z, 0) = \begin{cases} \omega^{A\bar{w}}_{in}, & z = 0 \\ \omega^{A\bar{w}}_{col}, & \text{else} \end{cases}$$

as our initial condition. Here $\omega^{A\bar{w}}_{in}$ is the mass fraction of the solution we are pumping into the bottom of the column and $\omega^{A\bar{w}}_{col}$ is the mass fraction of the solution inside of the column at time $t = 0$. This states that the incoming brine is at base of the column at time $t = 0$.

The bottom boundary condition is

$$\frac{\partial \omega^{A\bar{w}}(0, t)}{\partial t} = 0. \quad (2.37)$$

This says that we are pumping at a constant concentration [62].

To obtain a top boundary condition for $\omega^{A\bar{w}}$, we observe that there is no diffusion at the boundary, which yields [77]

$$\epsilon^{\bar{w}} \frac{\partial \omega^{A\bar{w}}}{\partial t}(L, t) = \frac{\partial \omega^{A\bar{w}}}{\partial z}(L, t). \quad (2.38)$$

Note that L is the height of the vertical column.

Next, we consider initial and boundary conditions for pressure p^w . For our initial condition, we use the hydrostatic condition since there is no flow in the column at time $t = 0$. This is expressed mathematically by

$$p^w(z, 0) = \rho^w(\omega^{A\bar{w}}(z, 0))g(z - L), \quad (2.39)$$

where z is the spatial mesh width [77].

To derive the bottom boundary condition, we set the volumetric flow rate $\frac{Q}{A_c}$ equal to our measured velocity,

$$\epsilon^{\bar{w}} v^{\bar{w}}(0, t) = \frac{Q}{A_c}. \quad (2.40)$$

We then reformulate (2.40) to determine

$$\frac{\partial p^w}{\partial z}(0, t) = \frac{-Q\hat{\mu}^w}{\hat{k}A_c} - \rho^w(\omega^{A\bar{w}}_0)g^{\bar{w}},$$

which is the bottom boundary condition for pressure. In (2.40), \hat{k} is the intrinsic permeability and $g^{\overline{w}}$ is acceleration due to gravity.

Finally, we know that the outgoing pressure of the fluid is zero, so we take the top boundary condition to be

$$p^w(L, t) = 0. \quad (2.41)$$

With initial and boundary data for both of the primary computational variables, the problem is well-posed.

2.7 Linearized Version of the TCAT Model

In this section, we outline the assumptions we make in order to simplify the TCAT model. While it is preferable to simulate the high-fidelity TCAT model when possible, the mixed-effects model, which we develop and analyze in Chapter 7, cannot use the high-fidelity TCAT model due to the dependence of closure relations on incoming mass fraction. Therefore, our goal in this section is to arrive at an observation-independent form of the TCAT model that we can use for our mixed-effects model.

2.7.1 Simplifying Assumptions

The linearized TCAT model sacrifices some of the accuracy and robustness of the full TCAT model due to the assumptions needed to simplify the model. Namely, we assume certain quantities are constant in the linearized model, whereas the full model allows these quantities to be spatially varying.

The most important assumption that we make is that our solution has some dilute characteristics. This allows us to simulate a breakthrough curve independent of the incoming concentration of the fluid so that we can run observation-independent mixed-effects simulations.

We make several assumptions to arrive at the linear model. Because we are neglecting spatial gradients, we know that both mass fraction $\omega^{A\overline{w}}$ and mole fraction $x^{A\overline{w}}$ are small. Thus, we assume that

$$1 - x^{A\overline{w}} \approx 1 \quad (2.42)$$

$$1 - \omega^{A\overline{w}} \approx 1. \quad (2.43)$$

Next, we consider how the linearization affects the density and viscosity of the system. Because the viscosity $\mu^{\bar{w}}$ and the density ρ^w are functions of mass fraction, there is little spatial variation and we therefore assume them to be constant over space. We assume that its density is equal to the density of water and set

$$\rho^w = 1 \frac{\text{g}}{\text{cm}^3}. \quad (2.44)$$

We also assume that the viscosity of the solution is equal to the viscosity of water and set

$$\mu^{\bar{w}} = 0.89 \text{ cp}. \quad (2.45)$$

For both density and viscosity, the spatial gradients are zero.

In the dilute case, the Darcy velocity simplifies to a constant. Recall that the Darcy velocity is given by

$$\epsilon^{\bar{w}} v^{\bar{w}} = \frac{-\hat{k}}{\mu^{\bar{w}}} \left(\frac{\partial p^w}{\partial z} + \rho^w g \right). \quad (2.46)$$

The pressure p^w is close to constant and therefore its spatial gradient is negligible. Intrinsic permeability \hat{k} and gravity g are both model constants, and we have already determined that the density and viscosity are constant. Thus, the Darcy velocity is a known constant and is given by

$$\epsilon^{\bar{w}} v^{\bar{w}} = 5.77 \times 10^{-3} \text{ m/s}. \quad (2.47)$$

We also assume that diffusion is spatially invariant. Mathematically, this is equivalent to stating that

$$\hat{D}^{ABwe} = \hat{D}^{AB}. \quad (2.48)$$

Next, we assume that the pressure and activity gradients have negligible contributions to the model and set

$$\Gamma_z = P_z = 0. \quad (2.49)$$

Finally, we can simplify our weighted molecular weight $M W_w$, which is given by

$$M W_w = M W_A x^{\bar{Aw}} + M W_B (1 - x^{\bar{Aw}}). \quad (2.50)$$

Because of assumption (2.43), we can assume that

$$M W_W \approx M W_B. \quad (2.51)$$

2.7.2 Applying Simplifying Assumptions to TCAT

We now take the assumptions and apply them to the TCAT model. Recall that the 1-D flow equation is given by

$$\epsilon^{\bar{w}} \frac{\partial \rho^w}{\partial t} = - \frac{\partial}{\partial z} (\rho^w \epsilon^{\bar{w}} v^{\bar{w}}) \quad (2.52)$$

and the 1-D species conservation of mass equation is given by

$$\epsilon^{\bar{w}} \frac{\partial (\rho^w \omega^{A\bar{w}})}{\partial t} = - \frac{\partial}{\partial z} (\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} v^{\bar{w}}) + P_z + \Gamma_z + \Omega_z. \quad (2.53)$$

We begin by simplifying (2.52) with our dilute assumptions. Because $\epsilon^{\bar{w}}$ is a model constant and $\epsilon^{\bar{w}} v^{\bar{w}}$ is constant in space, we can rewrite (2.52) as

$$\frac{\partial \rho^w}{\partial t} = \frac{-\epsilon^{\bar{w}} v^{\bar{w}}}{\epsilon^{\bar{w}}} \frac{\partial \rho^w}{\partial z}. \quad (2.54)$$

Next, we simplify (2.53). We again assume that $\epsilon^{\bar{w}} v^{\bar{w}}$ is constant in space, as well as making the substitution determined in (2.49). Therefore, (2.53) becomes

$$\epsilon^{\bar{w}} \frac{\partial (\rho^w \omega^{A\bar{w}})}{\partial t} = -\epsilon^{\bar{w}} v^{\bar{w}} \frac{\partial}{\partial z} (\rho^w \omega^{A\bar{w}}) + \Omega_z. \quad (2.55)$$

Recall that

$$\Omega_z = \frac{\partial}{\partial z} \left[\epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W^2}{M W_A^2 M W_B} \right) \rho^w R \theta^{\bar{w}} \hat{D}^{ABwe} \frac{\partial \omega^{A\bar{w}}}{\partial z} \right]. \quad (2.56)$$

We can substitute (2.42) and (2.43) to obtain

$$\Omega_z = \epsilon^{\bar{w}} R \theta^{\bar{w}} \frac{\partial}{\partial z} \left[\left(\frac{M W_W^2}{M W_A^2 M W_B} \right) \hat{D}^{ABwe} \frac{\partial \omega^{A\bar{w}}}{\partial z} \right]. \quad (2.57)$$

We can further simplify (2.57) by considering (2.51) and (2.48) so that

$$\Omega_z = \frac{\epsilon^{\bar{w}} R \theta^{\bar{w}} \hat{D}^{AB} M W_B}{M W_A^2} \frac{\partial}{\partial z} \left(\rho^w \frac{\partial \omega^{A\bar{w}}}{\partial z} \right). \quad (2.58)$$

We next substitute (2.58) into (2.55) to obtain

$$\epsilon^{\bar{w}} \frac{\partial (\rho^w \omega^{A\bar{w}})}{\partial t} = -\epsilon^{\bar{w}} v^{\bar{w}} \frac{\partial}{\partial z} (\rho^w \omega^{A\bar{w}}) + \frac{\epsilon^{\bar{w}} R \theta^{\bar{w}} \hat{D}^{AB} M W_B}{M W_A^2} \frac{\partial}{\partial z} \left(\rho^w \frac{\partial \omega^{A\bar{w}}}{\partial z} \right) \quad (2.59)$$

and perform a change of variables such that

$$C = \rho^w \omega^{A\bar{w}}, \quad (2.60)$$

where C is the concentration of the solution.

The second term on the right-hand side of (2.59) must be analyzed before we can make the change of variables. We note that

$$\frac{\partial^2 C}{\partial z^2} = \frac{\partial^2}{\partial z^2} (\rho^w \omega^{A\bar{w}}) \quad (2.61)$$

$$= \frac{\partial}{\partial z} \left(\rho_z^w \omega^{A\bar{w}} + \rho^w \frac{\partial \omega^{A\bar{w}}}{\partial z} \right) \quad (2.62)$$

and therefore

$$\frac{\partial}{\partial z} \left(\rho^w \frac{\partial \omega^{A\bar{w}}}{\partial z} \right) = \frac{\partial^2 C}{\partial z^2} - \frac{\partial}{\partial z} (\rho_z^w \omega^{A\bar{w}}). \quad (2.63)$$

We have already determined that $\frac{\partial}{\partial z} \rho^w = 0$ so that (2.63) becomes

$$\frac{\partial^2 C}{\partial z^2} = \frac{\partial}{\partial z} \left(\rho^w \frac{\partial \omega^{A\bar{w}}}{\partial z} \right). \quad (2.64)$$

After making the substitution in (2.64), our species conservation of mass equation simplifies to the linear form

$$\epsilon^{\bar{w}} \frac{\partial C}{\partial t} = -\epsilon^{\bar{w}} v^{\bar{w}} \frac{\partial C}{\partial z} + D \frac{\partial^2 C}{\partial z^2}, \quad (2.65)$$

where

$$D = \frac{\epsilon^{\bar{w}} R \theta^{\bar{w}} \hat{D}^{AB} M W_B}{M W_A^2}. \quad (2.66)$$

2.7.3 Analytic Solution

A nearly complete set of solutions to the one-dimensional advection-diffusion problem were published by Javandel [35] given (2.65) with an inlet boundary condition of

$$C(0, t) = C_0, \quad (2.67)$$

an outlet boundary condition of

$$\frac{\partial C}{\partial z}(L, t) = 0, \quad (2.68)$$

and an initial condition of

$$C(z, 0) = 0. \quad (2.69)$$

Javandel found the analytic solution to be [35]

$$\frac{C}{C_0}(L, t) = \frac{1}{2} \operatorname{erfc}\left(\frac{L - \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right) + \frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} L}{D}\right) \operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right). \quad (2.70)$$

One problem, however, is that when we attempt to evaluate the solution (2.70), we encounter a problem with the term $\frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} L}{D}\right) \operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right)$. Specifically, as $D \rightarrow 0$, the exponentiation term tends to infinity, whereas the erfc term tends to zero. To determine the behavior of the term when D is sufficiently small, we analyze the limit of this term as $D \rightarrow 0$.

To determine the limit of the term $\frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} L}{D}\right) \operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right)$, we will use L'Hospital's rule. We rewrite the term as

$$\lim_{D \rightarrow 0} \frac{1}{2} \frac{\operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right)}{\left(\exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} L}{D}\right)\right)^{-1}}. \quad (2.71)$$

Taking the derivative of the top and bottom of (2.71) yields

$$\lim_{D \rightarrow 0} \frac{(Dt)^{1/2} (L + \epsilon^{\bar{w}} v^{\bar{w}} t)}{4\sqrt{\pi} \epsilon^{\bar{w}} v^{\bar{w}} L t} \exp\left(\frac{-(L - \epsilon^{\bar{w}} v^{\bar{w}} t)^2}{4Dt}\right). \quad (2.72)$$

We begin by addressing the quantity $\frac{(Dt)^{1/2}(L + \epsilon^{\bar{w}} \nu^{\bar{w}} t)}{2\sqrt{\pi}\epsilon^{\bar{w}} \nu^{\bar{w}} L t}$ in front of the exponential. We note that the quantity D only appears in the numerator within a square root; therefore

$$\lim_{D \rightarrow 0} \frac{(Dt)^{1/2}(L + \epsilon^{\bar{w}} \nu^{\bar{w}} t)}{4\sqrt{\pi}\epsilon^{\bar{w}} \nu^{\bar{w}} L t} \xrightarrow{D \rightarrow 0} 0. \quad (2.73)$$

Next, we consider the exponential term. The numerator is always negative or zero as it is the negative of the squared quantity $(L - \epsilon^{\bar{w}} \nu^{\bar{w}} t)^2$ and the denominator is always positive with $4Dt > 0$ since time and diffusion must always be positive. Therefore, the entire quantity to be exponentiated is nonpositive for this problem.

There are two cases we must consider when determining the behavior of the numerator for this problem. The first is when $L \neq \epsilon^{\bar{w}} \nu^{\bar{w}} t$, and the quantity $\left(\frac{-(L - \epsilon^{\bar{w}} \nu^{\bar{w}} t)^2}{4Dt}\right)$ is negative. This means that $\exp\left(\frac{-(L - \epsilon^{\bar{w}} \nu^{\bar{w}} t)^2}{4Dt}\right)$ is a negative exponential which tends to zero as $D \rightarrow 0$. We have thus shown

$$\lim_{D \rightarrow 0} \frac{1}{2} \frac{\text{erfc}\left(\frac{L + \epsilon^{\bar{w}} \nu^{\bar{w}} t}{2(Dt)^{1/2}}\right)}{\left(\exp\left(\frac{\epsilon^{\bar{w}} \nu^{\bar{w}} L}{D}\right)\right)^{-1}} = 0 \quad (2.74)$$

when $L \neq \epsilon^{\bar{w}} \nu^{\bar{w}} t$.

We next consider the case where $L = \epsilon^{\bar{w}} \nu^{\bar{w}} t$. This yields $\exp\left(\frac{-(L - \epsilon^{\bar{w}} \nu^{\bar{w}} t)^2}{4Dt}\right) = \exp^0 = 1$. Then, we evaluate

$$\lim_{D \rightarrow 0} \frac{(Dt)^{1/2}(L + \epsilon^{\bar{w}} \nu^{\bar{w}} t)}{4\sqrt{\pi}\epsilon^{\bar{w}} \nu^{\bar{w}} L t}, \quad (2.75)$$

which we have shown tends to zero as $D \rightarrow 0$. We have thus shown

$$\lim_{D \rightarrow 0} \frac{1}{2} \frac{\text{erfc}\left(\frac{L + \epsilon^{\bar{w}} \nu^{\bar{w}} t}{2(Dt)^{1/2}}\right)}{\left(\exp\left(\frac{\epsilon^{\bar{w}} \nu^{\bar{w}} L}{D}\right)\right)^{-1}} = 0 \quad (2.76)$$

for all admissible parameter values.

The exact solution for small D then becomes

$$\lim_{D \rightarrow 0} \frac{C}{C_0}(L, t) = \lim_{D \rightarrow 0} \left(\frac{1}{2} \text{erfc}\left(\frac{L - \epsilon^{\bar{w}} \nu^{\bar{w}} t}{2(Dt)^{1/2}}\right) + \frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} \nu^{\bar{w}} L}{D}\right) \text{erfc}\left(\frac{L + \epsilon^{\bar{w}} \nu^{\bar{w}} t}{2(Dt)^{1/2}}\right) \right)$$

$$= \frac{1}{2} \operatorname{erfc} \left(\frac{L - \epsilon^{\overline{w}} v^{\overline{w}} t}{2(Dt)^{1/2}} \right). \quad (2.77)$$

This formulation allows us to solve for $\frac{C}{C_0}$.

The analytic solution presented by Javandel is independent of incoming concentration—consequently, we can use this analytic solution as the basis for our mixed-effects model in Chapter 7.

2.8 Mathematical Challenges

Here, we summarize challenges we have encountered in this component of the investigation. We will also detail the mathematical contribution for this problem. In the next chapters, we will further discuss these mathematical challenges.

- We first determined how to reformulate (2.32), (2.33), and (2.34). Watson and Barry [76] used a Crank Nicholson method to discretize the derivatives and then wrote the equations in matrix form and performed a direct solve. Others used an operator splitting scheme [67]. We wanted a robust form of the model that we could solve with a packaged solver. Therefore, we chose to reformulate (2.32) and (2.33) as a partial differential algebraic equation (PDAE) in Section 3.3.
- Once we had the model in a form we could solve, we modified the inner workings of the codes. To reformulate the PDAE as an ordinary DAE, we used finite difference schemes to approximate the spatial derivatives, as shown in Section 3.4. We chose certain schemes to accurately model physical phenomena—for example, we used upwinding to approximate mass fraction, which allows the speed of our front to correctly move forward in time [48]. This yielded a large system of ordinary DAEs.
- After we reformulated the TCAT model as a large system of DAEs, we had to show that the system is index-1. Many DAE codes, such as ode15s, IDA, DASSL, and SPRINT, are only equipped to solve index-1 problems [8]. To numerically show the DAE is index-1, we showed that the Jacobian with respect to the algebraic variables is nonsingular [3] in Section 3.5.1. Because we showed that we have an index-1 DAE, our stable form allows us to run simulations for a variety of incoming concentrations and parameter values.

- After we finished updating the model with closure relations and boundary conditions, we estimated parameters with the Levenberg-Marquardt algorithm. Our goal was to have a single parameter that describes all data collected at the University of North Carolina. While we were not able to find a unique parameter that provided a good fit to data for all experiments, we show in Section 4.3 that we were able to find a parameter that could describe data with an incoming mass fractions less than 0.15.
- In Chapter 5, we constructed a regression-based surrogate model for TCAT using cubic splines to emulate the parameter-dependent response of the TCAT model. This surrogate model significantly decreases run times and makes Bayesian parameter estimation possible.
- We applied Bayesian inference to the TCAT model and determined distributions for our parameter with respect to different experimental scenarios in Chapter 6.
- In Chapter 7, we created and implemented a mixed-effects model for a simplified version of the TCAT model to quantify the contributions of the fixed and random parameter effects.

CHAPTER

3

MATHEMATICAL SIMULATION OF THE TCAT MODEL

In this chapter, we present the necessary mathematics required to simulate the TCAT model.

3.1 The TCAT Model

In Chapter 2, we derived the TCAT model,

$$\epsilon^{\bar{w}} \frac{\partial}{\partial t} \rho^w = - \frac{\partial}{\partial z} \left(\rho^w \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}} \right) \quad (3.1)$$

$$\epsilon^{\bar{w}} \frac{\partial}{\partial t} \left(\rho^w \omega^{A\bar{w}} \right) = - \frac{\partial}{\partial z} \left(\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}} \right) + P_z + \Gamma_z + \Omega_z \quad (3.2)$$

$$\epsilon^{\bar{w}} \mathbf{v}^{\bar{w}} = - \frac{1}{\hat{R}^w - \hat{R}_1^d} \left(\frac{\partial p^w}{\partial z} - \rho^w g \right). \quad (3.3)$$

The closure relations are

$$P_z = \frac{\partial}{\partial z} \left[\epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W}{M W_A} \right) (\rho^w \bar{V} - 1) \omega^{A\bar{w}} \hat{D}^{ABwe} \frac{\partial p^w}{\partial z} \right] \quad (3.4)$$

$$\Gamma_z = \frac{\partial}{\partial z} \left[\epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W}{M W_A^2} \right) \left(\frac{\rho^w R \theta^{\bar{w}}}{\hat{\gamma} i w} \right) \omega^{A\bar{w}} \hat{D}^{ABwe} \frac{\partial \hat{\gamma} i w}{\partial z} \right] \quad (3.5)$$

$$\Omega_z = \frac{\partial}{\partial z} \left[\epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W^2}{M W_A^2 M W_B} \right) \rho^w R \theta^{\bar{w}} \hat{D}^{ABwe} \frac{\partial \omega^{A\bar{w}}}{\partial z} \right] \quad (3.6)$$

$$\hat{R}^w = \frac{\hat{\mu}^w}{\hat{k}} \quad (3.7)$$

$$\hat{R}_1^d = 0 \quad (3.8)$$

$$x^{\bar{A}\bar{w}} = \frac{M W_B \omega^{A\bar{w}}}{M W_B \omega^{A\bar{w}} + M W_A (1 - \omega^{A\bar{w}})} \quad (3.9)$$

$$M W_w = M W_A x^{\bar{A}\bar{w}} + M W_B (1 - x^{\bar{A}\bar{w}}) \quad (3.10)$$

$$\bar{V} = \frac{1}{\rho^w} + (1 - \omega^{A\bar{w}}) \frac{\partial}{\partial \omega^{A\bar{w}}} \left(\frac{1}{\rho^w} \right) \quad (3.11)$$

$$m = \frac{1000 \omega^{A\bar{w}}}{M W_A (1 - \omega^{A\bar{w}})} \quad (3.12)$$

$$\hat{D}^{ABwe} = \left[1 - \left(\frac{\epsilon^{\bar{w}} \hat{R}_2^d}{\rho^w} \right) \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_w}{M W_A} \right) \hat{D}^{AB} \right]^{-1} \hat{D}^{AB} \quad (3.13)$$

$$\hat{R}_2^d = c_1 \frac{\partial \rho^w}{\partial z}. \quad (3.14)$$

We have experimentally determined relations for density ρ^w and viscosity $\hat{\mu}^w$ as a function of mass fraction, where

$$\rho^w(\omega^{A\bar{w}}) = .8250(\omega^{A\bar{w}})^3 + .5038(\omega^{A\bar{w}})^2 + .8411\omega^{A\bar{w}} + .9982 \quad (3.15)$$

$$\hat{\mu}^w(\omega^{A\bar{w}}) = .01 \times e^{17.18(\omega^{A\bar{w}})^3 - 6.262(\omega^{A\bar{w}})^2 + 1.861\omega^{A\bar{w}} - .03221}. \quad (3.16)$$

We determine the activity, $\hat{\gamma} i w$, by applying an extended form of the Debye-Huckel equation. This relates mole fraction m to activity through an intermediate ionic strength I via the relations [60]

$$I = 3m \quad (3.17)$$

$$\hat{\gamma} i w = \exp\left(\frac{-a_1 \sqrt{I}}{1 + a_2 \sqrt{I}} + a_3 m + a_4 m^2 + a_5 m^3 + a_6 m^4 + a_7 m^5 + a_8 m^6\right), \quad (3.18)$$

where a_1, \dots, a_8 are coefficients determined by parameter fitting.

Finally, we have the initial and boundary conditions

$$\omega^{A\bar{w}}(z, 0) = \begin{cases} \omega^{A\bar{w}}_{in}, & z = 0 \\ \omega^{A\bar{w}}_{col}, & \text{else} \end{cases} \quad (3.19)$$

$$\frac{\partial \omega^{A\bar{w}}(0, t)}{\partial t} = 0 \quad (3.20)$$

$$\epsilon^{\bar{w}} \frac{\partial \omega^{A\bar{w}}}{\partial t}(L, t) = \frac{\partial \omega^{A\bar{w}}}{\partial z}(L, t) \quad (3.21)$$

$$p^w(z, 0) = \rho^w(\omega^{A\bar{w}}(z, 0))g(z - L) \quad (3.22)$$

$$\frac{\partial p^w}{\partial z}(0, t) = \frac{-Q\hat{\mu}^w}{\hat{k}A_c} - \rho^w(\omega^{A\bar{w}}_0)g^{\bar{w}} \quad (3.23)$$

$$p^w(L, t) = 0 \quad (3.24)$$

so that the model is well-posed.

In the remainder of this chapter, we show how we transform the PDEs (3.1) and (3.2) into a form that we can solve easily in a prepackaged ODE solver. We will first introduce the concept of a differential-algebraic equation and will then transform the PDEs into a system of partial differential-algebraic equations. Next, we show how we used the method of lines to solve the transformed equations in MATLAB's ODE solver `ode15s`. We will then discuss how `ode15s` yields a numerical solution to the problem through time integration of the differential-algebraic equations. We next present the Jacobian pattern we provided to the solver and discuss solver settings as well as modifications we made to `ode15s`. Finally, we will discuss the adaptive gridding method and perform a spatial grid refinement study to show our simulations are grid independent.

3.2 Differential-Algebraic Equations

In this section, we introduce some of the terminology pertaining to differential-algebraic equations (DAEs), including the concept of index.

A DAE is a mixture of both ODEs and algebraic constraints. The general form of a DAE is

$$F(t, y, y') = 0. \quad (3.25)$$

The reformulated TCAT model is a semi-explicit PDAE, which means that it can be written as a set of differential equations and a set of algebraic constraints. The general form is

$$x' = f(t, x, z) \quad (3.26)$$

$$0 = g(t, x, z). \quad (3.27)$$

In a semi-explicit DAE, we distinguish between the differential variables x and the algebraic variables z . Here (3.26) are the differential equations and (3.27) are the algebraic constraints. When we consider DAEs, we must also consider the concept of index.

Definition 3.2.1. *The index of a DAE is the number of differentiations with respect to t needed to transform the DAE into an explicit ODE system [3, 8].*

A semi-explicit DAE will be index-1 if the Jacobian of the algebraic constraint with respect to the algebraic variables, generally given by $\frac{\partial g}{\partial z}$, is nonsingular [3]. In our model, $\omega^{A\overline{w}}$ is the differential variable and p^w is the algebraic variable. While showing analytically that the Jacobian $\frac{\partial g}{\partial p^w}$ is nonsingular is difficult, we will later numerically verify the index of the DAE by computing the rank of the Jacobian during simulations.

We next show how we transform the TCAT model into a semi-explicit DAE.

3.3 Transforming the TCAT Model

We start with the two PDEs (3.1) and (3.2). We expand the time derivatives on the left-hand side and isolate $\epsilon^{\overline{w}} \frac{\partial}{\partial t} \omega^{A\overline{w}}$ on the left-hand side to obtain

$$\epsilon^{\bar{w}} \frac{\partial}{\partial t} \omega^{A\bar{w}} = -\frac{1}{(\rho^w)'} \frac{\partial}{\partial z} (\rho^w \epsilon^{\bar{w}} v^{\bar{w}}) = F_1 \quad (3.28)$$

$$\epsilon^{\bar{w}} \frac{\partial}{\partial t} \omega^{A\bar{w}} = \frac{-1}{\rho^w + (\rho^w)' \omega^{A\bar{w}}} \frac{\partial}{\partial z} (\rho^w \omega^{A\bar{w}} \epsilon^{\bar{w}} v^{\bar{w}} + P + \Gamma + \Omega) = F_2. \quad (3.29)$$

Finally, we subtract (3.29) from (3.28) to obtain

$$\begin{pmatrix} \epsilon^{\bar{w}} \\ 0 \end{pmatrix} \begin{pmatrix} \omega^{A\bar{w}} \\ p^w \end{pmatrix}' = \begin{pmatrix} F_1 \\ F_1 - F_2 \end{pmatrix}. \quad (3.30)$$

In (3.30), the prime denotes the time derivative, and the goal is to solve for our primary computational variables p^w and $\omega^{A\bar{w}}$. Because the mass matrix is singular, we now have a coupled system that consists of one PDE and one algebraic constraint. This is a semi-explicit partial differential-algebraic equation.

Whereas it is tempting to use substitution to rewrite (3.30) as a single ODE, this approach has several drawbacks. First, we must remember that a DAE is a collection of relations between variables. Reformulating this DAE can create variables with less physical significance, and it can cause us to lose valuable information about the system [8, 38]. Further, rewriting the DAE can destroy inherent sparsity patterns, which can lead to unnecessarily performing a dense solve, thus wasting valuable computation time. We instead solve the DAE directly to explore changes to the model and variations in parameters [8].

Next, we use the method of lines to reformulate the partial differential-algebraic equation as a large system of ordinary DAEs. To accomplish this, we use finite difference approximations to estimate the spatial derivatives on the right-hand side of (3.30). This yields a large system of ordinary DAEs [38, 44]. Reformulating the TCAT equations as a large system of ordinary DAEs has the advantage of allowing us to use an advanced ODE solver to perform our simulations.

3.4 The Method of Lines

In this section, we show how we use the method of lines to transform the PDAEs into a large system of ordinary DAEs. Generally, the method of lines involves using approximations for

the spatial derivatives to transform a system of PDEs into a large system of ODEs. From there, one uses a temporal integrator to solve for the time derivative and arrive at a solution to the PDE that preserves its spatial and temporal characteristics. For the TCAT model, we take the spatial derivatives on the left-hand side of (3.30) and replace them with finite difference approximations to obtain the large system of ordinary DAEs. Next, we outline the finite difference approximations we use when considering the method of lines.

3.4.1 Finite Difference Schemes

When choosing our finite difference scheme with respect to the spatial mesh for $\omega^{A\bar{w}}$, we consider upwinding. This allows us to accurately simulate advection. When we consider a front that is moving in time, the PDE that describes flow is asymmetric because the equations model transport at a given speed. Therefore, it is advantageous to use a one-sided difference instead of a symmetric centered difference.

Upwinding involves using a one-sided, first-order finite difference approximation for the spatial mesh. In our case, we know that the transport speed is $u > 0$. To preserve stability, we use the backward method

$$\frac{\partial \omega^{A\bar{w}}}{\partial z} \approx \frac{\omega^{A\bar{w}}_i - \omega^{A\bar{w}}_{i-1}}{h}. \quad (3.31)$$

This allows us to accurately track our front [44, 48, 65, 81].

To accurately approximate $\frac{\partial p^w}{\partial z}$, we use a first-order forward difference equation

$$\frac{\partial p^w}{\partial z} \approx \frac{p^w_{i+1} - p^w_i}{h}. \quad (3.32)$$

Finally, we must also take a spatial derivative of our activity, $\hat{\gamma} i w$. We simply taken an analytic derivative of the Debye-Huckel equation and multiply it by $\frac{\partial \omega^{A\bar{w}}}{\partial z}$ to obtain $\frac{\partial \hat{\gamma} i w}{\partial z}$.

As a future direction for this project, one could investigate the effect of a second-order spatial discretization. A second order method could lead to more accurate solutions and could potentially resolve some of the difficulties we encountered simulating the highest concentration experiments.

3.4.2 Conservation Form

There are several quantities in our model that need to be evaluated in conservation form. First, the derivatives $\frac{\partial}{\partial z}(\rho^w \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}})$ and $\frac{\partial}{\partial z}(\omega^{A\bar{w}} \rho^w \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}})$ are products of spatially varying quantities. The fluxes P_z , Ω_z , and Γ_z are also the derivative of products of terms that vary in space.

When considering these quantities, we take the derivatives in conservation form, which allows us to better approximate them [41]. This means that we multiply all of the internal terms together and use a finite difference approximation of the product to take the derivative with respect to z . For example, consider P_z , which is given by

$$P_z = \frac{\partial}{\partial z} \left[\epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W}{M W_A} \right) (\rho^w \bar{V}_{Aw} - 1) \omega^{A\bar{w}} \hat{D}^{ABwe} \frac{\partial p^w}{\partial z} \right]. \quad (3.33)$$

To illustrate how we take the derivative with respect to z , we rewrite

$$\xi(z) = \epsilon^{\bar{w}} \left(\frac{1 - x^{\bar{A}\bar{w}}}{1 - \omega^{A\bar{w}}} \right) \left(\frac{M W_W}{M W_A} \right) (\rho^w \bar{V}_{Aw} - 1) \omega^{A\bar{w}} \hat{D}^{ABwe} \quad (3.34)$$

and thus (3.33) becomes

$$P_z = \frac{\partial}{\partial z} \left(\xi(z) \frac{\partial p^w}{\partial z} \right). \quad (3.35)$$

Taking the finite difference approximation of (3.35) yields

$$P_z \approx \frac{(\xi(z) \frac{\partial p^w}{\partial z})_i - (\xi(z) \frac{\partial p^w}{\partial z})_{i-1}}{h}. \quad (3.36)$$

The remaining diffusion terms Ω_z and Γ_z , as well as the products $\frac{\partial}{\partial z}(\rho^w \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}})$ and $\frac{\partial}{\partial z}(\omega^{A\bar{w}} \rho^w \epsilon^{\bar{w}} \mathbf{v}^{\bar{w}})$, are treated similarly. Note that we have used a first-order backward difference approximation. This approximation allows us to add diffusion to the model and maintain the correct speed of the front.

3.5 Time Integration in MATLAB

Now that we have reformulated our PDAEs as a large system of ordinary DAEs, we next consider how to solve for the temporal derivative. We use MATLAB's `ode15s`, which is an

advanced ODE solver for stiff problems. The `ode15s` code also solves DAEs, but they must be index-1. In the next section, we present numerical verification that the transformed TCAT model is index-1.

3.5.1 Numerical Verification of Index

Recall that the reformulated TCAT equations are given by

$$\epsilon^{\bar{w}} \frac{\partial \omega^{A\bar{w}}}{\partial t} = F_1 \quad (3.37)$$

$$0 = F_1 - F_2. \quad (3.38)$$

This is a semi-explicit DAE with an algebraic constraint of $F_1 - F_2$.

Our goal is to show that the Jacobian of the algebraic constraints, with respect to the algebraic variables p^w , given by

$$\frac{\partial (F_1 - F_2)}{\partial p^w} \quad (3.39)$$

is nonsingular. This will verify that our system of DAEs is index-1 [3].

Since the temporal integrator `ode15s` does not distinguish between an ODE and an index-1 DAE [68], it uses the same direct approach for solving index-1 DAEs that is presented in the codes DASSL, LSODI, IDA, and SPRINT [70]. DASSL, and consequently `ode15s`, cannot solve systems with a greater index than 1 without changes to several components in the code [8, 10]. For a DAE of higher index, the error estimates utilized in DASSL and `ode15s` do not move towards zero as the step size is decreased. The code attempts to reduce the step size until the iteration matrix is drastically ill-conditioned, and this causes the Newton solver to fail [8]. Therefore, it is important to verify that we are solving an index-1 DAE.

To verify numerically that (3.39) is nonsingular, we determined that the Jacobian with respect to the algebraic variables was full rank, and therefore, we have shown that our DAE is index-1.

3.5.2 The Methodology of `ode15s`

First, we derived a TCAT-based PDE model to simulate concentrated flow through porous media. Then, we rewrote the model as a system of PDAEs. Next, we used the method of

lines to reformulate the model as a large system of semi-explicit ordinary DAEs. Finally, we verified that the system of DAEs is index-1. The next step in this process is to solve for the time derivate y' . We accomplish this by integrating over time.

We run simulations in MATLAB, which has a suite of ODE solvers that solve the initial value problem

$$M(t)y' = F(t, y) \quad (3.40)$$

along the time interval $[t_0, t_f]$ with initial condition $y(t_0) = y_0$. In (3.40), $M(t)$ is the mass matrix [70]. The mass matrix can be singular, which occurs if we are solving a DAE.

Backward differentiation formulas (BDFs) are a well-known method for solving stiff problems. The formula for the k^{th} order BDF, also referred to as BDFk, is given by

$$\sum_{m=1}^k \frac{1}{m} \nabla^m y_{n+1} - hF(t_{n+1}, y_{n+1}) = 0, \quad (3.41)$$

where $\frac{1}{m} \nabla^m$ are optimally chosen coefficients that allow the solver to use the maximum order m [69] and F is the right-hand side of the DAE. One solves the implicit equation (3.41) with a simplified Newton method and arrives at a new approximation y_{n+1} . The use of BDF methods to solve DAEs was initially proposed by Gear [17] and has been analyzed by numerous researchers [8, 9, 16, 45]. Many of the classic codes, such as LSODI and DASSL, use BDFs [8] and vary the order of the BDF method so that the problem can be solved most efficiently [69]. However, the drawback to BDFs is that as the order increases, the stability decreases. Therefore, that MATLAB ODE suite uses a similar approach to the BDF method, but improves stability.

The MATLAB ODE suite uses numerical differentiation formulas (NDFs) instead of BDFs. The NDFs were derived by Gregory in 1957 [25] and later derived explicitly by Spitzbart and Macon [73]. The developers of the MATLAB ODE suite chose NDFs because they are more stable than BDFs [69].

A method of the form

$$\sum_{m=1}^k \frac{1}{m} \nabla^m y_{n+1} - hF(t_{n+1}, y_{n+1}) - \kappa \gamma_k (y_{n+1} - y_{n+1}^{(0)}) = 0 \quad (3.42)$$

is called a numerical differentiation formula. In (3.42), κ is a scalar and the γ_k 's are computed with

$$\gamma_k = \sum_{j=1}^k \frac{1}{j}. \quad (3.43)$$

The NDFs are more stable than the BDFs; however, the developers of the MATLAB ODE suite have determined optimal κ to take advantage of the accuracy of BDFs while exploiting the stability of NDFs [69].

The ode15s solver implements backward NDFs in order to integrate over time. The code performs a Newton iteration

$$y_{n+1}^{(i+1)} = y_{n+1}^{(i)} + \Delta^{(i)}. \quad (3.44)$$

The code finds the correction $\Delta^{(i)}$ by solving

$$\left(M - \frac{h}{(1-\kappa)\gamma_k} J \right) \Delta^{(i)} = \frac{h}{(1-\kappa)\gamma_k} F(t_{n+1}, y_{n+1}^{(i)}) - \psi - (y_{n+1}^{(i)} - y_{n+1}^{(0)}). \quad (3.45)$$

In (3.45), M is the mass matrix, J is a finite-difference approximation to the Jacobian of F , and ψ is given by [69]

$$\psi = \frac{1}{(1-\kappa)\gamma_k} \sum_{m=1}^k \gamma_m \nabla^m y_n. \quad (3.46)$$

In our model, the mass matrix M , as well as the Jacobian J are sparse. By taking advantage of the sparse structure of a matrix, we are able to avoid computations on zero entries. This saves us both computation time and memory [12, 19, 64]. We are able to use the MATLAB sparse command to indicate to the solver that our matrices are sparse [4].

Because we declare M and J to be sparse, the iteration matrix in (3.45), given by

$$\left(M - \frac{h}{(1-\kappa)\gamma_k} J \right) \quad (3.47)$$

is also sparse. Further, instead of storing a full matrix, MATLAB is able to store the matrix as three vectors. The first vector is the entries of the iteration matrix. The second and third vectors are the corresponding row and column, respectively, of the location of the nonzero entry [64]. Thus, exploiting the inherent sparsity pattern in this problem saves computation time and memory [12, 19, 64, 69].

Once we have stored the sparse iteration matrix, `ode15s` computes a sparse LU factorization [12]. From there, the solver performs a sparse direct solve to determine the step size $\Delta^{(i)}$. The solver continues until it finds a value of the step that passes its error checking routine, then continues moving forward in time until it either reaches the user-specified end time or fails.

3.5.3 Differential-Algebraic Equation Consistency

There is one remaining condition we need to satisfy before we can solve DAEs numerically, which is that the user-provided initial conditions must be consistent with the equation being solved. The formal definition of consistency is outlined below.

Definition 3.5.1. *A DAE has consistent initial conditions if the vector of initial conditions y_0 has the slope y'_0 that satisfies [68]*

$$M y'_0 = F(t_0, y_0). \quad (3.48)$$

In `ode15s`, the user has the option of supplying consistent initial conditions. However, if the user-provided initial conditions are inconsistent, the solver will attempt to find a vector \bar{y} that is both close to the user input and consistent [68]. The consistency condition is important, as a DAE with inconsistent initial conditions will not be able to initialize the simulation [70].

3.6 Sparse Jacobian Pattern

Here, we will derive the sparsity pattern for our Jacobian J that we use to determine our iteration matrix in (3.47). We know that we have a sparse Jacobian pattern, and we will exploit this in order to compute our solution more quickly.

For efficiency purposes, we have ordered our unknowns as

$$y = [(\omega^{A\bar{w}})_1, (p^w)_1, (\omega^{A\bar{w}})_2, (p^w)_2, \dots, (\omega^{A\bar{w}})_{N+1}, (p^w)_{N+1}]^T. \quad (3.49)$$

This means that, in our code, we organize F as

$$F = [\omega^{A\bar{w}}(0, t), p^w(0, t), (F_1)_2, (F_1 - F_2)_2, \dots, (F_1)_N, (F_1 - F_2)_N, \dots]^T. \quad (3.50)$$

We want to determine our Jacobian J such that

$$J_{ij} = \frac{\partial F_i}{\partial y_j}. \quad (3.51)$$

To determine our Jacobian pattern, we will first consider the case when i is odd in (3.51). Then, $F_i = (F_1)_{\frac{i+1}{2}}$ and $y_i = (\omega^{A\bar{w}})_{\frac{i+1}{2}}$. For simplicity, we shall let $\hat{i} = \frac{i+1}{2}$. From Section 3.4, we established that we write F_1 in the discrete form

$$(F_1)_{\hat{i}} = -\frac{1}{(\rho^w_{\hat{i}})'} \left(\frac{\rho^w_{\hat{i}} \epsilon^{\bar{w}} v_{\hat{i}}^{\bar{w}} - \rho^w_{\hat{i}-1} \epsilon^{\bar{w}} v_{\hat{i}-1}^{\bar{w}}}{h} \right). \quad (3.52)$$

Further, based on the difference schemes we used in Section 3.4, we know that

$$\rho^w_{\hat{i}} = \rho^w_{\hat{i}}(\omega^{A\bar{w}}_{\hat{i}}) \quad (3.53)$$

$$\epsilon^{\bar{w}} v_{\hat{i}}^{\bar{w}} = \epsilon^{\bar{w}} v_{\hat{i}}^{\bar{w}}(\omega^{A\bar{w}}_{\hat{i}}, p_{\hat{i}+1}^w, p_{\hat{i}}^w). \quad (3.54)$$

This means that

$$\rho^w_{\hat{i}} \epsilon^{\bar{w}} v_{\hat{i}}^{\bar{w}} = \rho^w_{\hat{i}} \epsilon^{\bar{w}} v_{\hat{i}}^{\bar{w}}(\omega^{A\bar{w}}_{\hat{i}}, \omega^{A\bar{w}}_{\hat{i}-1}, p_{\hat{i}+1}^w, p_{\hat{i}}^w) \quad (3.55)$$

and

$$\rho^w_{\hat{i}-1} \epsilon^{\bar{w}} v_{\hat{i}-1}^{\bar{w}} = \rho^w_{\hat{i}-1} \epsilon^{\bar{w}} v_{\hat{i}-1}^{\bar{w}}(\omega^{A\bar{w}}_{\hat{i}-1}, p_{\hat{i}}^w, p_{\hat{i}-1}^w). \quad (3.56)$$

Consequently, it follows that

$$(F_1)_{\hat{i}} = (F_1)_{\hat{i}}(\omega^{A\bar{w}}_{\hat{i}}, \omega^{A\bar{w}}_{\hat{i}-1}, p_{\hat{i}+1}^w, p_{\hat{i}}^w, p_{\hat{i}-1}^w). \quad (3.57)$$

We note that $(F_1)_{\hat{i}}$ is a function of five discrete variables. However, because of how we have chosen to order our unknowns, we will need to include a bandwidth of six.

The nonzero entries in the \hat{i}^{th} column of the Jacobian

$$J(\hat{i}, :) = [0, \dots, 0, \omega^{A\bar{w}}_{\hat{i}-1}, p_{\hat{i}-1}^w, \omega^{A\bar{w}}_{\hat{i}}, p_{\hat{i}}^w, \omega^{A\bar{w}}_{\hat{i}+1}, p_{\hat{i}+1}^w, 0, \dots, 0] \quad (3.58)$$

are denoted by their respective variable. Note that $\omega^{A\bar{w}}_{\hat{i}}$ corresponds with the diagonal entry of the Jacobian. Therefore, we need to indicate to our solver that we have a bandwidth of six with two subdiagonals and three superdiagonals.

To finish determining our bandwidth, we again consider the Jacobian

$$J_{ij} = \frac{\partial F_i}{\partial y_j} \quad (3.59)$$

and determine what our bandwidth must be when i is even. When i is even, our Jacobian entry becomes

$$J_{ij} = \frac{\partial (F_1 - F_2)}{\partial y_j}. \quad (3.60)$$

We repeat a similar process as we did when i was odd to find that

$$(F_1 - F_2)_i = (F_1 - F_2)_i(\omega^{A\bar{w}}_i, \omega^{A\bar{w}}_{i-1}, \omega^{A\bar{w}}_{i-2}, p_{i+1}^w, p_i^w, p_{i-1}^w, p_{i-2}^w). \quad (3.61)$$

Note that in (3.61), we have more terms than we had in (3.57). This is because we are differentiating the diffusion closure relations in conservative form, as discussed in Section 3.4.2.

The nonzero entries in the \hat{i}^{th} column of the Jacobian

$$J(\hat{i}, :) = [0, \dots, \omega^{A\bar{w}}_{i-2}, p_{i-2}^w, \omega^{A\bar{w}}_{i-1}, p_{i-1}^w, \omega^{A\bar{w}}_i, p_i^w, \omega^{A\bar{w}}_{i+1}, p_{i+1}^w, 0, \dots] \quad (3.62)$$

are denoted by their respective variables. Our bandwidth has now increased to eight, with nonzero entries on the diagonal, four subdiagonals, and three superdiagonals. Therefore, we specify to ode15s that there are nonzero entries on the diagonal, four subdiagonals, and three superdiagonals.

3.7 Solver Settings

The ode15s code provides users with several options that they can set before they initialize the solver [51]. We consider the initial step size. To initialize our code properly, we find that we must set the initial time step to a small value. Since we are solving a DAE, we need to have consistent initial conditions. Recall that this means

$$M y'_0 = f(t_0, y_0) \quad (3.63)$$

for our initial conditions y_0 . The DAE capabilities within `ode15s` allow the user to input an initial estimate for y_0 , and from there the code computes a consistent initial condition [68]. The initialization scheme within `ode15s` is well-defined only for a small initial step [70], therefore, setting a small initial step size allows our solver to compute consistent initial conditions and consequently allows the solver to move forward in time [68].

3.8 Modifications to `ode15s`

We made two changes to `ode15s`. The first change was to increase the number of Newton iterations the solver performs before deciding to reject the new predicted value y_{n+1} . The default in `ode15s` is 4 iterations; however, we found that we had solver failure with such a small number of iterations. We increased the number of Newton iterations to 20. This allows the solver time to adjust the rate of convergence and accept a viable step in lieu of rejecting the step too early.

The second change was to alter a norm. Deep within the file, the code calculates the norm of the $\Delta^{(i)}$, which is the difference between the predicted value y_{n+1} at time t_n and y_{n+1} at time t_{n+1} . It appears in (3.44).

The default, which is hard coded, is to compute the infinity norm of $\Delta^{(i)}$. However, when we are tracking a sharp front that moves forward in time, this is not the preferred choice of norm. In `ode15s`, the weighted infinity norm of the step size vector $\Delta^{(i)}$ is given by

$$\|\text{invwt} * \Delta^{(i)}\|_{\infty} = \max_j |\text{invwt} * \Delta^{(i)}|. \quad (3.64)$$

In (3.64), the norm of the step size is weighted by a quantity called `invwt`, which is defined as

$$\text{invwt} = 1 ./ \max(\max(\text{abs}(y), \text{abs}(y_{\text{new}})), \text{threshold}).$$

Here y is the old value of the solution, y_{new} is the update value of the solution ($y + \Delta^{(i)}$), and `threshold` is the absolute tolerance divided by the relative tolerance.

When we are tracking a sharp front, we see large changes at the location of the front, whereas at other locations, the derivatives are much smaller, or in many cases, zero. Thus, the infinity norm is not a good choice for our specific problem. We instead replaced the

infinity norm with the scaled 2-norm, which is given by

$$\|\text{invwt} * \Delta^{(i)}\|_2 = \left(\sum_{i=1}^N |\text{invwt} * \Delta^{(i)}|^2 \right)^{\frac{1}{2}}. \quad (3.65)$$

The 2-norm allows us to take information from all of the Newton steps.

3.9 Adaptive Gridding Technique and Grid Convergence

To obtain grid independent solutions, we must use a very fine grid to resolve the front in the TCAT model. However, using a very fine fixed grid means that simulation times for the model can be up to 1 day. To have a grid independent solution that runs in a realistic amount of time, we use the spatially adaptive method of lines (SAMOL) to resolve the spatial mesh.

3.9.1 Spatially Adaptive Method of Lines

The SAMOL method uses an adaptive grid in space. Grid point placement is determined based on $\frac{\partial \omega}{\partial z}$, the spatial gradient of mass fraction [78]. When $\|\frac{\partial \omega}{\partial z}\|$ is large, it shows us where the location of the front is, and where we need to place the most grid points. This algorithm was implemented in the TCAT model by Timothy Weigand at the University of North Carolina. The general steps are summarized in Algorithm 1.

The new grid is a function of the spatial gradient of the mass fraction. At the grid points where the front is located, the grid points are placed using $\Delta z = \frac{.005}{\|\frac{\partial \omega}{\partial z}\|_{\infty}}$. In the code, this is called `zfine`, and it accounts for the fine portion of the grid. The coarse grids on either side of the front are determined by using a slightly finer transition region to the left and right of the front and using a very coarse set approximation of $\Delta z = 0.5$ away from the front.

This adaptive gridding technique has built-in checks to ensure that the location of the front is within the computational domain. The code also utilizes transition regions to the left and right of the front. If the transition region goes outside the domain, it is parsed to keep all grid points within the computational domain.

The old solution is projected onto the new grid using a cubic `pchip` spline, which is a piecewise cubic Hermite spline. This is a built-in function in MATLAB. The time step is fixed as the amount of time that has elapsed in Step 2(d) of Algorithm 1. However, as we are

Algorithm 1 SAMOL Algorithm

1. Resolve ω and p on a reasonable grid.
 2. While $t < t_{end}$
 - (a) Use previous value of ω to determine location of the front and generate the new grid.
 - (b) Manually move the location of the front based on Darcy velocity.
 - (c) Project solution onto new grid using a cubic spline.
 - (d) Integrate (use ode15s) to advance to the next time step.
- endwhile
-

using ode15s, the solver will optimally determine the number of time steps needed within the prescribed interval.

Users can modify the simulation grid in a number of ways. First, one can change the initial starting grid. The prescribed time step in Step 2(d) of Algorithm 1 can be changed, but theoretically this should not affect the output as ode15s is a variable step solver. There is a safety factor, which can be set to keep the transition region from becoming too small. We can also set the minimum and maximum step sizes for the fine and coarse grids.

Now that we have discussed our adaptive gridding technique, we will perform a grid refinement study to show that the solutions are grid independent.

3.9.2 Grid Refinement Study

Our goal is to establish that our simulation grid converges to a solution as we increase the number of spatial grid points. To do this, we perform a grid refinement study. The general concept of a grid refinement study involves running simulations on grids that are designed to be progressively finer. From there, we look at the ratios of the norm of the error and determine whether the error terms are converging to the appropriate ratio.

Performing a grid refinement study for the TCAT model is challenging for two reasons. First, we do not have an exact solution to compare the output to. Second, the grid is nonuniform, so we need to use a method that accounts for noninteger grid refinement.

There are two methods that we can use to account for the fact that we do not have an exact solution. The first method is self-estimation, where the error is measured across successive grids. Consider a computed solution with step size h , denoted $U(h)$, and the exact solution \hat{u} . Self-estimation states that

$$U(h) - \hat{u} = U(h) - U\left(\frac{h}{2}\right) + U\left(\frac{h}{2}\right) - \hat{u}. \quad (3.66)$$

However, if $U(h)$ is a good approximation, then $U\left(\frac{h}{2}\right) - \hat{u}$ is very small and is negligible. Thus, we can discard $U\left(\frac{h}{2}\right) - \hat{u}$ in equation (3.66) and write the error with spatial step size h as

$$E(h) = U(h) - U\left(\frac{h}{2}\right). \quad (3.67)$$

We repeat this process with a number of different grids, then compute the ratio of the errors. The error ratio should converge to

$$R = \frac{\|E(h)\|_\ell}{\|E\left(\frac{h}{2}\right)\|_\ell} \approx 2^p, \quad (3.68)$$

where p is the order of the method. Since our spatial discretization for the derivative of mass fraction is first-order, we expect that $p = 1$ and the ratio of the errors should converge to 2.

The other method that we can use to account for a lack of an analytic solution is to use a fine-grid reference solution. Here, we run a solution on the finest grid computationally possible, and treat that value, denoted by \tilde{u} , as our exact solution. From there, we compute

$$E(h) = U(h) - \tilde{u}. \quad (3.69)$$

We again repeat this process with a number of different grids, and calculate an error ratio R . We should again have that $p = 1$ and expect R to converge to 2.

To overcome the difficulties due to noninteger grid refinement, we can use an estimate for p given by [47]

$$p \approx \frac{\log(\|E(h_1)\|_\ell / \|E(h_2)\|_\ell)}{\log(h_1/h_2)}, \quad (3.70)$$

where h_1 and h_2 are any two grid spacings. We also must perform interpolation to ensure that the error is evaluated at consistent points. We chose to interpolate to the coarsest grid h_1 to minimize interpolation error.

Here, we perform noninteger grid refinement using both self-estimation and a fine-grid reference solution. We will present results for three different norms in (3.70), $\ell = 1, 2, \infty$.

First, we present results from self-estimation. We start at the coarsest grid h_1 and run simulations all the way to the finest grid h_8 . Results are presented in Table 3.1. We expect $p \rightarrow 1$ as the grid becomes finer.

We note that the order p appears to converge but stagnates around 0.78 in all three norms. This result could be for a number of reasons. First, in nonlinear problems, we often do not see convergence to exact integer values due to the nonlinear nature of the problem. Second, the interpolation can introduce error that prevents us from seeing convergence to the exact order of the method. Figure 3.1 is a plot of the number of grid points versus the logarithm of the error. As we observed in Table 3.1, we see convergence that is sublinear at first, then becomes linear as the grid becomes finer.

Next, we will investigate grid convergence using a reference solution. We ran our reference solution on the finest grid that we could reasonably simulate, given by $N \approx 5.5 \times 10^4$ grid points. Results of the reference solution grid refinement study are shown in Table 3.2. We can see that as we approach the h_7 and h_8 grids, the results have slightly overshoot the target of $p = 1$, indicating that the reference solution grid refinement study has converged. Figure 3.2 shows the number of grid points versus the error. We again see that the error decreases rapidly at the beginning, then appears to become linear as the grid becomes very large.

Table 3.1: Grid refinement study results from self-estimation.

Grid	p (1-norm)	p (2-norm)	p (∞ -norm)
h_1	0.68203	0.58938	0.48790
h_2	0.71313	0.65058	0.57986
h_3	0.73847	0.70168	0.65205
h_4	0.76426	0.73993	0.71035
h_5	0.78492	0.76604	0.74923
h_6	0.79651	0.78098	0.77222
h_7	0.78467	0.78467	0.78132
h_8	0.78501	0.77838	0.78157

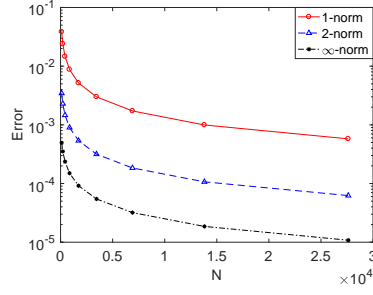


Figure 3.1: Results of the self-estimation grid study in three different norms.

Table 3.2: Grid refinement study results from the reference solution.

Grid	p (1-norm)	p (2-norm)	p (∞ -norm)
h_1	0.72596	0.65809	0.58168
h_2	0.75541	0.71141	0.65788
h_3	0.78526	0.75824	0.72522
h_4	0.820116	0.80172	0.78178
h_5	0.86427	0.85051	0.84128
h_6	0.93284	0.92249	0.91717
h_7	1.0691	1.0623	1.0612
h_8	1.0602	1.0561	1.0590

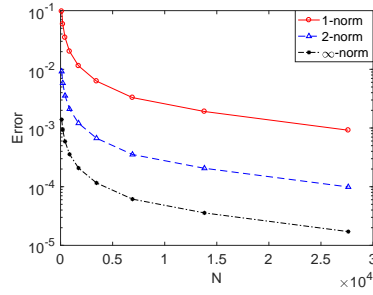


Figure 3.2: Results of the reference solution grid study in three different norms.

Here, we have shown that our solution converges via both self-estimation and the reference solution. While we found that p stagnated around 0.78 for self-estimation, we believe the grid has converged and this is an artifact of interpolation error or the challenging, nonlinear nature of the problem. We were able to show that p approaches 1 using the reference solution, and therefore we have achieved grid convergence.

CHAPTER

4

PARAMETER ESTIMATION

In Chapter 3, we discussed the numerical issues associated with simulating the TCAT model. Our goal in this chapter is to calibrate the model to experimental data by estimating parameters using the nonlinear least squares algorithm Levenberg-Marquardt. To implement the Levenberg-Marquardt algorithm, we use MATLAB's routine `lsqnonlin`.

4.1 The Levenberg-Marquardt Algorithm

Our objective is to employ the Levenberg-Marquardt algorithm to estimate model parameters so that the TCAT model fits experimental data. To achieve this, we consider the nonlinear least squares problem

$$f(x) = \frac{1}{2} \sum_{i=1}^N |r_i(x)|^2 = \frac{1}{2} R(x)^T R(x), \quad (4.1)$$

where $R(x) = (r_1(x), \dots, r_N(x))$ is the residual and $r_i(x) = f_i(x) - d_i$, where f_i is the model output and d_i is the experimental data.

The function $f(x)$ has a fixed but unknown value x^* that minimizes the residual $\frac{1}{2}R(x)^T R(x)$. We find x^* using an iterative method where we calculate a new estimate x_+ at each iteration. The formula for the new iterate x_+ is

$$x_+ = x_c + s, \quad (4.2)$$

where x_c is the current iterate and s is the step size. The iteration terminates when $f(x_+) < \text{tol}$, where tol is a user-specified termination tolerance. Once the iteration has successfully terminated, $x^* = x_+$.

We now discuss how we calculate the step size s . Since we want $f(x)$ to decrease, we search for our new iterate along a ray from x_c in a descent direction where f is locally decreasing as formally defined here.

Definition 4.1.1. A vector $d \in \mathbb{R}^N$ is a descent direction for f at x if [39]

$$\left. \frac{df(x + td)}{dt} \right|_{t=0} = \nabla f(x)^T d < 0. \quad (4.3)$$

The Newton descent direction is

$$-(R'(x)^T R'(x))^{-1} R'(x) R(x), \quad (4.4)$$

where

$$\nabla f = R'(x)^T R(x). \quad (4.5)$$

The condition (4.4) is only a descent direction if $R'(x)$ is full rank. This yields the Gauss-Newton iteration

$$x_+ = x_c - (R'(x)^T R'(x))^{-1} R'(x) R(x). \quad (4.6)$$

A damped Gauss-Newton method includes the addition of the Armijo rule [39]. The Armijo rule is a line search, where one searches along the proper descent direction, but reduces the step size s until there is sufficient decrease in the norm of the gradient.

Whereas (4.6) can be very good for some problems, the convergence analysis shows that the Jacobians $R'(x)$ and $R'(x)^T$ must be uniformly bounded and well-conditioned for the algorithm to terminate successfully [39].

The Levenberg-Marquardt method [46, 50] addresses these issues by adding a parameter $\nu > 0$ to $R'(x)^T R'(x)$, so that (4.6) becomes

$$x_+ = x_c - (\nu I + R'(x)^T R'(x))^{-1} R'(x) R(x). \quad (4.7)$$

Here ν is called the Levenberg-Marquardt parameter [39, 58].

We consider the two parameters in the TCAT model that will be estimated based on the incoming concentration of the brine. The first parameter is \hat{D}^{AB} , which quantifies the effective dispersion in the experimental column. This parameter is estimated for each of the three tracer experiments. For the higher concentration simulations, which are defined as having an incoming concentration greater than 10^{-6} , we estimate the parameter $\log(c_1)$. This parameter also appears in our spatially varying diffusion calculation. For these higher concentration simulations, we consider two types of experiments. In the first, which we denote as Case One, a brine of known concentration is pumped into a column consisting of sand and fresh water. In the second, which we denote as Case Two, a brine of known concentration is pumped into a column that contains sand and a resident brine of a lower concentration.

4.2 Tracer Parameter Estimation

All three tracer experiments have an incoming concentration of 10^{-6} and are run through the column prior to any higher concentration solution entering the column. The quantity \hat{D}^{AB} is used in our spatially varying diffusion relation (3.14). Once the tracer breakthrough curve is generated for a specific column, we use `lsqnonlin` to determine the optimal value of \hat{D}^{AB} . After that, \hat{D}^{AB} is treated as a constant value in (3.14) and the value used in each simulation depends which column and tracer were used to generate the high concentration breakthrough curve.

Optimal values for \hat{D}^{AB} for different experiments are tabulated in Table 4.1. We estimate the logarithm of \hat{D}^{AB} because the numerical value of \hat{D}^{AB} is very small, e.g., on the order of 10^{-11} . In Table 4.1, the first column denotes the number of the laboratory column that holds the porous media. The second column indicates the tracer number. Both of these columns are used to organize the data. The third column is the value of $\log(\hat{D}^{AB})$, which is what we explicitly estimate with the Levenberg-Marquardt algorithm. The fourth column is simply the third column, exponentiated, so that we can see the true value of \hat{D}^{AB} .

Table 4.1: Effective dispersion estimations for the three different tracers.

Column	Tracer	$\log(\hat{D}^{AB})$	\hat{D}^{AB}
11	1	-23.3650	7.1238×10^{-11}
11	2	-23.2710	7.8259×10^{-11}
12	1	-23.4290	6.6821×10^{-11}

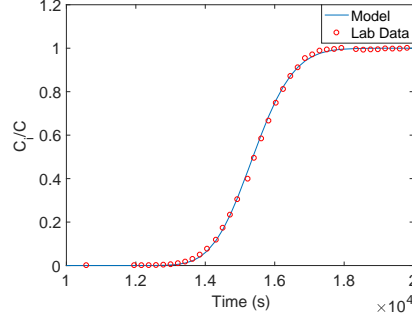


Figure 4.1: Model and experimental breakthrough curves for a tracer experiment.

Figure 4.1 shows the fit to lab data for a selected tracer experiment—Column 11, Tracer 1. We see that the model provides an accurate estimation of the tracer data.

Now that we have determined values of \hat{D}^{AB} to use for each column and tracer, we can estimate parameters for the higher concentration experiments.

4.3 High Concentration Parameter Estimation

Recall that any experiment that is not a tracer and has an incoming mass fraction greater than 10^{-6} is deemed a high concentration experiment. The available high concentration experimental data ranges in incoming mass fractions from $\omega^{A\bar{w}}_{in} \in [.0125, .4]$. Our first goal is to choose three experiments along the lower to middle range of this interval with which we can estimate a single value of $\log(c_1)$. We then perform forward simulations with this global value of $\log(c_1)$ and investigate the robustness of our model. Finally, we present a local parameter estimation where we estimate one experiment at a time and compare the

optimal values of $\log(c_1)$ as well as residual norms. Note that because of the large magnitude of c_1 , we estimated $\log(c_1)$ in all cases.

4.3.1 Global Estimation for $\log(c_1)$

From earlier work with the TCAT model, as discussed in Section 2.5, we determined it was reasonable to estimate a single parameter for a set of three experiments that are on the low to middle end of available data, with incoming mass fractions ranging from 0.01254 to 0.1508. We made the decision to choose these experiments due to the difficulty we experienced fitting some of the highest concentration experiments. As the incoming concentration increases, the problem becomes more stiff and the experimental data is consequently more difficult to fit to the model. We will demonstrate this rigorously in Sections 4.3.2 and 4.3.3. The experiments included in the parameter estimation are tabulated in Table 4.2. Here, $\omega^{A\bar{w}}_{in}$ is the incoming mass fraction and $\omega^{A\bar{w}}_{col}$ is the resident mass fraction in the column at $t = 0$. Note that for this investigation, we used only experiments from Case One, where $\omega^{A\bar{w}}_{col} = 0$.

We ran `lsqnonlin` for $\log(c_1)$ and found that the value of

$$\log(c_1) = 32.785 \quad (4.8)$$

yields the smallest residual across our three selected experiments. Residual norms are listed in Table 4.2. We can see that with the sharp increase in concentration from experiment B7 to experiment B14, the residual norm also greatly increases. This indicates that the global choice of $\log(c_1)$ does not yield model results that are as close to data as the incoming concentration increases. The model breakthrough curves fit to data are provided

Table 4.2: Experiments considered for the global parameter estimation.

Column	Tracer	Experiment Number	Incoming Mass Fraction $\omega^{A\bar{w}}_{in}$	Column Mass Fraction $\omega^{A\bar{w}}_{col}$	Residual Norm
11	2	B11	0.01254	0	4.3042×10^{-3}
11	2	B7	0.02541	0	3.97864×10^{-3}
11	2	B14	0.1508	0	1.7443×10^{-1}

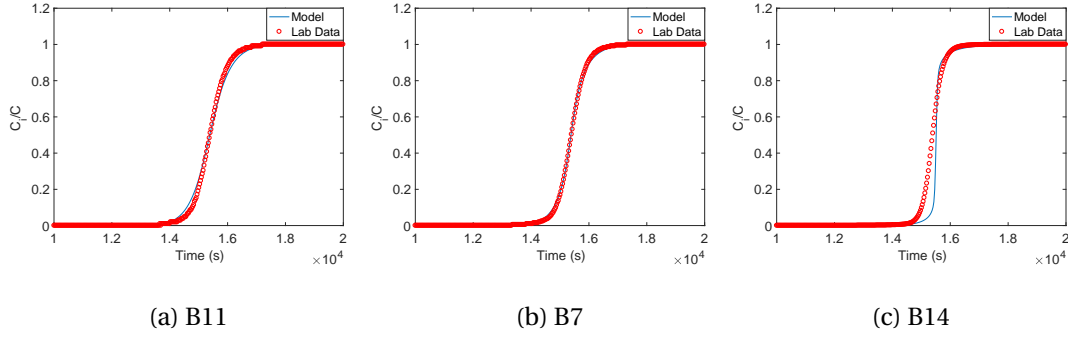


Figure 4.2: Breakthrough curves for the three selected experiments, B11, B7, and B14.

in Figure 4.2. We again see that the two lower concentration simulations appear to closely match the data, whereas B14 fails to capture the initial behavior as the front enters the column.

4.3.2 Forward Simulations for Global $\log(c_1)$

Next, we employed the optimal value of $\log(c_1) = 32.785$ and ran forward simulations for other high concentration experiments. The experiments considered as well as the residual norms for the forward simulations are listed in Table 4.3.

Table 4.3: Selected experiments for the forward simulations.

Column	Tracer	Experiment Number	Incoming Mass Fraction $\omega^{A\bar{w}}_{in}$	Column Mass Fraction $\omega^{A\bar{w}}_{col}$	Residual Norm
11	2	B8	0.1003	0	7.8486×10^{-2}
11	1	B6	0.2009	0	1.7680×10^{-1}
11	2	B9	0.4012	0	4.3386×10^{-1}
12	1	B2	0.2716	0.1508	7.7533×10^{-2}
12	1	B3	0.3706	0.2716	8.8850×10^{-2}
12	1	B4	0.4547	0.3706	1.3689×10^{-1}
11	2	B19	0.5267	0.4550	1.8850×10^{-1}

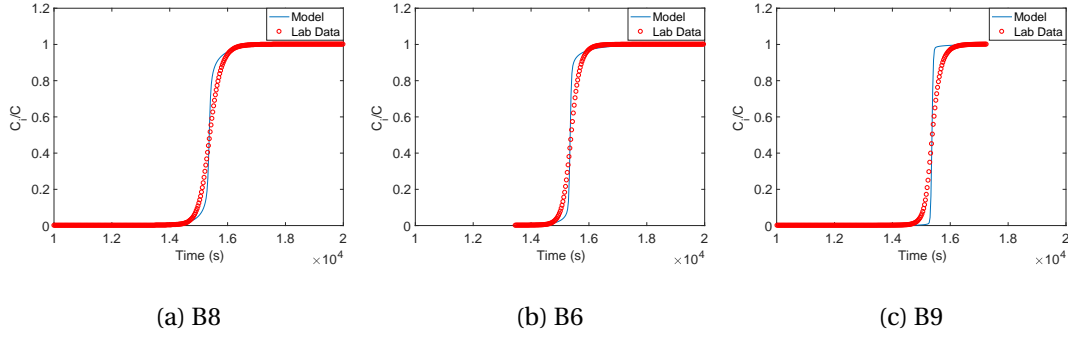


Figure 4.3: Forward simulations of breakthrough curves for Case One, where there is fresh water in the column at the start of the experiment.

Figure 4.3 shows the forward simulations compared to experimental data for the experiments in Case One, where there is fresh water in the column at the start of the experiment. We see that as the incoming concentration increases, the curves generated with the global value of $\log(c_1)$ become sharper. While the curve generated for B8 is only moderately sharper than the data, the curve for B9 is far too sharp to replicate data behavior. This observation, along with the fact that the residual norms increase as the incoming concentration increases, shows that $\log(c_1)$ is a sharpening parameter that has more pronounced effects on the curve as the incoming concentration increases. Therefore, this investigation seems to indicate that smaller values of $\log(c_1)$ need to be used to accurately simulate the highest concentration experiments.

Figure 4.4 shows the forward simulations compared to model data for the selected experiments in Case Two, where there is a lower concentration resident brine in the column at the start of the experiment. Here, we again see that the residual norms increase with an increase in incoming concentration; however, for experiments in Case Two, the increase in $\log(c_1)$ has an opposite effect and tends to smooth the curve as its value increases. From the tabulated values in Table 4.4, we determine that the optimal value of $\log(c_1)$ increases as the incoming concentration increases, whereas the residual norms are close in value. This smoothing phenomena appears to be due to the fact that the difference between $\omega^{A\bar{w}}_{in}$ and $\omega^{A\bar{w}}_{col}$ decreases as the incoming concentration increases. Therefore, a higher value of $\log(c_1)$ is needed to replicate the behavior of the data as incoming concentration increases.

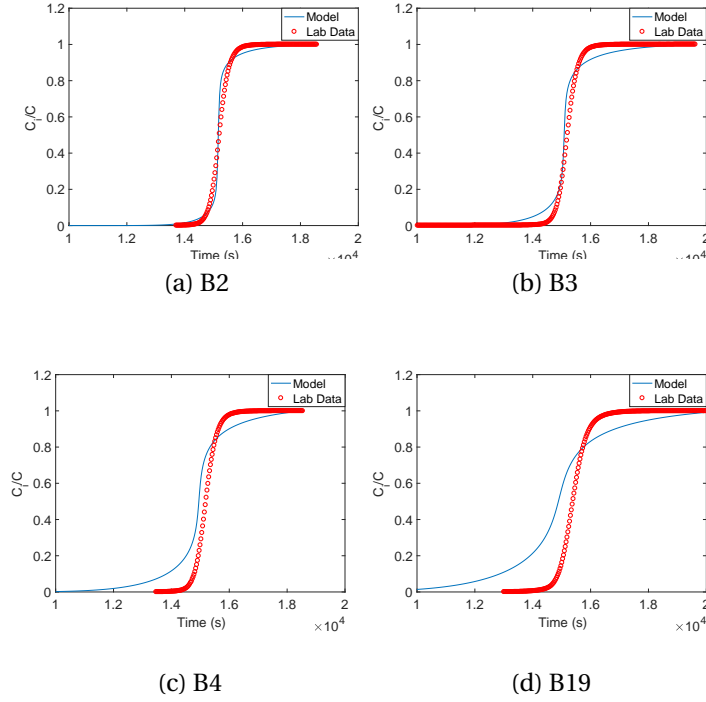


Figure 4.4: Forward simulations of breakthrough curves for Case Two, where there is a resident brine in the column at the start of the experiment.

This again shows that a single value of $\log(c_1)$ is not adequate to estimate all experimental data.

4.3.3 Individual Parameter Estimation

To investigate how the parameters change with incoming concentration, we next estimated individual parameters for the experiments considered in Sections 4.3.1 and 4.3.2. The experiments used for the individual least squares simulations are listed in Table 4.4, along with the optimal values of $\log(c_1)$ and the residual norms. From Table 4.4, we see that for the first four experiments in Case One, which are also the first four experiments listed in Table 4.4, the optimal value of $\log(c_1)$ decreases as the incoming concentration increases. For the remaining two experiments in Case One, we observe that because the incoming concentrations of B6 and B9 are so high, the model has difficulty replicating the

Table 4.4: Optimized parameter values for selected experiments.

Column	Tracer	Experiment Number	Incoming Mass Fraction $\omega^{A\bar{w}}_{in}$	Column Mass Fraction $\omega^{A\bar{w}}_{col}$	$\log(c_1)$	Residual Norm
11	2	B11	0.01254	0	33.312	1.2303×10^{-3}
11	2	B7	0.02541	0	32.854	3.6947×10^{-3}
11	1	B8	0.1002	0	31.921	3.9206×10^{-2}
11	2	B14	0.1508	0	31.790	8.7852×10^{-2}
11	1	B6	0.2009	0	31.955	1.4479×10^{-1}
11	2	B9	0.4012	0	33.280	4.1103×10^{-1}
12	1	B2	0.2716	0.1508	32.564	7.3970×10^{-2}
12	1	B3	0.3706	0.2716	33.206	7.6935×10^{-2}
12	1	B4	0.4547	0.3706	33.816	7.7136×10^{-2}
11	2	B19	0.5267	0.4550	34.177	7.4622×10^{-2}

experimental behavior and tends towards oversharpening to minimize the residual norm. We also see that the residual norm increases as the incoming concentration increases, indicating the higher concentration experiments are more difficult to estimate.

The experiments from Case Two in Table 4.4 show that the optimal value of $\log(c_1)$ increases as the incoming concentration increases. This verifies what we saw in Section 4.3.2 for Case Two, where the decrease in difference between the incoming and resident brine concentrations means that $\log(c_1)$ must increase with incoming concentration in order to best fit the data. The residual norms are close in value, indicating that all of the experiments are moderately difficult to estimate.

Figure 4.5 shows the model and experimental breakthrough curves for Case One. We can visually verify that the residual norms increase as the incoming concentrations increase, as quantified in Table 4.4.

Figure 4.6 shows the model and experimental breakthrough curves for Case Two. We again notice that as the incoming concentration increases, the model seems to become sharper than the data as the optimizer selects larger optimal values of $\log(c_1)$.

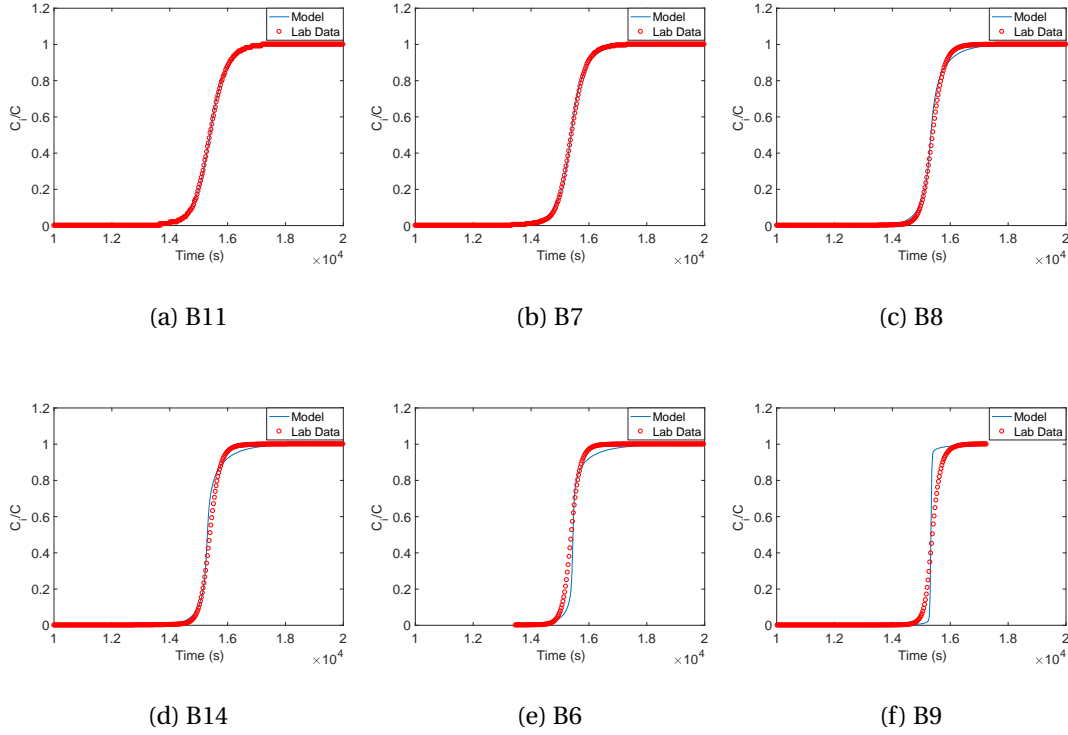
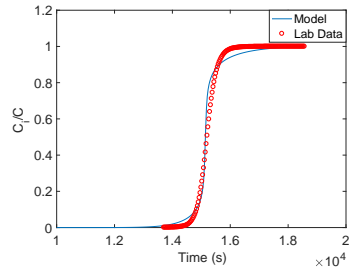


Figure 4.5: Individually estimated breakthrough curves for Case One.

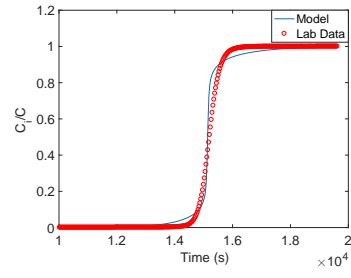
4.4 Conclusions

These investigations show that a single value of $\log(c_1)$ is not adequate to describe all available experimental data. As incoming mass fractions increase from 0.01254 to 0.4012 for Case One, we see that we need varying values of $\log(c_1)$ to describe the experimental data. For incoming mass fractions from 0.01254 to 0.1508, the optimal value of $\log(c_1)$ decreases; however, it begins to increase as incoming mass fractions become greater than 0.20. For Case Two, the optimal value of $\log(c_1)$ increases from 32.564 to 34.177 as the incoming mass fractions increase from 0.2716 to 0.4550. The modeled breakthrough curves become more smooth as incoming concentration increases for Case Two. Therefore, a larger value of $\log(c_1)$ is needed to replicate experimental behavior as incoming mass fraction increases.

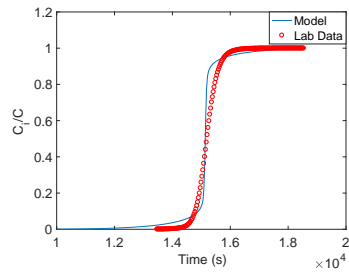
In Chapter 5, we will select four experiments from Case One and develop a surrogate model used for Bayesian inference.



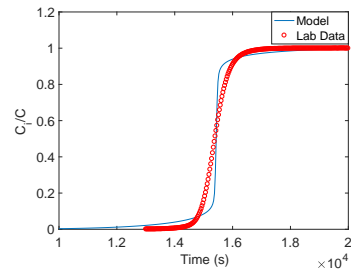
(a) B2



(b) B3



(c) B4



(d) B19

Figure 4.6: Individually estimated breakthrough curves for Case Two.

CHAPTER

5

SURROGATE MODELING

In this chapter, we construct a surrogate model for TCAT, which we will later use for Bayesian parameter estimation.

5.1 Justification for Constructing a Surrogate Model

The numerical techniques used for parameter estimation in the previous chapter require costly function evaluations, typically on the order of 15-20 minutes for a single evaluation. Despite the fact that we are implementing a 1-D model, the fine grid that we must use to achieve grid independence, as well as the stiffness of the problem, present a challenge to our numerical solver. Our goal in this chapter is to develop a surrogate model to overcome long run times so that we can implement parameter estimation and uncertainty quantification for the TCAT model using Bayesian inference.

As we will discuss in Chapter 6, Bayesian inference involves determining a posterior distribution for the parameters. For most problems, this distribution cannot be determined analytically; therefore, we will use a Delayed Rejection Adaptive Metropolis

(DRAM) algorithm to simulate the posterior distribution. To obtain chain convergence, DRAM typically requires tens of thousands of function evaluations.

For problems and models with long run times, DRAM simulations could potentially take weeks or months, therefore rendering Bayesian inference techniques infeasible [71]. To address this problem, we construct and verify a surrogate model for TCAT. The purpose of a surrogate model is to replicate the response of a computationally expensive numerical model in a computationally more efficient manner. This surrogate model will allow us to perform uncertainty quantification on the TCAT model by providing reasonable simulation times.

In many cases, the first step in creating a surrogate model is to eliminate parameters that are not identifiable or influential [66].

Definition 5.1.1. *A parameter q is identifiable at value q^* if, given the mapping $y = f(q)$, $q = [q_1, \dots, q_p]$, $f(q) = f(q^*)$ implies that $q = q^*$ for all $q \in \mathbb{Q}$, where \mathbb{Q} is the space of admissible parameters. If q is identifiable only in a subset of the parameter space denoted $I(q)$, then $I(q)$ is called the identifiable subspace.*

A parameter that fails to be identifiable is termed unidentifiable. Moreover, we consider noninfluential parameters that do not significantly impact model output. We define the space $NI(q)$ of noninfluential parameters as $|f(q) - f(q^*)| < \epsilon$ for all q and q^* contained in $NI(q)$ [66].

Parameters that are nonidentifiable or noninfluential cannot be uniquely determined by either the least squares parameter selection techniques of Chapter 4 or the Bayesian inference techniques we will discuss in Chapter 6 that utilize a noninformative prior.

Therefore, before constructing a surrogate model, one must determine which subset of model parameters are identifiable. There are both local and global sensitivity methods to determine whether a parameter is identifiable. Local sensitivity methods perturb parameters about a fixed value and measure the change in model response. Global sensitivity methods determine the effect of parameters over the entire range of admissible parameter values, and include Sobol analysis and Morris screening [66].

Active subspace methods consist of algorithms that determine parameters that most strongly influence the model response. For linear problems, active subspace methods involve the singular value decomposition (SVD) as well as randomized algorithms. Nonlinear active subspace methods are more challenging and include variance analysis and model linearization. For more details on these methods, see [71]. Unlike active

subspace methods, parameter subset selection ranks the most influential $n_p \leq p$ parameters but does not determine which parameters are more influential than others within the subset [66].

Recall that in Section 2.5, we gave a discussion of the history and evolution of the TCAT model. The model presented in Chapter 3, and considered in this dissertation, was reduced from two parameters to one parameter, c_1 . In the previous chapter, we showed that $\log(c_1)$ is identifiable. Therefore, we have shown that we have a model with one identifiable parameter, and we can create a surrogate model for TCAT.

5.2 Surrogate Model Classification

There are different types of surrogate models, including regression-based models, interpolation-based models, projection-based models, and hierarchical models. Regression and interpolation-based models are also generally called response surface models, which indicate that they are based upon fits to model response data [71].

We consider the statistical model

$$\Upsilon_i = f_i(q) + \epsilon_i, \quad i = 1, \dots, n, \quad (5.1)$$

where Υ_i are random variables with realizations v_i . The parameter-dependent model $f(q)$ yields a model response based on input parameters q . Random variables ϵ_i describe observative errors between the model response and measurements. We assume ϵ_i are independently and identically distributed (iid) with $\epsilon_i \sim N(0, \sigma^2)$. These errors are generally associated with the high-fidelity model and not the surrogate approximate. The model (5.1) allows us to construct surrogates while accounting for observative error.

The goal of a response surface model is to use model responses v_i associated with model input q to construct an approximation or emulator $\tilde{f}(q)$ of the original model $f(q)$. The emulator $\tilde{f}(q)$ allows us to construct model responses for values of q not directly simulated by the original model.

Both regression and interpolation-based models treat the original model as a black box, where the user calls the model repeatedly, often with randomly determined inputs in a prescribed interval. Regression-based models emulate the trends in the model output as a regression relationship between a simple model with a set of fixed but unknown coefficients. In many applications, polynomial surrogates are used, and the linear

coefficients are determined by least squares parameter estimation between the surrogate output and the output from the full model [1]. In this dissertation, we will use a regression-based model to construct the surrogate.

When constructing an interpolation-based model, one constructs a relationship between the input and the output by interpolating the responses from the code using appropriate weights [1, 49].

To construct projection-based models, also called reduced order models, one projects responses and parameters onto a lower-order subspace, using techniques such as proper orthogonal decomposition or high-dimensional model representation. More information on reduced order models can be found in Chapter 13 of [71]. Finally, to construct hierarchical models, one typically modifies the model in a way that reduces simulation times, including coarsening grids or simplifying physical assumptions. For more information on hierarchical models, see [13].

5.3 Regression-Based Model Construction

In this section, we summarize the construction of a regression-based model.

The general regression-based model is given by

$$f(q) = \tilde{f}(q) + \epsilon, \quad (5.2)$$

where $f(q)$ is the expensive simulation output, $\tilde{f}(q)$ is the surrogate model output, which is efficient to evaluate, and ϵ is the error in the surrogate model response. In regression-based surrogate models, $\tilde{f}(q)$ is often taken to be a polynomial or spline-based model [59].

The example we will give in the next section utilizes polynomial regression where coefficients are fit to model responses using a least squares algorithm. We begin by considering linear, quadratic, and cubic polynomial surrogate models. The linear model is

$$\tilde{f}(q) = \beta_0 + \sum_{i=1}^n \beta_i q_i + \epsilon, \quad (5.3)$$

whereas quadratic polynomials have the form

$$\tilde{f}(q) = \beta_0 + \sum_{i=1}^n \beta_i q_i + \sum_{i=1}^n \sum_{j \geq i}^n \beta_{ij} q_i q_j + \epsilon. \quad (5.4)$$

The cubic model is

$$\tilde{f}(q) = \beta_0 + \sum_{i=1}^n \beta_i q_i + \sum_{i=1}^n \sum_{j \geq i}^n \beta_{ij} q_i q_j + \sum_{i=1}^n \sum_{j \geq i}^n \sum_{k \geq j}^n \beta_{ijk} q_i q_j q_k + \epsilon. \quad (5.5)$$

Here, $\tilde{f}(q)$ is the response of the polynomial model plus an iid error model response with $\epsilon_i \sim N(0, \sigma^2)$. The terms β_0 , β_i , β_{ij} , and β_{ijk} are the unknown polynomial coefficients. The q_i , q_j , and q_k terms are components of the input parameter values. We use ordinary least squares to compute the unknown values of β_0 , β_i , β_{ij} , and β_{ijk} [1].

In addition to using polynomials to construct a regression-based model, one can also consider other forms of the surrogate model. When we construct the surrogate model for TCAT, we use a piecewise cubic spline. To construct our spline, we use the MATLAB function `spline`. The function `spline` automatically breaks up the model response into intervals, then uses a cubic polynomial whose intercept is the realization of the left boundary of the spline interval to approximate the surrogate model. The coefficients of the polynomial are again fit using least squares [56].

When building the surrogate model for TCAT, a potential future direction for this project includes investigating the use of logistic regression as the relative concentration is bounded between 0 and 1 and is increasing. One could attempt to fit the shape parameters for different experiments and determine if the logistic model is able to outperform the cubic spline surrogate model.

5.4 Surrogate Example

To illustrate a regression-based surrogate model, we consider the function

$$f(q) = \exp(0.7q_1 + 0.3q_2). \quad (5.6)$$

We will construct a polynomial regression-based model, $\tilde{f}(q)$ to approximate this function of two variables.

In many cases, a quadratic polynomial can quantify the physical response and is a natural choice for optimization. To avoid overfitting, one should check the Bayesian information criterion (BIC)

$$\text{BIC} = k \log(n) - 2 \log(L) \quad (5.7)$$

or the Akaike information criterion (AIC)

$$\text{AIC} = 2k - 2\log(L). \quad (5.8)$$

In both of these equations, k is the order of the polynomial, n is the number of observations, and L is the sum of squares likelihood. The goal is to choose the order k that yields the smallest AIC and BIC [49].

For data, we will generate synthetic responses for q_1 and q_2 sampled from the interval $[-1, 1]$ with $\epsilon_i \sim N(0, 0.1^2)$. To generate the training points, we use the `rand` function in MATLAB. We then use the AIC and BIC to determine the optimal order of the response surface model. For this case, the AIC and BIC results for polynomials of order 1 to 5 are compiled in Table 5.1. Note that both criteria show that the optimal order of the polynomial is 2 since this produces the lowest AIC and BIC scores.

Since the optimal surrogate model is second-order, it will have six coefficients and takes the form

$$\tilde{f}(q, \beta) = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \beta_{12} q_1 q_2 + \beta_{11} q_1^2 + \beta_{22} q_2^2. \quad (5.9)$$

The optimal coefficients as determined by `polyfitn` are computed in Table 5.2.

Next, we generate another set of 300 test points with a known variance of $\sigma = 0.1$ to verify the accuracy of the response surface model. The fit of the response surface versus the test points is shown in Figure 5.1. The root mean square error takes on a value of 0.09567, indicating a good fit between the response surface model and the test points.

Table 5.1: Information criterion comparison for the (5.6).

Polynomial Order	Akaike Information Criterion	Bayesian Information Criterion
1	-1.14×10^2	-1.07×10^2
2	-1.82×10^2	-1.67×10^2
3	-1.76×10^2	-1.50×10^2
4	-1.67×10^2	-1.28×10^2
5	-1.61×10^2	-1.06×10^2

Table 5.2: Coefficients of the second-order response surface model for (5.6).

Coefficient	Value
β_0	0.3081
β_1	0.2311
β_2	0.7399
β_{12}	0.05248
β_{11}	0.3414
β_{22}	0.9746

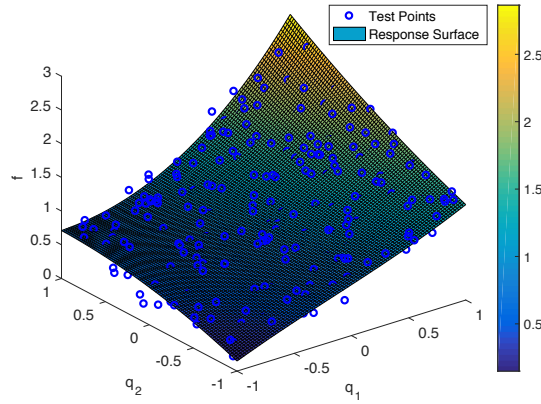


Figure 5.1: Response surface model and test points for (5.6).

5.5 Constructing a Surrogate Model for TCAT

Here, we illustrate the construction of a regression-based surrogate model for the TCAT model. We construct the surrogate by generating sets of training and test points. Two sets of 300 values of $\log(c_1)$ were randomly selected from the interval $[0, 35]$, which is the range of feasible inputs for the model. Note that as in Chapter 4, we used the natural logarithm of c_1 due to its large magnitude.

After the test and training points were selected, we ran the model for each value of $\log(c_1)$ and stored the output—in this case, a breakthrough curve giving the relative concentration over time at the top of the column. Whereas this was a computationally intensive effort, we

utilized high-performance computing resources, including MATLAB's parallel computing toolbox, to reduce simulation times.

Once the test and training points were stored with appropriate model responses, we use the set of training points to determine the surrogate model response. A surrogate for the lowest concentration non-tracer experiment, Column 11, B11, was first generated to reduce simulation time. This simulation yields the relative concentration at $n = 525$ output times. As some of the behavior of the breakthrough curve is less interesting, for instance, before the front reaches the top of the column or at steady state flow when the relative concentration is 1, we chose to begin to investigate our regression-based model during breakthrough so that we could see variability between the responses.

We constructed a response surface model using a polynomial approximation. In the case of the TCAT model, we found that $k = 2$ minimizes these criteria. The results of the AIC and BIC are tabulated in Table 5.3.

Since the TCAT model has only one parameter ($p = 1$), the quadratic response surface model is

$$\tilde{f}(q, \beta) = \beta_0 + \beta_1 q + \beta_{11} q^2, \quad (5.10)$$

where $q = \log(c_1)$ is the parameter of interest. The values of $\beta_0, \beta_1, \beta_{11}$ were determined using `polyfit` in MATLAB [52] and are shown in Table 5.4. From Table 5.4, we determine that both β_0 and β_1 are small and the polynomial is dominated by the quadratic term.

Despite being the optimal choice from the AIC and BIC, we determined that a quadratic polynomial did not have the correct characteristics to accurately emulate the TCAT model. As illustrated in Figure 5.2, the surrogate does not correctly approximate the behavior of the TCAT model.

Table 5.3: Information criterion comparison for the TCAT model.

Polynomial Order	Akaike Information Criterion	Bayesian Information Criterion
1	-1.81	5.60
2	-2.05	1.09
3	1.28	1.60
4	2.64	2.11
5	3.90	2.61

Table 5.4: Coefficients of the second-order response surface model for TCAT.

Coefficient	Value
β_0	1.0137×10^{-14}
β_1	-2.6803×10^{-13}
β_2	1.2540×10^{-2}

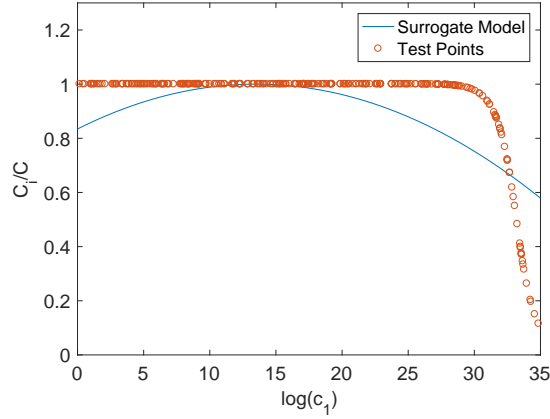


Figure 5.2: Polynomial response surface model and test points for TCAT.

Based on the shape of the data, a polynomial of any order cannot accurately represent the dynamics of the TCAT model response as $\log(c_1)$ increases. The model response is almost constant until it has a sharp logarithmic-like decline towards the end of the range of identifiable $\log(c_1)$. Due to this behavior in the model response, we instead fit a cubic spline to the model response, using `spline` in MATLAB. Utilizing a spline in place of the typical polynomial is increasingly popular in the surrogate literature, including [28, 74].

In Figure 5.3, we illustrate the response surface model fit with a cubic spline surrogate versus the test points for $t = 1.32 \times 10^4$ seconds, which is during breakthrough. We observe that the cubic spline generates a much better fit to data for the response surface model applied to TCAT. The root mean square error is much improved at a value of 8.56×10^{-7} , as compared to 0.136 for the quadratic model (5.10).

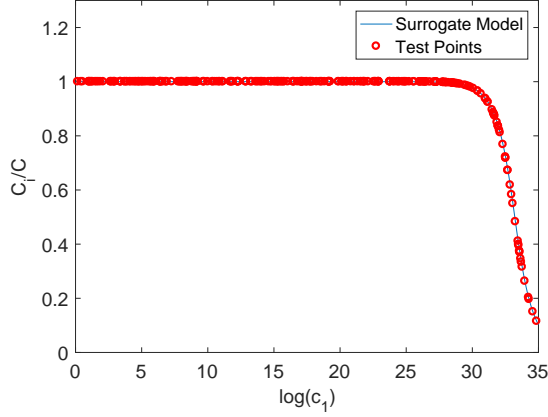


Figure 5.3: Cubic spline response surface model and test points for TCAT.

5.6 Evaluating the Surrogate Model Over Time

For our investigation in the previous section, we purposefully chose one of the more challenging time points in our model, given by breakthrough, or the brine front reaching the top of the column. For verification purposes, we fit the model at several other time points, which are quantified in Table 5.5. We note that all of the root mean square errors are small, indicating that the surrogate is a good approximation of the high-fidelity model.

We then wrote a script that creates an output of relative concentration for each of our $n = 525$ time points. The response surface model now outputs a breakthrough curve, with steepness controlled by $\log(c_1)$.

Breakthrough curves for selected values of $\log(c_1)$ are shown in Figure 5.4. We see that the curves sharpen as $\log(c_1)$ increases; however, there is not much change in output between $\log(c_1) \in [0, 28]$.

The norm of the root mean square error for the time-dependent surrogate model is 4.56×10^{-9} and the error bound is 1.14×10^{-7} . Recall that the model output $\frac{c_i}{c}$ is bounded between 0 and 1; therefore, a root mean square error on the order of 10^{-9} indicates that the errors in the surrogate model are small in comparison to the high-fidelity model output. The error bound is also small in comparison to the high-fidelity model, indicating that the surrogate model is an appropriate emulator for the high-fidelity model for all admissible

Table 5.5: Root mean square errors for different time .

Time (s)	Root Mean Square Error
7.52×10^3	7.60×10^{-19}
1.32×10^4	8.56×10^{-8}
1.47×10^4	1.53×10^{-5}
1.56×10^4	3.43×10^{-6}
1.62×10^4	1.58×10^{-6}
1.77×10^4	6.03×10^{-8}
2.07×10^4	6.11×10^{-8}

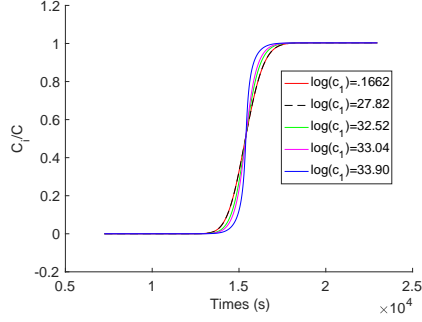


Figure 5.4: Surrogate breakthrough curves for varying $\log(c_1)$.

values of $\log(c_1)$ over time. Consequently, we have constructed a surrogate model that is an accurate emulator of the high-fidelity model.

5.7 Surrogate Models for Additional Experiments

In the previous two sections, we considered a surrogate model for the experiment in Column 11, B11, which has an incoming concentration of 0.01254. Next, we construct response surface models over time using a spline for three additional experiments. Experiments considered are listed in Table 5.6, and ω^{Aw}_{in} represents the incoming concentration for each experiment. We also tabulate the norm of the root mean square error and the error bound over time for each experiment. We note that while error

Table 5.6: Errors from surrogate modeling.

Experiment Number	$\omega^{A\bar{w}}_{in}$	Norm of Root Mean Square Error	Error Bound
B11	0.01254	4.56×10^{-9}	1.14×10^{-7}
B7	0.02541	7.73×10^{-9}	1.09×10^{-6}
B8	0.1002	7.90×10^{-9}	9.19×10^{-7}
B14	0.1508	1.46×10^{-8}	4.96×10^{-7}

increases as the incoming concentration increases, all of these values indicate a good fit to the high-fidelity model.

Next, we visualize the output of the surrogate model compared to test points at breakthrough for values of $\log(c_1)$. These investigations are again performed as breakthrough occurs to qualify the behavior of the model. Figure 5.5 shows that the surrogate provides an accurate and robust emulator of the full TCAT model at breakthrough.

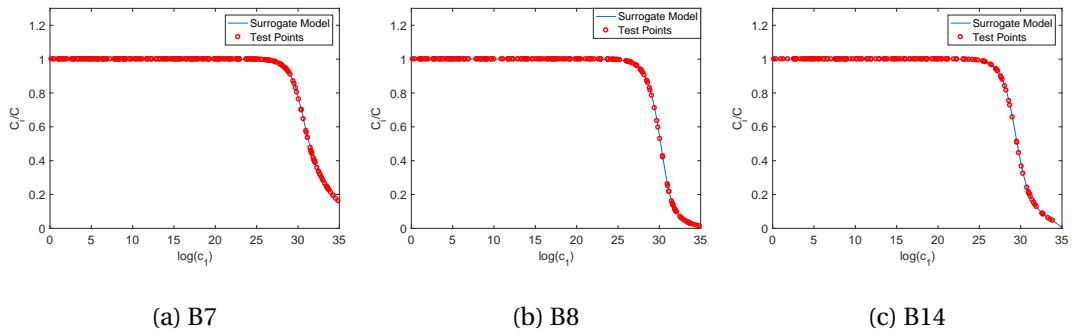


Figure 5.5: Surrogate models compared to test points at breakthrough.

5.8 Surrogate Model and Bayesian Inference

The surrogate models take approximately 0.01 seconds per function evaluation, compared to 15-20 minutes for the high-fidelity TCAT model. These computational savings allow us to run our Bayesian inference parameter estimation in less than 20 minutes, which is approximately the elapsed time for one high-fidelity TCAT simulation.

Now that we have a surrogate that emulates the response of the TCAT model, we can perform uncertainty quantification using Bayesian inference, as detailed in Chapter 6.

CHAPTER

6

BAYESIAN INFERENCE

In Chapter 4, we estimated parameters using experimental data and the Levenberg-Marquadt algorithm. Here, we perform statistical analysis of our model using Bayesian inference [71, 72].

Unlike frequentist inference, which is based on the tenet that parameters have unknown but fixed values, parameters are assumed to be random variables in the Bayesian framework. Hence, they have associated probability density functions (pdfs) that are quantified using Bayes' relation.

6.1 Statistical Modeling

In the previous chapter, we considered the statistical model

$$\Upsilon_i = f(t_i, q) + \epsilon_i, \quad i = 1, \dots, n. \quad (6.1)$$

Recall that Υ_i are random variables with realizations v_i . The model $f(t_i, q)$ yields a response based on input parameters q . We denote observative error by the random variables ϵ_i ,

which we assume are independent and identically distributed (iid) with mean 0 and variance σ^2 . We begin by plotting the kernel density estimates of the errors in Figure 6.1 to validate that $\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$. The distributions are approximately normal with means close to zero.

6.2 Frequentist and Bayesian Approaches

There are two contrasting approaches to inference that differ in how they consider probabilities. The first is the frequentist approach, where probability is defined as the frequency with which an event occurs if it is repeated many times. The goal of the frequentist approach is to determine parameters q such that they minimize the residual between the model response and data. In this framework, q is considered a fixed but unknown value. Once we have determined optimal q , we can construct confidence

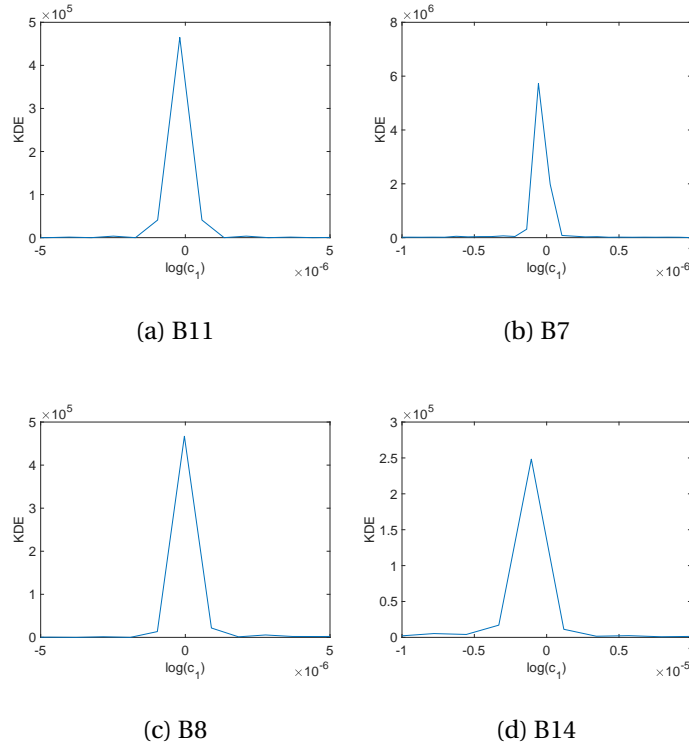


Figure 6.1: Error distributions for the four considered experiments.

intervals and sampling distributions. While this analysis can be useful for applications, the frequentist approach does not directly provide a distribution for parameters [71].

The other inference technique, which is utilized in this chapter, is the Bayesian approach. Unlike frequentist inference, which is based on the tenet that parameters have fixed but unknown values, Bayesian inference considers parameters as random variables. Thus, they have associated probability densities that are quantified using Bayes' relation [71]. The Bayesian approach can be advantageous for problems where there is uncertainty in the model [72].

6.3 Bayes' Relation

Parameter estimation within the Bayesian framework allows us to consider our parameters as random variables with distributions. Bayes' relation,

$$\pi(q|v) = \frac{\pi(v|q)\pi_0(q)}{\pi(v)}, \quad (6.2)$$

quantifies these distributions. Here $\pi(v|q)$ is the likelihood function, which incorporates information from the model and data. Specifically, it tells us the probability of obtaining observations v given input q [71].

The prior is denoted by $\pi_0(q)$, and it incorporates any prior knowledge about the parameters or data [72]. If the prior information is uncertain, we employ a flat prior [71].

When possible, it is preferred to use a conjugate prior so that one can obtain a closed-form solution of the posterior distribution. A conjugate prior has the same parametric form as the posterior distribution and allows us to determine a closed-form expression for the posterior. In this dissertation, we consider normally distributed observative errors, and therefore the likelihood function is in the inverse-gamma family. Thus, the conjugate prior is also in the inverse-gamma family [71]. More detailed information on priors can be found in [7, 36, 71].

The denominator in (6.2), $\pi(v)$, is the marginal density [71]. Mathematically, it is the integral over the parameter space of the numerator, and is given by

$$\pi(v) = \int_{\mathbb{R}^p} \pi(v|q)\pi_0(q) \, dq, \quad (6.3)$$

where p is the number of parameters [71].

The goal of using Bayesian inference is to determine the posterior pdf $\pi(q|\nu)$. Recall that we assume our statistical model has the form

$$\nu = f(q) + \epsilon. \quad (6.4)$$

In this dissertation, we use the assumption that the error is iid and normally distributed with mean 0 and variance σ^2 ,

$$\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2). \quad (6.5)$$

In this case, the likelihood function is

$$\pi(\nu|q) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2}, \quad (6.6)$$

where SS_q is the sum of squares error

$$SS_q = \sum_{i=1}^n (\nu_i - f_i(q))^2. \quad (6.7)$$

In practice, it is typically impossible to evaluate Bayes' relation directly. Markov Chain Monte Carlo (MCMC) methods solve this problem by allowing us to sample from the posterior [72].

We next discuss the Metropolis algorithm, which is a practical implementation of MCMC.

6.4 Markov Chain Monte Carlo (MCMC) and the Metropolis Algorithm

The marginal density in Bayes' relation, given by (6.3), can be a high-dimensional integral which is difficult or impossible to evaluate analytically. Markov Chain Monte Carlo (MCMC) methods allow us to address this issue by sampling from the posterior [72]. Using MCMC, we can explore the parameter space to determine the distribution of parameters. MCMC methods create a Markov Chain whose stationary distribution is the posterior distribution. Further, we can use realizations of our Markov chain to sample from the posterior and

determine marginal densities for the parameters [71]. Next, we will introduce the Metropolis algorithm, which allows us to implement MCMC.

The Metropolis algorithm allows us to generate a Markov Chain whose stationary distribution is that of the posterior [11, 26, 72]. The Metropolis algorithm uses an accept-reject criteria, where a new candidate point q^* is determined from a proposal distribution $J(q^k|q^{k-1})$ [72]. We use the proposal distribution

$$J(q^*|q^{k-1}) = N(q^{k-1}, V), \quad (6.8)$$

where V is the covariance matrix, to construct our candidate q^* . The explicit formula for the candidate q^* is then

$$q^* = q^{k-1} + Rz, \quad (6.9)$$

where R is the Cholesky factorization of V and $z \sim N(0, 1)$. Our proposal distribution $J(q^*|q^{k-1})$ is symmetric, which means $J(q^*|q^{k-1}) = J(q^{k-1}|q^*)$.

Next, we consider the probability of accepting the candidate q^* . We consider the ratio of posterior densities

$$r(q^*|q^{k-1}) = \frac{\pi(q^*|v)}{\pi(q^{k-1}|v)} = \frac{\pi(v|q^*)\pi_0(q^*)}{\pi(v|q^{k-1})\pi_0(q^{k-1})}. \quad (6.10)$$

We can eliminate the normalization constants since $\frac{\pi_0(q^*)}{\pi_0(q^{k-1})} = 1$ so that (6.10) becomes a ratio of likelihoods where

$$r(q^*|q^{k-1}) = \frac{\pi(v|q^*)}{\pi(v|q^{k-1})}. \quad (6.11)$$

We assume that the errors are iid and normally distributed with mean 0 and variance σ^2 so that (6.11) becomes

$$r(q^*|q^{k-1}) = e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}, \quad (6.12)$$

where s_{k-1}^2 is an estimate for σ^2 .

We accept the candidate with probability

$$\alpha(q^*|q^{k-1}) = \min(1, r(q^*|q^{k-1})). \quad (6.13)$$

If we accept the candidate q^* , we set $q^k = q^*$ and also update the sum of squares error such that $SS_{q^k} = SS_{q^*}$. If we reject the candidate q^* , we revert to the previous value of q such

that $q^k = q^{k-1}$ and $SS_{q^k} = SS_{q^{k-1}}$. We update s_k by sampling $s_k \sim \text{Inv-gamma}(a_{val}, b_{val})$. For details on how to derive these shape parameters, see [71]. The result of the Metropolis algorithm is a chain whose stationary distribution is the posterior. The Metropolis algorithm, taken directly from [71], is summarized in Algorithm 2.

The Metropolis Hastings (MH) algorithm is a variation of the Metropolis algorithm where we allow the proposal distribution $J(q^*|q^{k-1})$ to be asymmetric [71, 72]. We do not further consider the MH algorithm in this dissertation.

Once we have constructed a chain $[q^1, \dots, q^N]$, we must show that q converges to the posterior distribution $\pi(q|\nu)$. In the next section, we show how we determine that the chain has converged to the posterior distribution.

Algorithm 2 Metropolis Algorithm

1. Set input parameters.
2. Determine q^0 by performing a least squares optimization.
3. Determine the initial error, $SS_{q^0} = \sum_{i=1}^n (v_i - f_i(q^0))^2$.
4. Determine the initial variance estimate, $s_0^2 = \frac{SS_{q^0}}{n-p}$.
5. Construct and compute the Cholesky factorization R of a covariance estimate V .
6. Loop over M , the number of chain iterates.

(Continued on the next page)

Algorithm 2 Metropolis Algorithm (continued)

7. For $i = 2, \dots, M$

(a) Sample $z_k \sim N(0, 1)$.

(b) Choose a new candidate $q^* = q^{k-1} + Rz_k$.

(c) Sample $u_\alpha \sim U(0, 1)$.

(d) Compute $SS_{q^*} = \sum_{i=1}^n [v_i - f_i(q^*)]^2$.

(e) Determine $r(q^*|q^{k-1}) = e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}$.

(f) Determine $\alpha(q^*|q^{k-1}) = \min(1, r(q^*|q^{k-1}))$.

(g) If $u_\alpha < \alpha(q^*|q^{k-1})$,

$$q^k = q^* \text{ and } SS_{q^k} = SS_{q^*}.$$

else

$$q^k = q^{k-1} \text{ and } SS_{q^k} = SS_{q^{k-1}}.$$

endif

(h) Update $s_k \sim \text{Inv-gamma}(a_{val}, b_{val})$, where $a_{val} = \frac{1}{2}(n_s + n)$, $b_{val} = \frac{1}{2}(n_s s_{k-1}^2 + SS_{q^k})$.

6.5 Convergence

With enough iterations, chains will converge to the posterior [71]. For the first several MCMC iterations, runs typically experience a burn-in period. This is a certain number of chain iterations before the algorithm converges to the posterior distribution. Depending on the problem and q^0 , the burn-in period can vary significantly. To qualitatively establish convergence, one starts with a large number of chain iterations and plots the chain for each parameter. We can visually observe when the chain has converged to the correct distribution [72], although we will also use statistical tests to prove the chains have converged. One metric τ is the integrated autocorrelation time, which is the estimated number of steps needed to draw independent samples from the posterior distribution. Smaller values of

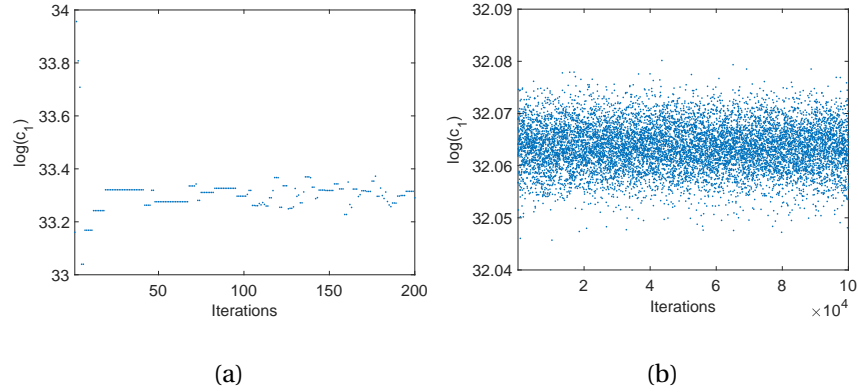


Figure 6.2: (a) Burn-in period for the chain and (b) chain that has converged.

τ indicate that the chain converges more quickly [14]. A second convergence metric is a parameter geweke, which splits the chain into pieces and tests the resulting pieces for similarity. Typically, the chain is split into the first half and final ten percent, and the means are compared. A value of geweke close to 1 indicates that the first and last elements are from the same distribution. Whereas it is tempting to interpret this as indicating evidence, the chain may be stuck in a local minima. Hence, while these statistical tests can establish non-convergence, they may not establish convergence. A more rigorous test of convergence is to run multiple chains and compute the Rubin-Gelman statistic [72].

The Rubin-Gelman statistic R is computed from $m \geq 2$ independent chains with length $2n$ and different initial starting values across the admissible parameter space. We use the last n entries of each chain and compute a target variance R , which quantifies a comparison of the mean of each chain to the mean of the m independent chains and also measures the scaled total variance of the chains. We expect $R \rightarrow 1$, which proves that the chains converge across the admissible parameter space and the chain is not stuck in a local minima [21].

Figure 6.2 shows two Markov chains. Figure 6.1(a) shows the first 200 iterations of the Markov chain while burn in is occurring. Figure 6.1(b) shows a Markov chain that has burned in and has converged. While these plots do not show that the chain has converged over the admissible parameter space, we can infer that the chain has converged to a local minima.

In the next section, we will discuss Delayed Rejection Adaptive Metropolis (DRAM), which is an improvement on Algorithm 2. We used DRAM to determine our posterior distributions of parameters that we present in Section 6.7.

6.6 Implementation: DRAM

The implementation of MCMC that we use is Delayed Rejection Adaptive Metropolis (DRAM) [29]. Delayed rejection (DR) is an altered form of the Metropolis algorithm [29, 75]. The goal of DR is to improve the efficiency of the estimators. DR accomplishes this goal by altering how the Metropolis algorithm addresses rejected candidates q^* [29]. In the Metropolis algorithm, when we reject q^* , we revert to q^{k-1} in the chain. However, DR provides an alternative candidate q^{*2} if q^* is rejected. Our second-stage candidate is proposed from the distribution

$$q^{*2} = q^{k-1} + \gamma_2 R_k z_k. \quad (6.14)$$

Here, $z_k \sim N(0, 1)$, R_k is the Cholesky factorization of the covariance matrix V , and $\gamma_2 = \frac{1}{5}$ is a design parameter that narrows the second-stage proposal function and increases mixing in the now smaller region. The code that we use employs a second-stage candidate [71]. We explicitly outline the construction of q^{*2} in Algorithm 4.

Adaptive Metropolis (AM) allows us to better choose the proposal distribution while utilizing past samples of the chain [29]. If we consider step 5 in Algorithm 2, we see that our proposal distribution J depends on the initial covariance estimate V . However, AM takes this one step further by instead allowing J to be updated using the previous chain samples. To determine our distribution, we take the covariance matrix at the k^{th} step, which is given by [71]

$$V_k = s_p \text{cov}(q^0, \dots, q^{k-1}) + \varepsilon I_p. \quad (6.15)$$

Here, s_p is a scaling parameter that is set to $\frac{2.4}{p^2}$, where p is the number of parameters in the model [29], ε is a parameter that guarantees that V_k is positive definite, and I_p is the $p \times p$ identity matrix. In most practical applications, we set $\varepsilon = 0$ [29, 71]. In the code that we use, the covariance matrix is updated every 100 iterations. We perform a Cholesky decomposition where $V_k = R_k R_k^T$. Next, we take the candidate from the proposal

distribution to be

$$q^* = q^{k-1} + R_k z_k, \quad (6.16)$$

where $z_k \sim N(0, 1)$. This is step 6(i) in Algorithm 3.

The advantage of AM is that by allowing the proposal distributions to be updated as we update the chain, we can incorporate past information and arrive at a more plausible candidate from the proposal distribution [71]. However, because we are considering past time states, the process is no longer Markovian. Haario et al [29] showed that the chain generated by DRAM converges uniquely to the posterior distribution.

We use the implementation of DRAM in MATLAB written by Marko Laine. This open-source software is publicly available; for more details see [42]. The algorithm used in [42] for a flat prior is provided in [71], and is given in Algorithm 3.

Algorithm 3 DRAM Algorithm

1. Set input parameters.
2. Determine q^0 by performing a least squares optimization.
3. Determine the initial error, $SS_{q^0} = \sum_{i=1}^n (v_i - f_i(q^0))^2$.
4. Determine the initial variance estimate, $s_0^2 = \frac{SS_{q^0}}{n-p}$.
5. Construct and compute the Cholesky factorization R of a covariance estimate V .

(Continued on the next page)

Algorithm 3 DRAM Algorithm (continued)

6. For $k = 1, \dots, M$

(a) Sample $z_k \sim N(0, 1)$.

(b) Construct a candidate $q^* = q^{k-1} + Rz_k$.

(c) Compute $SS_{q^*} = \sum_{i=1}^n [v_i - f_i(q^*)]^2$.

(d) Sample $u_\alpha \sim U(0, 1)$.

(e) Determine $r(q^*|q^{k-1}) = e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}$.

(f) Determine $\alpha(q^*|q^{k-1}) = \min(1, r(q^*|q^{k-1}))$.

(g) If $u_\alpha < \alpha$,

$$q^k = q^* \text{ and } SS_{q^k} = SS_{q^*}.$$

else

enter DR Algorithm 3.

endif

(h) Update $s_k \sim \text{Inv-gamma}(a_{val}, b_{val})$, where $a_{val} = \frac{1}{2}(n_s + n)$, $b_{val} = \frac{1}{2}(n_s s_{k-1}^2 + SS_{q^k})$.

(i) If $\text{mod}(k, 100) = 1$,

$$\text{Update } V_k = s_p \text{cov}(q^0, \dots, q^k), V_k = V_{k-1}.$$

else

$$\text{Update } R = \text{chol}(V_k).$$

endif

Algorithm 4 DR Algorithm

1. Set design parameter $\gamma_2 = \frac{1}{5}$.
 2. Sample $z_k \sim N(0, 1)$.
 3. Construct a second-stage candidate $q^{*2} = q^{k-1} + \gamma_2 R_k z_k$.
 4. Sample $u_\alpha \sim U(0, 1)$.
 5. Compute $SS_{q^{*2}} = \sum_{i=1}^n (v_i - f_i(q^{*2}))^2$.
 6. Compute $\alpha_2(q^{*2}|q^{k-1}, q^*) = \min\left(1, \frac{\pi(q^{*2}|v)J(q^*|q^{*2})(1-\alpha(q^*|q^{*2}))}{\pi(q^{k-1}|v)J(q^*|q^{k-1})(1-\alpha(q^*|q^{k-1}))}\right)$.
 7. If $u_\alpha < \alpha_2$,
 - $q^k = q^{*2}$ and $SS_{q^k} = SS_{q^{*2}}$.
 - else
 - $q^k = q^{k-1}$ and $SS_{q^k} = SS_{q^{k-1}}$.
 - endif
-

For details on how we compute $\alpha_2(q^{*2}|q^{k-1}, q^*)$ for the second-stage candidate, see [71]. We use the DRAM algorithm to generate a chain whose stationary distribution is the posterior [29]. In the next section, we will discuss the application of DRAM to the TCAT model.

6.7 DRAM Applied to the TCAT Model

Recall that in the previous chapter, we constructed time-dependent surrogate models for four selected experiments. Here, we use DRAM to construct the posterior distribution for these experiments: B11, B7, B8, and B14. Table 6.1 lists the experiments simulated along with the associated incoming concentration $\omega^{A\overline{w}}_{in}$.

Table 6.1: Experiments considered for DRAM simulations.

Experiment Number	$\omega^{A\bar{w}}_{in}$
B11	0.01254
B7	0.02541
B8	0.1002
B14	0.1508

These simulations were performed with a burn-in period of 10^4 iterations. The final 10^5 function evaluations are shown in Figure 6.3. Both visual inspection and statistical tests show that the chains have converged. Further, we see that the mean of $\log(c_1)$ decreases with the increase in concentration.

We plot kernel density estimates (KDEs) for the parameter distributions for the four experiments in Figure 6.4. We see that σ^2 decreases across the first three experiments, then increases for experiment B14. This phenomena is due to the complex closure relation for diffusion employed by TCAT that causes the DAEs to become more stiff as incoming concentration increases. We performed worst-case estimates of the spatially varying diffusion calculation and found that the value of D^{ABwe} is markedly smaller for B14 than for the other three experiments. The value of D^{ABwe} is on the order of 10^{-20} for B14 at the location of the front. This means that when simulating experiment B14, the ode15s algorithm may fail and produce results with prohibitively large errors. This consequently leads the surrogate to emulate a breakthrough curve that may be inaccurate. Therefore, this anomaly in the variance is most likely due to the failure of the ode15s solver for higher concentration problems.

In Table 6.2, we summarize the summary statistics from DRAM. Note again that the mean value of $\log(c_1)$ decreases with an increase in incoming concentration. The standard deviation also decreases as the incoming concentration becomes larger, with the exception of B14, as discussed earlier. These results verify what we noted visually with the chain and KDE plots.

Further, in Table 6.2, we note that the MCMC error is relatively small for all four cases, indicating that the batch standard deviations are close to one another. The final two columns in Table 6.2 are the convergence metrics τ and geweke that we presented in Section 6.5. We see that all values of τ remain small, indicating that our DRAM simulation is efficient and

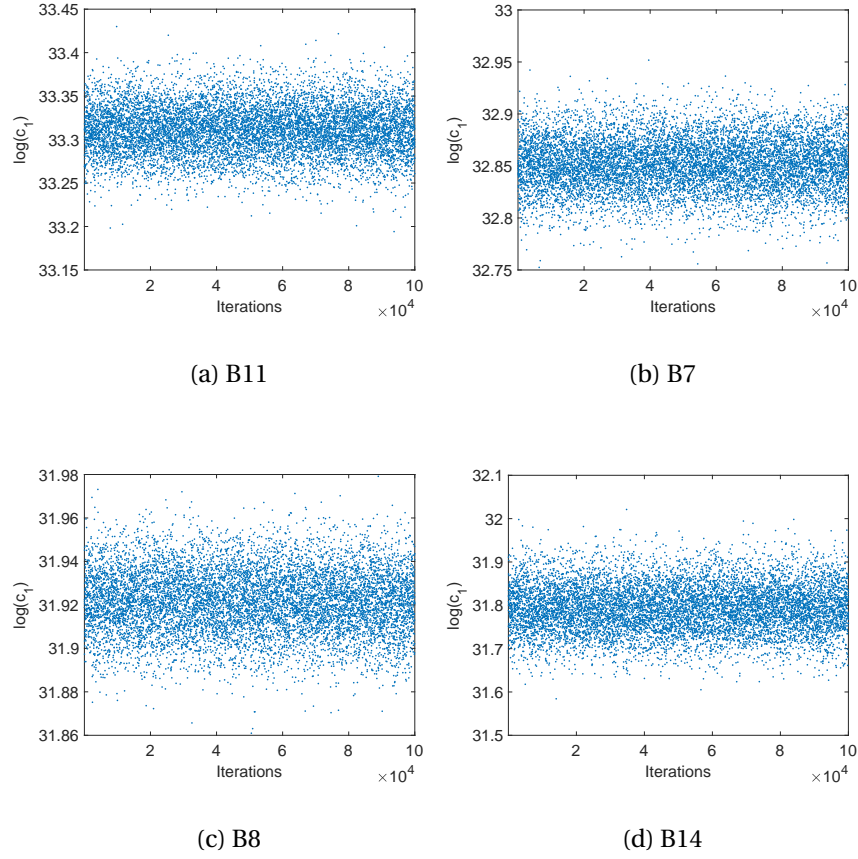


Figure 6.3: MCMC chains for the four considered experiments.

converges quickly. In all four cases, we see that our value of geweke is close to 1. Further, the Rubin-Gelman statistic R is close or equal to 1 for all cases, indicating that the chains have converged across the admissible parameter space.

In the next chapter, we will use both frequentist and Bayesian techniques to investigate mixed-effects models for a simplified version of the TCAT model.

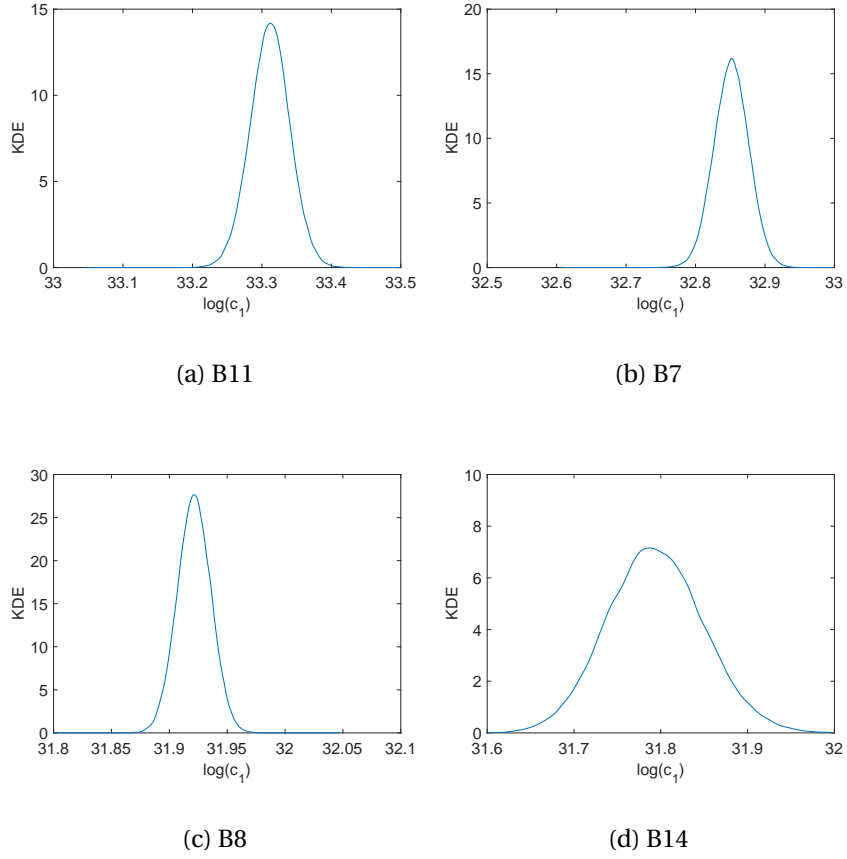


Figure 6.4: Kernel density estimates for the four considered experiments.

Table 6.2: Summary statistics from DRAM for the four considered experiments.

Experiment Number	ω_{in}	Mean ($\log(c_1)$)	σ^2	MC Error	τ	geweke	R
B11	0.01254	33.312	2.8384×10^{-2}	2.5086×10^{-4}	3.93	0.999	1.000
B7	0.02541	32.852	2.5061×10^{-2}	1.1397×10^{-4}	3.86	0.999	0.999
B8	0.1002	31.921	1.4450×10^{-2}	7.8285×10^{-5}	3.80	0.999	0.999
B14	0.1508	31.794	5.5299×10^{-2}	2.4870×10^{-4}	3.90	0.999	0.999

CHAPTER

7

MIXED-EFFECTS MODELING

In this chapter, we will discuss mixed-effects modeling and show how it can be applied to the TCAT model.

7.1 Motivation

Consider a problem from [20] where the heights of 26 boys are measured on nine occasions over time. A plot of the measurement time versus data is pictured in Figure 7.1.

Visual inspection of Figure 7.1 yields two basic ideas; first, it is clear that the initial height of the boys varies. Second, the data corresponding to growth over time appear to be linear with little variation in the slope across different subjects.

Our goal is to formulate a model that quantifies how individual effects arise across repeated measurements over time. These effects can be from individual variation across a population or from variations in experimental conditions [66].

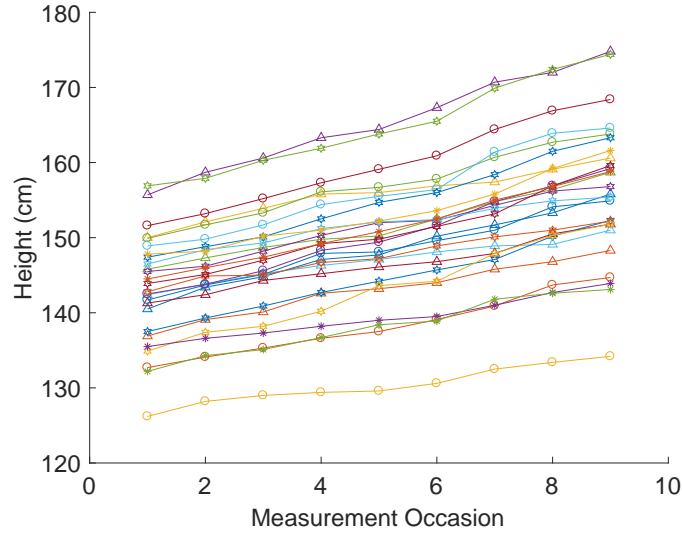


Figure 7.1: Measurement heights (cm) for 26 boys on 9 occasions [20].

Mixed-effects models present a way to quantify variations from individual responses within a population. Instead of considering a parameter θ as a single quantity, we consider

$$\theta_i = \beta + r_i. \quad (7.1)$$

Here β are the fixed effects from the model, which are parameters that are consistent across a population or set of experimental data. The random effects r_i are sample-dependent and will be determined for each experiment or observation [55]. Note that we have changed notation for a parameter from q in the previous chapters to θ . This change of notation indicates the parameter contains both a fixed and a random effect.

The statistical mixed-effects model is given by

$$y_{ij} = f(x_{ij}; \beta, r_i) + \epsilon_{ij}. \quad (7.2)$$

In this model, i denotes each individual and j denotes the number of measurements. The model f is evaluated dependent on the independent variables x , the fixed effects β , and the random effects r_i . Random variables ϵ_{ij} represent the measurement error. Here y_{ij} is

the j th observation for individual i [66]. We assume that the random effects and error are independent and normally distributed such that

$$r_i \sim N(0, \psi) \quad (7.3)$$

$$\epsilon_{ij} \sim N(0, \sigma^2), \quad (7.4)$$

where ψ is the covariance of random effects and σ^2 is the variance of the measurement errors. In this model, the unknowns are the fixed effects β , the random effects r_i , the variance of the error σ^2 , and the covariance ψ [66].

7.2 Linear Mixed-Effects Models

A linear mixed-effects model has the standard form

$$y = X\beta + Zb + \epsilon. \quad (7.5)$$

Here $X\beta$ represent the fixed effects, Zb represent the random effects, and ϵ represents the measurement error. The $n \times 1$ response vector is given by y , where n is the number of observations, and X is the $n \times p$ fixed effects design matrix. Here p is the number of fixed effects, Z is the $n \times q$ random effects design matrix, where q is the number of random effects, b is the $q \times 1$ random effects vector, and ϵ is the $n \times 1$ vector of observation errors [57].

We assume that b and ϵ have normal prior distributions given by

$$b \sim N(0, \sigma^2 D(\theta)) \quad (7.6)$$

$$\epsilon_{ij} \sim N(0, \sigma^2 I). \quad (7.7)$$

Here I is the $n \times n$ identity matrix and D is a symmetric positive semidefinite matrix parameterized by a variance component vector θ [57].

A classic example of the linear mixed-effects model was introduced in the previous section [20, 82] where researchers measured the heights of 26 boys on 9 occasions as illustrated in Figure 7.1. The model is given by

$$y_{ij} = \beta_0 + r_{0i} + \beta_1 x_{ij} + \epsilon_{ij}. \quad (7.8)$$

Here y_{ij} represents the measured heights of the boys, β_0 represents the average starting height for the population, r_{0i} represents the variation in starting height, β_1 represents the change in height over time, x_{ij} represents the j th vector of independent variables, in this case, time, for the i th subject, and ϵ_{ij} is the measurement error. Here, it is assumed that

$$r_i \sim N(0, \psi) \quad (7.9)$$

$$\epsilon_{ij} \sim N(0, \sigma^2). \quad (7.10)$$

Note that in this model, only β_0 has an associated random effect, whereas β_1 does not. This is because treating β_1 as a fixed effect leads to an issue of parameter non-identifiability as the slope of the boys' growth is fairly constant across different subjects [66].

From (7.5), the growth model can be formulated as

$$y_{ij} = \begin{bmatrix} 1 & x_{ij} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \epsilon_{ij}. \quad (7.11)$$

Recall that we have four unknowns in the mixed-effects model: β , b , σ^2 , and θ . We will use maximum likelihood estimation to determine our unknown parameters as detailed in [54, 57]. The likelihood function to be optimized is

$$L(\beta, \sigma^2, \theta, |y) = \prod_{i=1}^M p(y_i | \beta, \sigma^2, \theta). \quad (7.12)$$

Because the random effects for each group are independent, (7.12) can be formulated as

$$L(\beta, \sigma^2, \theta, |y) = \prod_{i=1}^M \int p(y_i | \beta, b_i, \sigma^2, \theta) p(b_i | \sigma^2, \theta) db_i. \quad (7.13)$$

The pdf $p(y | \beta, b, \sigma^2, \theta)$ is a multivariate normal distribution centered about $X\beta + Zb$ with variance σ^2 such that

$$y | \beta, b, \sigma^2, \theta \sim N(X\beta + Zb, \sigma^2 I). \quad (7.14)$$

Therefore, it follows that

$$\begin{aligned}
p(y|\beta, b, \sigma^2, \theta) &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(\frac{-\sum_{i=1}^M |y_i - X_i\beta - Z_i b|}{2\sigma^2}\right) \\
&= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(\frac{|y - X\beta - Zb|}{2\sigma^2}\right).
\end{aligned} \tag{7.15}$$

Because $b_\ell \sim N(0, \psi)$ we can simplify

$$\begin{aligned}
p(b|\theta, \sigma^2) &= \frac{1}{(2\pi)^{q/2} |\psi|^{1/2}} \exp\left(\frac{-1}{2} b^T \psi^{-1} b\right) \\
&= \frac{1}{(2\pi)^{q/2} |D(\theta)|^{1/2}} \exp\left(\frac{-1}{2} b^T D(\theta)^{-1} b\right).
\end{aligned} \tag{7.16}$$

Therefore, our likelihood becomes

$$L(\beta, \sigma^2, \theta|y) = \frac{\text{abs}|\Delta(\theta)|}{(2\pi\sigma^2)^{n/2}} \int \frac{\exp\left[-(|y - X\beta - Zb| + |\Delta(\theta)b|)/2\sigma^2\right]}{(2\pi\sigma^2)^{q/2}} db. \tag{7.17}$$

Here $\Delta(\theta)$ is a non-unique matrix matrix that satisfies the condition $D(\theta)^{-1} = \Delta(\theta)^T \Delta(\theta)$ [66].

Note that b still occurs in the right-hand side of (7.17). To transform (7.17) into the form that is amenable to MATLAB's optimization codes, we will first find the conditional modes, which are the differences between the average response and the response for each individual [15]. From there, one would construct a profiled likelihood, which is parameterized by θ only. This yields the likelihood that is used in practice for MATLAB's codes for maximum likelihood estimation of mixed-effects models [54, 66].

MATLAB has two options for maximum likelihood estimations of linear mixed-effects models: `fitlme` and `fitlmematrix`. The first, `fitlme`, accepts a linear mixed-effects model, specified by a user-provided formula, to the user-provided data table [57]. The other option, `fitlmematrix`, accepts design matrices X and Z , user-provided data, and an optional grouping variable [53]. Both options use a quasi-Newton optimization to determine optimal values of fixed and random effects [66].

We now apply `fitlmematrix` to the problem outlined in (7.8). The fixed effects are

$$\begin{aligned}
\beta_0 &= 64.56 \\
\beta_1 &= 6.524.
\end{aligned}$$

There are 26 random effects associated with the intercept β_0 . A scatter plot of the random effects is provided in Figure 7.2. The covariance of the random effects is given by $\psi = 7.9391$ and the variance is given by $\sigma^2 = 2.9241$. We selected 5 of the 26 boys to serve as a representative sample for comparing the model to measured data. The model fit and residuals are pictured in Figure 7.3.

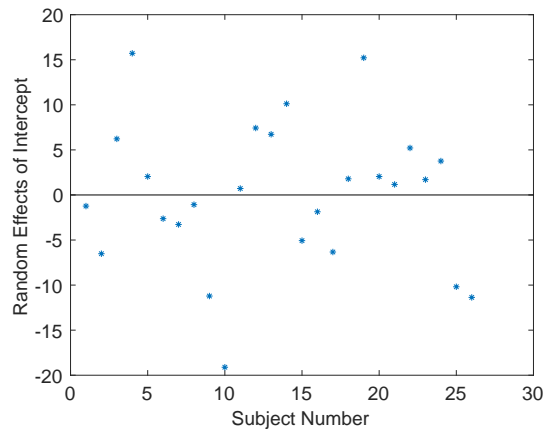


Figure 7.2: Random effects plotted by subject.

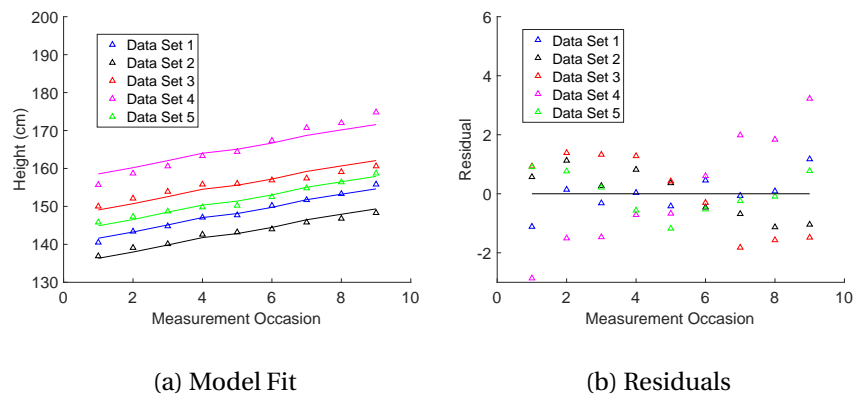


Figure 7.3: Model fit and residuals for the Oxford boys growth model.

7.3 Mixed-Effects Applied to TCAT

Here, we construct a mixed-effects model for TCAT. Unfortunately, we cannot use the full TCAT model to construct our mixed-effects model due to the dependence of the diffusion term on the incoming concentration. The nonlinear mixed-effects software that we will use, MATLAB's `nlmefit`, does not support observation-dependent design [55]. We will instead use a simplified version of the TCAT model to perform our mixed-effects simulations. This simplified version of the TCAT model is based on classical work by Javandel, who derived a set of analytic solutions for simple advection-diffusion problems. In this set of solutions, the unknown parameter D determines how much diffusion is simulated. The solution is nonlinear in D and independent of incoming concentration [35]. The simplified version of the TCAT model is provided only to illustrate the process of constructing a mixed-effects model for an advection-diffusion problem.

Recall from Section 2.7 that the analytic solution to our simplified TCAT model is [35]

$$\frac{C}{C_0}(L, t) = \frac{1}{2} \operatorname{erfc}\left(\frac{L - \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right) + \frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} z}{D}\right) \operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right), \quad (7.18)$$

and for D sufficiently small, we have the analytic solution

$$\begin{aligned} \lim_{D \rightarrow 0} \frac{C}{C_0}(L, t) &= \lim_{D \rightarrow 0} \left(\frac{1}{2} \operatorname{erfc}\left(\frac{L - \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right) + \frac{1}{2} \exp\left(\frac{\epsilon^{\bar{w}} v^{\bar{w}} L}{D}\right) \operatorname{erfc}\left(\frac{L + \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right) \right) \\ &= \frac{1}{2} \operatorname{erfc}\left(\frac{L - \epsilon^{\bar{w}} v^{\bar{w}} t}{2(Dt)^{1/2}}\right). \end{aligned} \quad (7.19)$$

As we detailed in Chapters 4 and 6, we can view parameters from the frequentist or Bayesian perspectives. We will estimate the unknown parameter D from a frequentist perspective as we did in Chapter 4. We will also use Bayesian inference to estimate D and we will again use DRAM to simulate posteriors for each of our random effects. From there, we will compare the output of the frequentist and Bayesian approaches.

The linearized TCAT model, which yields a relative concentration $\frac{C}{C_0}$, provides a breakthrough curve that is independent of incoming concentration. We use breakthrough curves generated by the model as a comparison to four different data sets generated in the lab. Details on the selected experiments are listed in Table 7.1, where

$\omega^{A\bar{w}}_{in}$ is the incoming concentration. The linearized TCAT model has only one unknown parameter D and the independent variable is time. Therefore, the mixed-effects model in (7.2) can be formatted as

$$y_j = f(t_j; \beta, r_i) + \epsilon_j, \quad (7.20)$$

with

$$r_i \sim N(0, \psi) \quad (7.21)$$

$$\epsilon_j \sim N(0, \sigma^2). \quad (7.22)$$

7.4 Frequentist Mixed-Effects Parameter Estimation

Although we are able to linearize the differential equations presented in the TCAT model, the analytic solution to the simplified TCAT model is nonlinear in the parameter of interest D . Recall that in the previous section, we derived a likelihood for the linear mixed-effects model and discussed how it could be parameterized exclusively by θ . For models that are nonlinear, this closed-form solution is often not possible because of the integral in (7.13).

Consequently, we use MATLAB's `nlmefit` to determine both the fixed and random effects in the TCAT model. The `nlmefit` code takes a set of predictors, a set of model responses, grouping variables 1-4, allowing the model to differentiate between the four experiments, the model to be evaluated, and an initial estimate β_0 as input [55]. Then, `nlmefit` approximates the likelihood function using the likelihood for the linear mixed-effects model for current values of β and r and uses `fminsearch` to optimize this estimated likelihood.

Table 7.1: Experiments considered for mixed-effects modeling.

Experiment Number	$\omega^{A\bar{w}}_{in}$
B11	0.01254
B7	0.02541
B8	0.1003
B14	0.1508

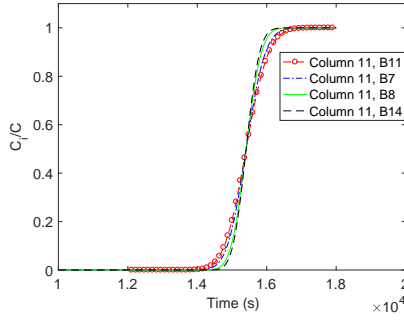


Figure 7.4: Mixed-effects model breakthrough curves.

Results from the mixed-effects model are listed in Table 7.2. The mean or fixed effect is given by $\beta = D = 1.7531 \times 10^{-4}$. The variance is given by $\sigma^2 = 5.7640 \times 10^{-4}$ and the covariance between the random effects is given by $\psi = 7.7211 \times 10^{-5}$. For each experiment listed in Table 7.2, r_i is the random effect and $D_i = D + r_i$ is the mean value of each parameter.

Next, we investigate the breakthrough curves generated by the mixed effects model, considering both fixed and random effects. Figure 7.4 shows that the curves get steeper as the data generated provides a sharper breakthrough curve, which is what we expect.

We next present model fits to data from `nlmefit`. Fits for the four experiments considered are shown in Figure 7.5. Residual norms from the mixed-effects TCAT model are listed in Table 7.3. These norms quantitatively indicate a close fit between the model and experimental data. These results show that the mixed-effects TCAT model is able to reproduce the behavior of the data well.

Table 7.2: Random effects from `nlmefit`.

Experiment Number	$\omega^{A\bar{w}}_{in}$	D_i	r_i
B11	0.0125	2.7199×10^{-4}	9.6680×10^{-5}
B7	0.02541	2.0195×10^{-4}	2.6648×10^{-5}
B8	0.1003	1.2710×10^{-4}	-4.8210×10^{-5}
B14	0.1508	1.0019×10^{-4}	-7.5117×10^{-5}

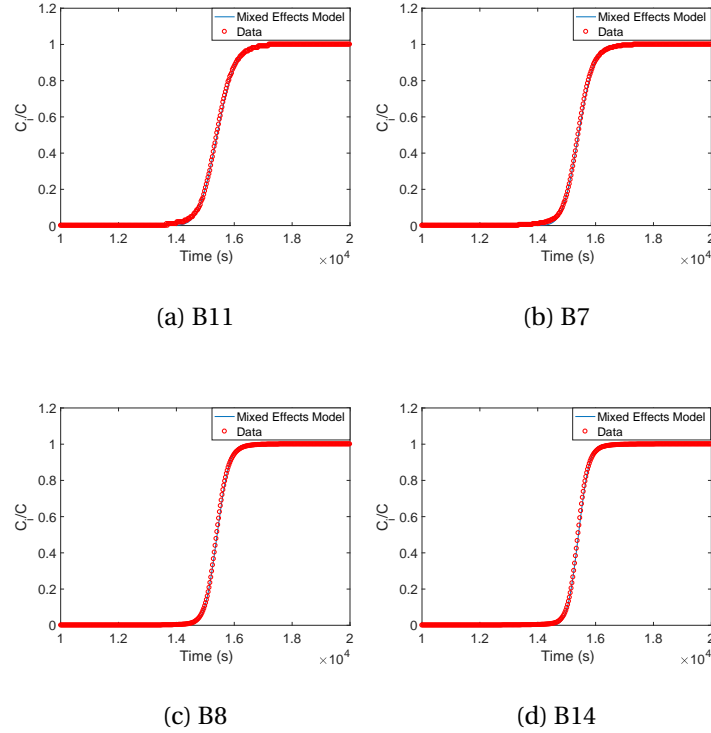


Figure 7.5: Mixed-effects responses compared to data.

7.5 Bayesian Parameter Estimation: DRAM

Here, we use Bayesian inference to gain additional insight into our mixed-effects TCAT model. As in the previous chapter, we will use Marko Laine's DRAM code to simulate the posterior [42]. Laine's code can be modified to fit a mixed-effects model by introducing batches of data, where one batch represents one experiment, and treating the set of parameters $[\beta_1, \dots, \beta_4]$ as a set of hyperparameters [66].

Recall from the previous chapter that the likelihood used by DRAM for n observations y , where the errors are iid, unbiased, and normally distributed with unknown variance σ^2 , is given by [71]

$$L(\beta, \sigma^2 | y) = \frac{1}{(\pi \sigma^2)^{n/2}} \exp \left(-\sum_{j=1}^n [y_j - f(t_j, \beta)]^2 / 2\sigma^2 \right). \quad (7.23)$$

Table 7.3: The residual norms from `nlmefit`.

Experiment Number	$\omega^{A\bar{w}}_{in}$	Residual Norm
B11	0.01254	1.5072×10^{-3}
B7	0.02541	2.9405×10^{-3}
B8	0.1003	9.4464×10^{-3}
B14	0.1508	1.4805×10^{-2}

For the TCAT mixed-effects model with 4 experiments, (7.23) becomes [66]

$$L([\beta_1, \dots, \beta_4], \sigma^2 | y) = \frac{1}{(\pi \sigma^2)^{n/2}} \sum_{i=1}^4 \exp \left(\sum_{j=1}^n [y_j - f(t_j, \beta_i)]^2 / 2\sigma^2 \right). \quad (7.24)$$

The DRAM code calculates both the posterior and variance for the average response, as well as the posterior and variance for each hyperparameter.

Initial estimates for hyperparameters were set equal to their optimal frequentist counterparts. Hyperparameters were bounded between $D \in [0, 1]$ with an initial variance of $\sigma^2 = 10^{-8}$. The mean is given by $D = 1.7539 \times 10^{-4}$, with a standard deviation of $\sigma^2 = 6.4746 \times 10^{-5}$ and a covariance of $\psi = 1.0711 \times 10^{-4}$. Results from the DRAM mixed-effects simulation are presented in Table 7.4.

Table 7.5 shows the summary statistics from DRAM. We see that the values of τ remain relatively small. Further, the geweke parameter is close to 1. These two metrics are consistent with the observations that the chains have converged. We also computed Rubin-Gelman statistics, as in Chapter 6, to verify that we are not stagnating at a local minimum. Results

Table 7.4: The random effects from DRAM.

Experiment Number	$\omega^{A\bar{w}}_{in}$	D_i	r_i
B11	0.01254	2.7199×10^{-4}	9.6600×10^{-5}
B7	0.02541	2.0199×10^{-4}	2.6600×10^{-5}
B8	0.1002	1.2708×10^{-4}	-4.8310×10^{-5}
B14	0.1508	1.0021×10^{-4}	-7.5180×10^{-5}

Table 7.5: The summary statistics from DRAM applied to the hyperparameters.

Experiment Number	$\omega^{A\bar{w}}_{in}$	D_i	σ^2	MC Error	τ	geweke	R
B11	0.01254	2.7199×10^{-4}	1.7827×10^{-6}	2.4895×10^{-8}	12.5	0.999	1.00
B7	0.02541	2.0199×10^{-4}	1.4355×10^{-6}	2.0894×10^{-8}	12.7	0.999	0.999
B8	0.1002	1.2708×10^{-4}	9.9651×10^{-7}	1.4480×10^{-8}	13.7	0.999	0.999
B14	0.1508	1.0021×10^{-4}	8.4078×10^{-7}	1.4261×10^{-8}	12.5	0.999	0.999

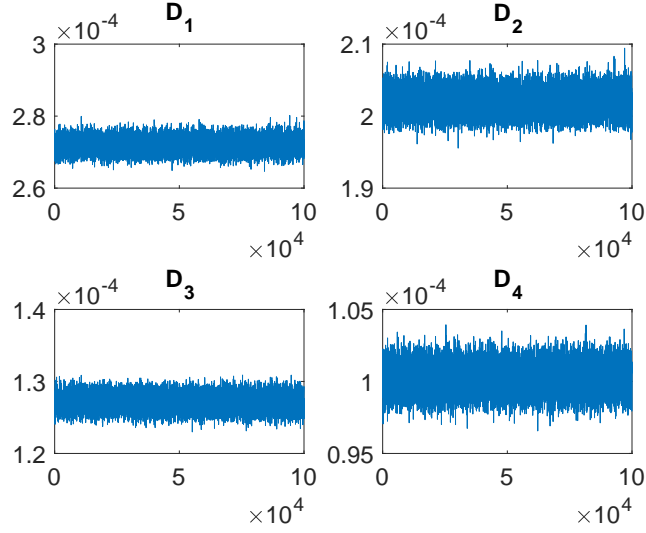


Figure 7.6: DRAM results for hyperparameters.

from all four hyperchains are close to or exactly equal to 1, thus showing that the DRAM simulation has converged across the range of admissible parameter values. As we saw in Chapter 6, the mean of the parameter decreases as the incoming concentration increases. Here, the variance and MC error decrease as incoming concentration increase because our closure relation for diffusion is not a function of the density gradient.

After a burn-in period of 10^4 iterations, the final parameter chains of length 10^5 are presented in Figure 7.6. In addition to the τ and geweke tests, we can visually observe that the chains for all four parameters have converged.

Finally, the hyperchains for D and ψ are presented in Figure 7.7. We again see that these chains have converged and the results of the final 10^5 iterations are shown. Next, we look at model fits to data from DRAM. Fits for the four experiments considered qualitatively agree with the data and are shown in Figure 7.8. Residual norms from the mixed-effects TCAT model are listed in Table 7.6.

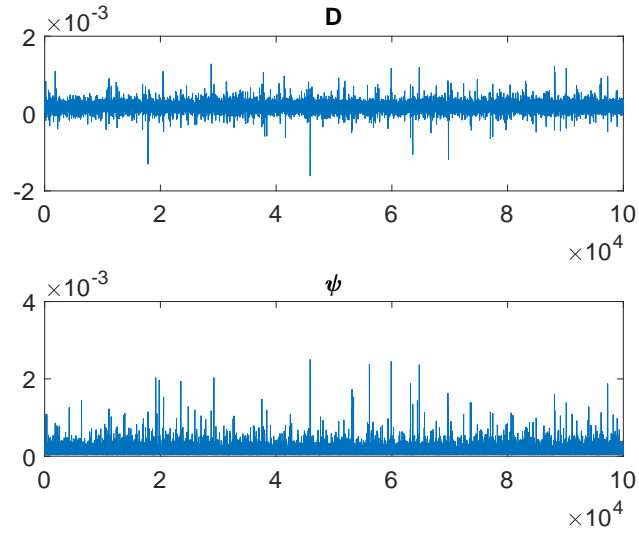


Figure 7.7: DRAM results for hyperparameters, continued.

Table 7.6: The residual norms from mixed-effects DRAM.

Experiment Number	$\omega^{A\overline{w}}_{in}$	Residual Norm
B11	0.01254	1.0572×10^{-3}
B7	0.02541	2.9405×10^{-3}
B8	0.1002	9.4464×10^{-3}
B14	0.1508	1.4805×10^{-2}

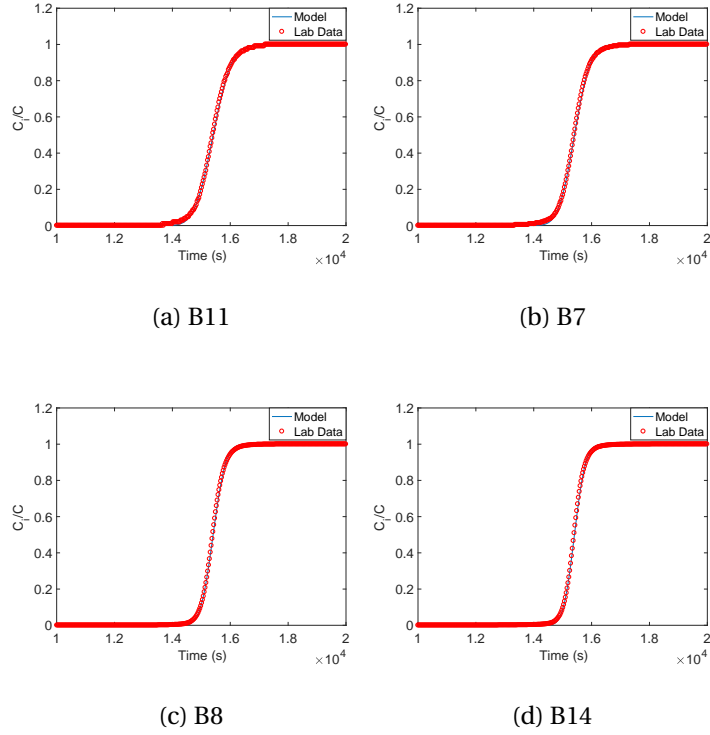


Figure 7.8: Mixed-effects responses compared to data.

7.6 Comparison of Bayesian and Frequentist Parameter Estimation Results

Table 7.7 lists the results from both the frequentist and Bayesian simulations. We can see that the results are generally in good agreement, with the hyperparameter estimates being within approximately 1-2% of one another. The Bayesian mixed-effects model predicts a larger variance and covariance between the hyperparameters.

Table 7.7: Estimated parameter values for TCAT from `nlmefit` and DRAM.

	<code>nlmefit</code>	DRAM
D	1.7531×10^{-4}	1.7539×10^{-4}
D_1	2.7199×10^{-4}	2.7199×10^{-4}
D_2	2.0195×10^{-4}	2.0199×10^{-4}
D_3	1.2710×10^{-4}	1.2708×10^{-4}
D_4	1.0019×10^{-4}	1.0021×10^{-4}
σ^2	5.7640×10^{-4}	6.4746×10^{-4}
ψ	7.7211×10^{-5}	1.0711×10^{-4}

7.7 Comparison of Mixed-Effects and Simplified TCAT Models

In this section, we compare results of the frequentist and Bayesian approaches for the simplified TCAT model as well as the mixed-effects TCAT model. Results are tabulated in Table 7.8. For each experiment, we tabulated the optimal value of D , then computed the residual norm. In all cases, the residual norms were in exact agreement to five figures. Therefore, we computed the difference between the residual norms in full precision and listed their respective differences. In all cases, the mixed-effects simulations slightly outperformed their counterparts. The values of D and residual norms being in close agreement shows that all methods used to simulate these models are consistently performing well.

Table 7.8: Comparison of mixed-effects model to the simplified TCAT model.

Experiment Number	Quantity	Frequentist		Bayesian	
		lsqnonlin	nlmefit	DRAM	Mixed-Effects DRAM
B11	D	2.7208×10^{-4}	2.7199×10^{-4}	2.7233×10^{-4}	2.7203×10^{-4}
	Residual Norm	1.5072×10^{-3}	1.5072×10^{-3}	1.5072×10^{-3}	1.5072×10^{-3}
	Difference in Residuals	1.5328×10^{-10}		3.2061×10^{-8}	
B7	D	2.0195×10^{-4}	2.0195×10^{-4}	2.0212×10^{-4}	2.0195×10^{-4}
	Residual Norm	2.9405×10^{-3}	2.9405×10^{-3}	2.9405×10^{-3}	2.9405×10^{-3}
	Difference in Residuals	4.1528×10^{-12}		2.9575×10^{-8}	
B8	D	1.2709×10^{-4}	1.2710×10^{-4}	1.2721×10^{-4}	1.2708×10^{-4}
	Residual Norm	9.4464×10^{-3}	9.4464×10^{-3}	9.4464×10^{-3}	9.4464×10^{-3}
	Difference in Residuals	7.9039×10^{-10}		1.6996×10^{-7}	
B14	D	1.0017×10^{-4}	1.0019×10^{-4}	1.0029×10^{-4}	1.0019×10^{-4}
	Residual Norm	1.4805×10^{-2}	1.4805×10^{-2}	1.4805×10^{-2}	1.4805×10^{-2}
	Difference in Residuals	1.7610×10^{-9}		2.8722×10^{-7}	

CHAPTER

8

CONCLUSIONS AND FUTURE WORK

This dissertation presents an exploration of a new class of models for single-fluid-phase-transport in porous media.

We reformulated the TCAT model as a partial differential-algebraic equation and used the method of lines to rewrite the PDAE as an ordinary DAE. From there, we used an advanced ODE solver, `ode15s`, to simulate the TCAT model. Our contributions included deriving a Jacobian pattern based on our chosen finite difference approximations, making changes deep within `ode15s` so that our simulations could run to completion, and performing a grid refinement study for the adaptive gridding method developed by Timothy Weigand.

We first used this numerical model to perform parameter estimation from a frequentist perspective using the Levenberg-Marquardt algorithm. We found that while our model could not accurately describe the wide range of available data with a single parameter value, we were able to find a single parameter that could describe the behavior of experiments with incoming mass fraction less than 0.15. We also performed individual parameter fits for numerous experiments and found varying optimal values of the parameter as the incoming

concentration changed. Our next goal was to perform parameter estimation from a Bayesian perspective; however, we found that the prohibitively long run times for the TCAT model did not allow us to perform Bayesian parameter estimation within a reasonable amount of time.

Therefore, we constructed a regression-based surrogate model for the TCAT model. Our surrogate utilizes a cubic spline to emulate the parameter-dependent response of the TCAT model over time. Next, we used the surrogate to perform Bayesian parameter estimation. We found that the mean value of the parameter decreases with an increase in incoming concentration. We also discovered that the variance decreases with an increase in incoming concentration for the three lower concentration experiments, then increases for the highest concentration experiment considered, B14.

Finally, we performed both frequentist and Bayesian mixed-effects parameter estimation. Because the mixed-effects code that we used, `nlmefit`, does not support observation-dependent design, we created a mixed-effects model for the simplified version of the TCAT model. The simplified version of the TCAT model has an analytic solution that is independent of initial concentration, and also has an unknown parameter that describes diffusion in the system. We found that the values of the random effects decrease as the incoming concentration increases. From the Bayesian approach, we also found that the variance decreases as the incoming concentration increases.

Future directions for this work include performing repeated brine intrusion experiments for the same or closely related incoming concentrations. From there, one could quantify the observative error across repeated measurements and use error bars to illustrate the standard error within observations. One could also perform Bayesian inference across a set of repeated measurements, treating the different experiments as batches of data and fitting hyperparameters. Another potential direction for this project includes nondimensionalizing the TCAT equations before simulating them numerically. Nondimensionalized equations present parameters in terms of dimensionless numbers, which can yield further insights about the underlying physics in the model and also lead to reduced stiffness in the model being simulated.

BIBLIOGRAPHY

- [1] B.M Adams, K. Coleman, R.W. Hooper, A. Lewis, R.C. Smith, L.P. Swiler, P.J. Turinsky, and B.J. Williams. User guidelines and best practices for CASL VUQ analysis using dakota. *CASL-U-2016-XXXX-000*, 2016.
- [2] R. Al-Raoush and A. Papadopoulos. Representative elementary volume analysis of porous media using X-ray computed tomography. *Powder Technology*, 200(12):69 – 77, 2010.
- [3] U. Ascher and L. Petzold. *Computer Methods of Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, 1998.
- [4] K. S. Basniev, N. M. Dmitriev, and G. V. Chilingar. *Mechanics of Fluid Flow*. Wiley, Hoboken, NJ, 2012.
- [5] J. Bear. *Dynamics of Fluids in Porous Media*. Elsevier, New York, NY, 1972.
- [6] J. Bear, R.R. Ross, and M.S. Beljin. *Fundamentals of ground-water modeling*. Superfund Technology Support Center for Ground Water, Ada, Oklahoma, 1992.
- [7] K.J. Beers. *Numerical Methods for Chemical Engineering - Applications in MATLAB*. Cambridge University Press, New York, NY, 2007.
- [8] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Elsevier, Philadelphia, PA, 1997.
- [9] K. E. Brenan and B. E. Engquist. Backward differentiation approximations of nonlinear differential/algebraic systems. *Mathematics of Computation*, 51(184):659–676, 1988.
- [10] S.L. Campbell and E. Griepentrog. Solvability of general differential algebraic equations. *SIAM Journal on Scientific Computing*, 16(2):257–270, 1995.
- [11] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.
- [12] T.A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.
- [13] M.S. Eldred and D.M. Dunlavy. Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA*, 2006.
- [14] D. Foreman-Mackey, D.W. Hogg, D. Lang, and J. Goodman. Emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306–312, 2013.

- [15] G. Fox, S. Negrete-Yankelevich, and V. Sosa. *Ecological Statistics: Contemporary Theory and Application*. Oxford University Press, Oxford, 2015.
- [16] C. W. Gear and L. R. Petzold. ODE methods for the solution of differential/algebraic systems. *SIAM Journal on Numerical Analysis*, 21(4):716–728, 1984.
- [17] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall, Englewood Cliffs, NJ, 1971.
- [18] A. Gelman and D.B. Rubin. Thermodynamically constrained averaging theory approach for modeling flow and transport phenomena in porous medium systems: 1. motivation and overview. *Advances in Water Resources*, 28(2):457–511, 1992.
- [19] J.R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in Matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.
- [20] H. Goldstein. *Multilevel Statistical Models*. Arnold Publishers, London, 1999.
- [21] W. G. Gray and C. T. Miller. Inference from iterative simulation using multiple sequences. *Statistical Sciences*, 7(4):161 – 180, 2005.
- [22] W. G. Gray and C. T. Miller. Thermodynamically constrained averaging theory approach for modeling flow and transport in porous media systems: 3. single-fluid-phase flow. *Advances in Water Resources*, 31(3):681–711, 2006.
- [23] W. G. Gray and C. T. Miller. Thermodynamically constrained averaging theory approach for modeling flow and transport in porous media systems: 5. single-fluid-phase transport. *Advances in Water Resources*, 32(5):1745–1765, 2009.
- [24] W. G. Gray and C. T. Miller. *Introduction to the Thermodynamically Constrained Averaging Theory for Porous Medium Systems*. Springer, New York, NY, 2013.
- [25] R. T. Gregory. A method for deriving numerical differentiation formulas. *The American Mathematical Monthly*, 64(2):79–82, 1957.
- [26] J.E. Griffin and S.G. Walker. On adaptive Metropolis Hastings methods. *Statistics and Computing*, 23(1):123–134, 2013.
- [27] A. Grillo, D. Logashenkob, S. Stichela, and G. Wittuma. Simulation of density-driven flow in fractured porous media. *Advances in Water Resources*, 33(12):1494–1507, 2010.
- [28] B. Grimstad, B. Foss, R. Heddle, and M. Woodman. Global optimization of multiphase flow networks using spline surrogate models. *Computers and Chemical Engineering*, 84:237–254, 2016.

- [29] H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354, 2006.
- [30] S.M. Hassanizadeh. Experimental study of coupled flow and mass transport: A model validation exercise. *ModelCARE 90: Calibration and Reliability in Groundwater Modelling*, 2(195):241–250, 1990.
- [31] S.M. Hassanizadeh and A. Leijnse. On the modeling of brine transport in porous media. *Water Resources Research*, 24(3):321–330, 1988.
- [32] S.M. Hassanizadeh and A. Leijnse. A nonlinear theory of high-concentration-gradient dispersion in porous media. *Advances in Water Resources*, 18(4):203–215, 1995.
- [33] J.M. Honig. *Thermodynamics*. Elsevier, New York, NY, 1999.
- [34] D.B. Ingham and I. Pop. *Transport Phenomena in Porous Media III*. Transport phenomena in porous media. Elsevier Science, New York, NY, 2005.
- [35] I. Javandel, C. Doughty, and C.F. Tsang. *Groundwater Transport*. American Geophysical Union, Washington, D.C., 1984.
- [36] D. Kaplan. *Methodology in the Social Sciences: Bayesian Inference for the Social Sciences*. Guilford Press, New York, NY, 2009.
- [37] K.R. Kavanaugh. *Nonsmooth Nonlinearities in Applications from Hydrology*. PhD thesis, North Carolina State University, July 2003.
- [38] C.E. Kees. *Multiphase flow modeling with DAE/MOL methods*. PhD thesis, Univeristy of North Carolina, 2001.
- [39] C.T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, PA, 1999.
- [40] D. Kulasiri. *Stochastic Dynamics : Modeling Solute Transport in Porous Media*. Elsevier, Boston, MA, 2002.
- [41] L.A. Kurtz, R.E. Smith, C.L. Parks, and L.R. Boney. A comparison of the method of lines to finite difference techniques in solving time-dependent partial differential equations. *Computers and Fluids*, 6(2):49 – 70, 1978.
- [42] M. Laine. MCMC toolbox for Matlab, 2013. <http://helios.fmi.fi/~lainema/mcmc/>.
- [43] A.J. Landman, R. Schotting, A. Egorov, and D. Demidov. Density-dependent dispersion in heterogeneous porous media part II: Comparison with nonlinear models. *Advances in Water Resources*, 30(2):2481–2498, 2007.

- [44] H.S. Lee, C.J. Matthews and R.D. Braddock, G.C. Sander, and F. Gandola. A Matlab method of lines template for transport equations. *Environmental Modeling and Simulation*, 19(2):603–614, 2004.
- [45] B. Leimkuhler, L. R. Petzold, and C. W. Gear. Approximation methods for the consistent initialization of differential- algebraic equations. *SIAM Journal on Numerical Analysis*, 28(1):205–226, 1991.
- [46] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Applied Mathematics*, 2:164–168, 1944.
- [47] R.J. LeVeque. Finite difference methods for differential equations: Course notes, 2005. http://www.edisciplinas.usp.br/pluginfile.php/41896/mod_resource/content/1/LeVeque20Finite20Diff.pdf.
- [48] R.J. Leveque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, Philadelphia, PA, 2007.
- [49] A.L. Lewis. *Gradient-Free Active Subspace Construction and Model Calibration Techniques for Complex Models*. PhD thesis, North Carolina State University, May 2015.
- [50] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [51] MathWorks. The matlab user’s guide: ode15s, 2014b. <http://www.mathworks.com/help/Matlab/ref/ode15s.html>.
- [52] MathWorks. The matlab user’s guide: polyfit, 2016a. <http://www.mathworks.com/help/Matlab/ref/polyfit.html>.
- [53] MathWorks. Documentation: fitlmematrix, 2016b. <https://www.mathworks.com/help/stats/nlmefit.html?requestedDomain=www.mathworks.com>.
- [54] MathWorks. *Estimating Parameters in Linear Mixed-Effects Models*. Statistics Toolbox: User’s Guide, 2016b. <https://www.mathworks.com/help/stats/estimating-parameters-in-linear-mixed-effects-models.html?requestedDomain=www.mathworks.com>.
- [55] MathWorks. Linear mixed effects models, 2016b. <https://www.mathworks.com/help/stats/nlmefit.html?requestedDomain=www.mathworks.com>.
- [56] MathWorks. The matlab user’s guide: spline, 2016b. <http://www.mathworks.com/help/Matlab/ref/spline.html>.

- [57] MathWorks. Nonlinear mixed effects models – nlmeFit, 2016b. <https://www.mathworks.com/help/stats/nlmeFit.html?requestedDomain=www.mathworks.com>.
- [58] J.J. Moré. The levenberg-marquardt algorithm: implementation and theory. *Numerical Analysis*, 630(5):105–116, 1978.
- [59] J. Mueller. Surrogate model types, 2016. courses.cit.cornell.edu/jmueller/Lecture_SurrogateModelTypes.pdf.
- [60] J.W. Murray. Chapter 6: Activity scales and activity corrections, 2001. www.ocean.washington.edu/courses/oc421/Lecture_Notes/CHPT6.pdf.
- [61] D.J. Poehls and G. J. Smith. *Encyclopedic Dictionary of Hydrogeology*. Elsevier, Boston, 2005.
- [62] A. Raoof and S. M. Hassanizadeh. Saturation-dependent solute dispersivity in porous media: Pore-scale processes. *Water Resources Research*, 49(4):1943–1951, 2013.
- [63] J.P. Reese. *Examining the Significance of Advective Acceleration to Single-Phase Flow Through Heterogeneous Porous Media*. PhD thesis, North Carolina State University, November 2006.
- [64] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- [65] P. Saucez, W.E. Schiesser, and A.V. Wouwer. Upwinding in the method of lines. *Mathematics and Computers in Simulation*, 56(2):171 – 185, 2001.
- [66] K.L. Schmidt. *Uncertainty Quantification for Mixed-Effects Models with Applications in Nuclear Engineering*. PhD thesis, North Carolina State University, August 2016.
- [67] R.J. Schotting, H. Moser, and M. Hassanizadeh. High-concentration-gradient dispersion in porous media: experiments, analysis and approximations. *Advances in Water Resources*, 22(7):665–680, 1999.
- [68] L.F. Shampine. Design of software for ODEs. *Journal of Computational and Applied Mathematics*, 205(2):901 – 911, 2007.
- [69] L.F. Shampine and M.W. Reichlet. The Matlab ODE suite. *SIAM Journal on Scientific Computation*, 18(1):1–22, 1997.
- [70] L.F. Shampine, M.W. Reichlet, and J.A. Kierzenka. Solving index-1 DAEs in Matlab and simulink. *SIAM Review*, 41(3):538–552, 1999.

- [71] R. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM, Philadelphia, PA, 2014.
- [72] A. Solonen. Monte Carlo Methods in Parameter Estimation of Nonlinear Models. Master's thesis, Lappeenranta University of Technology, December 2006.
- [73] A. Spitzbart and N. Macon. Numerical differentiation formulas. *The American Mathematical Monthly*, 64(10):721–723, 1957.
- [74] J. Steuben, J. Michopolous, A. Iliopoulos, and C. Turner. Inverse characterization of composite materials via surrogate modeling. *Composite Structures*, 132:694–708, 2015.
- [75] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22(4):1701–1728, 12 1994.
- [76] S.J. Watson and D.A. Barry. Numerical analysis of stable brine displacement for evaluation of density-dependent flow theory. *Advances in Water Resources*, 28(2):161 – 180, 2005.
- [77] S.J. Watson, D.A. Barry, R.J. Schotting, and S.M. Hassanizadeh. Validation of classical density-dependent solute transport theory for stable, high-concentration-gradient brine displacements in coarse and medium sands. *Advances in Water Resources*, 25(6):611–635, 2002.
- [78] T.M. Weigand, P.B. Schultz, D.H. Giffen, C.T. Kelley, and C.T. Miller. Modeling of density-dependent flow based on the thermodynamically constrained averaging theory. personal communication, 2016.
- [79] C.W. Winton. *Parameter Estimation in Groundwater Models Using Proper Orthogonal Decomposition*. PhD thesis, North Carolina State University, August 2012.
- [80] M. Wood, C.T. Simmons, and J.L. Huston. A breakthrough curve analysis of unstable density-driven flow and transport in homogeneous porous media. *Water Resources Research*, 40(3):1–9, 2004.
- [81] A.V. Wouwera, P.Saucez, W.E. Schiesser, and S.Thompson. A Matlab implementation of upwind finite differences and adaptive grids in the method of lines. *Journal of Computational and Applied Mathematics*, 183(2):245–258, 2005.
- [82] L. Wu. *Mixed Effects Models for Complex Data*. CRC Press, Boca Raton, FL, 2010.