

ABSTRACT

Numerical Modeling, Uncertainty Analyses, and Machine Learning for Decision Support in the Geosciences

Bulbul Ahmmmed, Ph.D.

Co-mentor: Scott C. James, Ph.D.

Co-mentor: John A. Dunbar, Ph.D.

My first paper shows the importance of numerical modeling and post-calibration uncertainty analyses for making decision to monitor waste transport at a naval waste repository site in Texas. For this, MODFLOW and MODPATH were used to simulate hydraulic head and particle/tracer travel times. Later, linear and nonlinear uncertainties were quantified for model parameters (hydraulic conductivities) and prediction of particle travel times along with identifiability and observation worth. Parameter uncertainties were reduced by up to 92%; a total of 19 parameters were at least moderately identifiable (>10%); travel-time uncertainties were reduced up to 92%. An observations-worth analysis found that 11 additional measurements at targeted locations could reduce travel-time uncertainties by factors from 1.04 to 4.3 over existing data. Finally, nonlinear uncertainty analyses predicted that conservative tracers exited the flow system within a year. My second paper explains a module for PFLOTTRAN, PFLOTTRAN-SIP, which was built to efficiently simulate waste remediation activities. PFLOTTRAN-SIP coupled PFLOTTRAN and E4D. PFLOTTRAN solves coupled flow and solute transport process models to estimate solute concentrations, which were used with Archie's Law to compute bulk electrical conductivities at near-zero frequency. These bulk electrical conductivities were modified using the Cole-Cole equation to account

for frequency dependence. Using the estimated frequency-dependent bulk conductivities, E4D simulates the real and complex electrical potential signals for selected frequencies for spectral impedance polarization. The **PFLOTTRAN-SIP** framework was demonstrated through a synthetic tracer-transport model simulating tracer concentration and electrical impedances for four frequencies. My third paper compares 20 machine learning (ML) models to predict reactive-mixing phenomena in subsurface porous media. The 20 ML emulators included linear methods, Bayesian methods, ensemble learning methods, and a multilayer perceptron (MLP). The ML emulators were trained to classify the state of mixing and predict three quantities of interest (QoIs) characterizing species production and decay. Linear classifiers and regressors failed; however, ensemble methods (classifiers and regressors) and the MLP accurately classified the state of reactive mixing and the QoIs. Computationally, trained ML emulators were $\approx 10^5$ times faster than the high-fidelity numerical simulations. These three works either support or expedite decision making process in the geosciences.

Numerical Modeling, Uncertainty Analyses, and Machine Learning for Decision
Support in the Geosciences

by

Bulbul Ahmmmed, Ph.D.

A Dissertation

Approved by the Department of Geosciences

Steven Driese, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

Approved by the Dissertation Committee

Scott C. James, Ph.D., Co-mentor

John A. Dunbar, Ph.D., Co-mentor

Jay Pulliam, Ph.D.

Peter Allen, Ph.D.

Ron Morgan, Ph.D.

Accepted by the Graduate School
August 2020

J. Larry Lyon, Ph.D., Dean

Copyright © 2020 by Bulbul Ahmmmed
All rights reserved

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	x
ACKNOWLEDGMENTS	xi
DEDICATION	xii
CHAPTER ONE	
Introduction	1
CHAPTER TWO	
Manuscript 1: NWIRP	7
Abstract	7
Introduction	7
Theoretical Background	12
Methods	14
Results	19
Conclusions	30
CHAPTER THREE	
Manuscript 2: PFLOTRAN-SIP	32
Abstract	32
Introduction	33
PFLOTRAN-SIP: Process Models and Coupling Framework	36
PFLOTRAN Process Models	39
PFLOTRAN-SIP Coupling	40
Petrophysical Transformation	41
Mesh Interpolation	42
Numerical Model Setup	43
SIP Inversion of Electrical Conductivity	45
Results	46
Discussion	50
Conclusions	51
CHAPTER FOUR	
Manuscript 3: ML to Reactive-transport Data	59
Abstract	59
Introduction	60
Governing Equations for Reactive Mixing	62

Machine Learning Emulators	66
ML Emulators.....	68
Results.....	71
Discussion	76
Conclusions.....	77
 CHAPTER FIVE	
Conclusions	89
 APPENDIX A	
NWIRP	94
 APPENDIX B	
Manuscript: ML to reactive-transport data	99
Variables.....	99
Greek Symbols	101
Linear Emulators	102
Ensemble ML Emulators	107
Artificial Neural Networks	110
 BIBLIOGRAPHY	112

LIST OF FIGURES

Figure 2.1.	Study location including streams, 43 observation wells (open circles), 99 hydraulic conductivity measurements (blue triangles), 77 pilot points (filled circles), and particle-release locations (green squares) with associated NWIRP administrative designations G, H, L, M, and S.	9
Figure 2.2.	Flow hydrograph from an unnamed spring in the study area near McGregor, Texas with no contribution from surface runoff (Clark, 2000, Fig. 2). Flow was measured with a wire and a pressure transducer data logger. The sharp peak in discharge indicates an increase in saturated thickness and gradient due to recharge and water rising into the upper, more conductive layer.	10
Figure 2.3.	Hydraulic heads in eight observation wells near administrative area M demonstrating the correlation with precipitation. Precipitation data: (Waco Regional Airport, 2016).	11
Figure 2.4.	(a) Schematics of the model domain with undulating layers and streams. (b) Cross sections during dry seasons, contaminants travel slower in the less-fractured, lower layer and during the wet season, they travel faster in the more fractured, upper layer.	11
Figure 2.5.	Picture of the fractured formation in the study area (scale is 0.5 m^2).	16
Figure 2.6.	Cross-plot of measured and calibrated heads.	20
Figure 2.7.	Reductions in pre-calibration uncertainties for the (a) upper- and (b) lower-layer pilot points upon application of the observation data set (43 head measurements).	21
Figure 2.8.	The 20 most identifiable parameters.	22
Figure 2.9.	Normalized observation worth of (a) monitoring wells (b) plus 11 hypothetical wells, symbol color and size indicate the observation worth; and (c) uncertainty reductions in travel times due to the existing wells (gray) and as augmented by 11 hypothetical wells (red).	24

Figure 2.10. Variances of pilot point, $\log_{10}(k)$, for the (a) upper and (b) lower layers for the pre-calibration (black) and NSMC (red) parameter realizations.	25
Figure 2.11. PDFs of highly identifiable parameters k_{u_5} and $k_{u_{45}}$ (top row), moderately identifiable parameters $k_{u_{39}}$ and $k_{u_{67}}$ (middle row), and minimally identifiable parameters $k_{l_{37}}$ and $k_{l_{41}}$ (bottom row). ...	26
Figure 2.12. PDFs of particle travel times.	27
Figure 2.13. Particle paths for 250 realizations with background color a representative hydraulic conductivity distribution.	28
Figure 3.1. Peer and child process model class of PFLOTTRAN (redrawn from reference (Johnson, Hammond, & Chen, 2017)).	52
Figure 3.2. Steps involved in coupling fluid flow, solute transport, and SIP process models in the PFLOTTRAN-SIP framework; further details are available (Johnson & Thomle, 2017; Lichtner et al., 2015).....	54
Figure 3.3. Schematic of interpolation of state variables (e.g., solute concentration) on the PFLOTTRAN mesh (cube) on to E4D mesh (tetrahedron), redrawn from (Johnson et al., 2017).....	54
Figure 3.4. Schematic of the initial boundary value problem.	55
Figure 3.5. Spatial distribution of tracer concentrations after one year.	55
Figure 3.6. Slices of simulated real (top) and complex (bottom) electrical potentials/impedances at $y = 250$ m for a single electrode-measurement configuration after one year.	56
Figure 3.7. Simulated and estimated frequency-dependent electrical conductivities at $y = 250$ m after one year (a)-(d) True-electrical conductivities from the PFLOTTRAN-SIP framework, (e)-(h) estimated bulk-real conductivities from SIP inversion, and (i)-(l) estimated bulk complex electrical conductivities from SIP inversion.	57
Figure 3.8. Distribution of (a) tracer concentration where I and V represents current and potential electrodes, respectively, (b) real potential, (c) complex potential, and (d) phase shift along the y -axis at $x = 250$ m and $z = -425$ m.	58

Figure 4.1. Schematic of the initial boundary value problem. L , $h_i^P(\mathbf{x}, t)$, c_A^0 , and c_B^0 are the length of the domain, diffusive flux on the boundary for i^{th} chemical species, initial concentration of species A , and initial concentration of species B , respectively. Species A and B were initially on the left and right sides of the domain, respectively. Initial concentrations of A and B were 1.0 and mixing commenced for $t > 0$	79
Figure 4.2. Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 10^3$ (high anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. Increased $\kappa_f L$ increases C production, especially at later times.	82
Figure 4.3. Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 100$ (medium anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. Lower anisotropy increased C production than higher anisotropy in Fig. 4.2.	83
Figure 4.4. Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 10$ (low anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. At low anisotropy, production of C increased. During late times (e.g., $t = 0.5$ and 1.0), diffusion dominates C production while $\kappa_f L$ and $\frac{\alpha_L}{\alpha_T}$ minimally affect C production.	84
Figure 4.5. Confusion matrices classifying the degree of mixing for the RF (left) and MLP (right) emulators.	85
Figure 4.6. This figure shows the true (markers) and RF emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A ; (d)–(f) of species B , and (g)–(i) of species C	86
Figure 4.7. This figure shows the true (markers) and GBM emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A ; (d)–(f) of species B , and (g)–(i) of species C	87
Figure 4.8. This figure shows the true (markers) and ANN emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A ; (d)–(f) of species B , and (g)–(i) of species C	88

LIST OF TABLES

<p>Table 2.1. Median travel times for the 882 pre-calibration and NSMC parameter fields used for particle tracking.....</p> <p>Table 4.1. Summary of training and testing data partitions used in ML emulator development and testing.</p> <p>Table 4.2. Hyperparameters and tunable parameters for generalized linear ML emulators, logistic regression, and KR with the best parameters in bold numbers and text in type-writer font.</p> <p>Table 4.3. Hyperparameters and tunable parameters for Bayesian emulators where bold numbers and text in type-writer font parameters were best suited parameters. Exponential sine squared ($\mathcal{K}(x, x') = \sigma^2 \exp(-2\sin^2(\pi x - x' /p)/l^2)$) is parameterized by a length-scale parameter (l) >0 and a periodicity (p) >0.</p> <p>Table 4.4. Hyperparameters and tunable parameters for ensemble ML emulators with the best parameters in bold numbers and text in type-writer font.....</p> <p>Table 4.5. Hyperparameters and tunable parameters for MLP emulator with the best parameters in bold numbers and text in type-writer font. ..</p> <p>Table 4.6. Performance metrics of ML emulators on training and test datasets for classifying the mixing state (i.e., degree of mixing) of the reaction-diffusion system.</p> <p>Table 4.7. Performance metrics of linear and Bayesian ML emulators (regressors). Note, GP and KR failed to converge even on 1% of training data because of a memory leak due to storage of a dense matrix.</p> <p>Table 4.8. Performance metrics of ensemble and MLP emulators.</p> <p>Table A.1. Locations and measurement of hydraulic conductivity (hk).</p> <p>Table A.2. Locations and average heads of observations.</p> <p>Table A.3. hk of pilot point (PP).</p>	<p>29</p> <p>68</p> <p>72</p> <p>72</p> <p>73</p> <p>73</p> <p>79</p> <p>80</p> <p>81</p> <p>94</p> <p>96</p> <p>97</p>
--	---

ACKNOWLEDGMENTS

At first, I thank my advisors to whom I am very much indebted, Drs. Scott C. James and John A. Dunbar. Their mentorship, motivation, wisdom comments on various processes helped me to accomplish such a great journey of my life. I also would like to thank my dissertation committee members, Drs. Peter Allen, Jay Pulliam, and Ron Morgan. I am indebted to Drs. Maruti K. Mudunuru, Satish Karra, and Hari Viswanathan of Los Alamos National Laboratory (LANL). I would like to give them special thanks for supporting me at LANL. I would like to thank Dr. Joe Yelderman for allowing me to participate in his fieldwork and shared his impressive geological knowledge with humor. Junayed Mahid and Joseph Thangraj helped me to collect core samples and I thank them for cooperating with me. I am also thankful to Tim Johnson and Hunter A. Knox of Pacific Northwest National Laboratory. I thank the Geological Society of America, the American Association Petroleum Geologist, and the Department of Geosciences at Baylor University for providing me funds for my research. I am indebted to Baylor graduate school for providing funds to attend professional and scientific conferences. I also thank Science Graduate Student Research (SCGSR) and Mickey Leland Energy Fellowship (MLEF) programs by the Department of Energy for providing me the fellowships. I appreciate Sandia and Los Alamos National Laboratories for hosting me during the summer of 2017 and 2018, respectively. I am very grateful to my loving parents who always bless me and my wife, Afroja Akter, who supports and encourages me in every minute. Finally, I acknowledge my ultimate source of inspiration, my son, Arno Ahmmmed, who inspires me to fight every second.

DEDICATION

To all COVID-19 fighters who won or lost the battle and the front-line workers who
helped both to win the battle!

CHAPTER ONE

Introduction

This dissertation is based on three manuscripts, each with a different application. This dissertation shows the importance of numerical models, uncertainty quantification (UQ), and machine learning (ML) for solving real world problems. The first manuscript (NWIRP) shows the importance of numerical modeling and post-calibration uncertainty analyses for supporting decision making regarding remediation activities. The second manuscript (PFLOTRAN-SIP) describes a code that coupled flow, reactive transport, and spectral impedance polarization (SIP) to simulate tracer flow, transport, and their electrical properties at different frequencies. The third manuscript (ML to reactive transport data) compared 20 machine learning models trained on reactive-transport data to predict species production and decay and to classify the degree of mixing. The first study applied numerical modeling and post-calibration uncertainty analyses to support remedial decision making of a contaminant geologic site. The study area is the Naval Weapons Industrial Reserve Plant (NWIRP), which occupies about 40 km² in southwest McGregor, Texas on a topographic divide underlain by a shallow groundwater system within fractured limestone bedrock. The NWIRP began manufacturing explosives in 1980 (Hare, 2000; T. L. Moore & McSpadden, 2009) and stored chemical waste in the vicinity of NWIRP. In 1998, several hazardous chemicals including ammonium perchlorate were discovered in lakes and streams surrounding the plant (Craig & Burdick, 2007; Ensafe Inc., 1999) above standard level defying estimates from groundwater velocities that suggested the contamination should not have entered streams and migrated offsite (Clark, 2000). The primary goal of this work was to predict tracer (a surrogate of hazardous chemicals) travel times and quantify uncertainty reduction of parameters

and predictions. To this end, a numerical groundwater model of the NWIRP site was developed to simulate hydraulic heads and to estimate tracer travel times.

Numerical modeling is a tool to simulate a real system that supports decision maker to make informed decision. However, often, numerical models fail to capture a complete physics of a system. To make a model useful, a comprehensive model interrogation is required. A comprehensive model interrogation consists of model calibration with existing data, predictive linear uncertainty analyses of parameters and predictions, estimating identifiabilities, quantifying data/observation worth, and nonlinear uncertainty analyses.

Model calibration process optimizes model parameters with existing data (here 43 hydraulic head measurements). Parameters represent site characteristics that were 154 hydraulic conductivities, K_n , and two horizontal anisotropies in this study. Predictive linear uncertainty reduction was calculated based on the pre- and post-calibration parameter uncertainties, where pre-calibration parameter variances (uncertainties) are specified according to measurements and expert judgment while post-calibration uncertainties are revealed through the calibration process. Identifiability is a metric indicating the level to which the calibration dataset constrains a parameter's value. Observation worth calculates how much a datum reduces uncertainties in parameters and predictions.

Moreover, nonlinear uncertainty analyses were accomplished through the Null-space Monte Carlo (NSMC) technique, which generated probability distributions of calibrated parameters and commensurate travel-time predictions (C. Moore, 2006; M. Tonkin & Doherty, 2009; M. Tonkin, Doherty, & Moore, 2007). The NSMC technique uses subspace approaches like singular value decomposition (C. Moore & Doherty, 2005) to identify only those model parameters informed by the observation dataset (C. Moore & Doherty, 2006). This facilitates inversion of over-parametrized models by only calibrating those variables about which the dataset has information

while relegating the rest to their user-preferred initial guesses through Tikhonov regularization (M. J. Tonkin & Doherty, 2005).

Using preceding techniques, this study addressed following questions: (1). How well can the groundwater flow model be calibrated using existing hydraulic-head measurements? (2). How well can the calibration-parameter values be identified? (3). How well can uncertainties in parameters and predictions be quantified? (4). Where should monitoring wells be drilled to minimize predictive uncertainty? (5). What are the estimated tracer travel times with nonlinear uncertainty bounds?

The second paper, PFLOTTRAN-SIP, describes a code that coupled PFLOTTRAN-SIP and E4D. PFLOTTRAN is a massively parallel flow and reactive-transport simulator while E4D is a parallel code that simulates and inverts electrical resistivities at different frequencies. PFLOTTRAN-SIP simulates flow, reactive-transport, and frequency-dependent electrical resistivities or SIP for subsurface media. The subsurface is dynamic due to natural and anthropogenic activities that alter porosity, permeability, fluid saturation, and geochemical properties over time (Council, 2000). Various geophysical techniques including seismic (deep or near-surface seismic) and potential-based methods (electromagnetic, magnetic, electrical resistivity tomography [ERT], and SIP) characterize changes in the subsurface (Kearey, Brooks, & Hill, 2013; Revil et al., 2011; Snieder et al., 2007). Among these, ERT and SIP map the distribution of bulk electrical conductivity (i.e., the reciprocal of resistivity) due to changes in fluid flow, temperature, deformation, and reactive transport (Byrdina, Rucker, Zimmer, Friedel, & Serfing, 2012; Carpenter, 2017; Gresse et al., 2017; Kaselow & Shapiro, 2004; Robinson, Johnson, & Slater, 2015). Because structural, topological, and geochemical properties (e.g., pore structures, fracture networks, electron donor, etc.) influence bulk electrical conductivity (Revil et al., 2011; Snieder et al., 2007), ERT and SIP are applied in environmental and energy industries to characterize subsurface interactions. Hence,

coupling ERT and/or SIP process models to flow and reactive-transport models can enhance interrogation of engineered subsurface systems.

SIP is a comprehensive method to extract subsurface polarization signals (e.g., flow, deformation, reactive transport, etc.) (Johnson et al., 2017). Detecting contaminants or chemical reactions in a system, a numerical inversion of SIP data is required (Vaudelot, Revil, Schmutz, Franceschi, & Bègassat, 2011). Flow and reactive-transport models can aid in constraining the inversion process of SIP data. To facilitate contaminant detection, this work coupled flow and reactive-transport simulator PFLOTTRAN and geoelectrical simulator E4D to include SIP in a framework called PFLOTTRAN-SIP. Here, in a porous medium with polarization properties, PFLOTTRAN-SIP was demonstrated with a representative tracer-transport model to provide more information than the ERT model.

The third paper applied ML emulators to predict reactive-transport outcomes and state of reaction. Reactive-transport simulation is a critical component for making decision related to applications such as CO₂ sequestration, spill distribution, algal-bloom forecasting, etc (Ahmmmed, 2015; Lagneau, Regnault, & Descostes, 2019; Lichtner, Steefel, & Oelkers, 2019; Molins & Knabner, 2019; Sin & Corvisier, 2019). Reactive-transport simulation computes species precipitation/dissolution (QoIs), which are critical to remediation activities. For QoIs, nonlinear partial differential equations are solved using high-fidelity numerical methods (e.g., finite-difference, -element, or -volume methods) that can take hours to days (for $\approx \mathcal{O}(10^6) - \mathcal{O}(10^9)$ degrees-of-freedom) on state-of-the-art, high-performance computing machines. Such computation times preclude real-time predictions, which can be critical to decision making for remediation activities. Hence, alternative faster approaches are needed and machine learning (ML)-based emulators show promise (Hulbert et al., 2019; Srinivasan et al., 2018; Valls et al., 2018; Viswanathan et al., 2018; Wu, Lin, Zhou, & Delorey, 2018). To make faster predictions, this study built and compared 20 ML emulators

to predict reactive-transport QoIs. The ML emulators were trained and tested using data from high-fidelity, finite-element numerical simulations, which explicitly reflect the underlying reaction-diffusion physics in anisotropic porous media.

Given appropriate and sufficient data, ML models can successfully detect, quantify, and predict different types of phenomena in the geosciences (Bergen, Johnson, Maarten, & Beroza, 2019; Reichstein et al., 2019). ML has been successfully applied in remote sensing (Mesa, Reichstein, Mahecha, Kraft, & Denzler, 2018; Valls et al., 2018), ocean wave forecasting (James, Zhang, & O'Donncha, 2018; O'Donncha, Zhang, Chen, & James, 2018, 2019), seismology (Hulbert et al., 2019; Leduc et al., 2017; M.-Zook & Ruppert, 2017; Reynen & Audet, 2017; Wu et al., 2018; Yuan et al., 2019), hydrogeology (Barzegar, Moghaddam, Deo, Fijani, & Tziritis, 2018; Srinivasan et al., 2018; Viswanathan et al., 2018), and geochemistry (Cracknell, Reading, & McNeill, 2014; Galiano, Castillo, Olmo, & Rivas, 2015; Kirkwood, Cave, Beamish, Grebby, & Ferreira, 2016; Oonk & Spijker, 2015; Zuo, 2017). ML emulators, surrogate models, or reduced-order models can be fast, reliable, and robust when trained on large datasets (Bergen et al., 2019; Reichstein et al., 2019; Salah, 2018).

ML models are classified as supervised and unsupervised algorithms (Knox, 2018; Müller & Guido, 2016). Supervised ML models learn relationship/function from data instead of solving specified functions/equations. Supervised ML models are constructed using training data (e.g., features and labels), which include inputs and outputs either from field data, experimental data, high-fidelity numerical simulations, or any combination of these (Brunton & Kutz, 2019; Salah, 2018). The supervised ML algorithm is designed to generalize to predict unseen data. Supervised ML can be used for classification and/or regression. For example, detecting spam emails, identifying tumors, and finding fraudulent activity on credit cards are classification problems while predicting a stock value is a regression problem. On the other hand, an unsupervised ML model is suitable for analyzing hidden properties in data. In unsupervised

ML, only input (or features) are known and no labels are provided. For example, topic identification in a blog and clustering customers into groups with similar preference are accomplished with unsupervised ML.

This study only applied supervised ML models to reactive-transport data to predict QoIs and to classify the state of reactive transport. The 20 ML emulators included a linear classifier, two Bayesian classifiers, an ensemble classifier, an MLP classifier, seven linear regressors, six ensemble regressors, and an MLP regressor. Emulator performance was assessed according to training and testing scores, training times, and R^2 scores on the QoIs from a blind dataset. The blind dataset included six realizations that were not presented to the algorithms during training and testing phases.

CHAPTER TWO

Manuscript 1: NWIRP

Abstract

The Naval Weapons Industrial Reserve Plant (NWIRP) in McGregor, Texas began manufacturing explosives in 1980 and several hazardous chemicals were discovered in lakes and streams surrounding the plant in 1998. This research demonstrates the importance of using a numerical study to support remedial decision making by investigating post-calibration linear and Null-space Monte Carlo (NSMC) nonlinear uncertainty analyses. Based on MODFLOW and MODPATH models, which simulated hydraulic heads and tracer travel times at the site, the following measures were quantified: parameter uncertainties, parameter identifiabilities, observation worth, and predictive uncertainties. Parameter uncertainties were reduced by up to 92%; a total of 19 parameters were at least moderately identifiable ($>10\%$); travel-time uncertainties were reduced up to 92%. An observations-worth analysis found that additional data (11 more measurements) could reduce travel-time uncertainties by factors from 1.04 to 4.3 over existing data if collected at targeted locations. Finally, travel-time predictions and post-calibration parameter distributions were generated using the NSMC technique. NSMC predicted that conservative tracers exited the flow system within a year, which agrees well with field data.

Introduction

The Naval Weapons Industrial Reserve Plant (NWIRP) occupies about 40 km^2 in southwest McGregor, Texas on a topographic divide underlain by a shallow groundwater system within fractured limestone bedrock (Figure 2.1). The NWIRP began manufacturing explosives in 1980 (Hare, 2000; T. L. Moore & McSpadden, 2009)

and several hazardous chemicals including ammonium perchlorate were discovered in lakes and streams surrounding the plant in 1998 (Craig & Burdick, 2007; Ensafe Inc., 1999) defying estimates from groundwater velocities that suggested the contamination should not have entered streams and migrated offsite (Clark, 2000). However, those estimates did not consider the increased fluxes and hydraulic heads affecting groundwater flow velocities during storm periods. Ensafe (1999) (Ensafe Inc., 1999) estimated groundwater flux in Georgetown Limestone at 2 m/year using average gradients, hydraulic conductivity from slug-test data, and porosity while assuming homogeneous and steady groundwater velocities throughout the area. But groundwater in Georgetown Limestone also flows through more conductive features when the water table rises during storm periods. Such recharge also increases groundwater velocity. The Georgetown Limestone, similar to other fractured carbonates like the Austin Chalk, exists as an upper, highly fractured and unsaturated zone overlying a low-permeability, moderately fractured zone (Ashworth & Hopkins, 1995; Barquest, 1989; Bingham, 1993; Chowdhury, Osting, Furdan, & Mathews, 2010; Mace, 1998). The highly fractured upper layer facilitates fast fluid flow during storms. Although no study was performed on variance of flux but an inference can be drawn from Figure 2.2 about the variability of the flow rate. As shown in Figure 2.2, the hydrograph from an unnamed spring, the mean flow rate from May through September, 1999 was 3.3 L/s with a spike to 18.7 L/s in July. The sharp response of the flow rate to the recharge event reflects the water table entering the upper, highly fractured layer, which was then quickly drained. Although the shallow groundwater is not used locally, the streams flow into two major water supply reservoirs in the region (Lake Belton and Lake Waco).

Recharge from precipitation significantly increases lateral flow through the Georgetown Limestone when the water table rises into the upper, highly fractured zone. The water table is sensitive to recharge (storms) as shown in Figure 2.3. A rising water

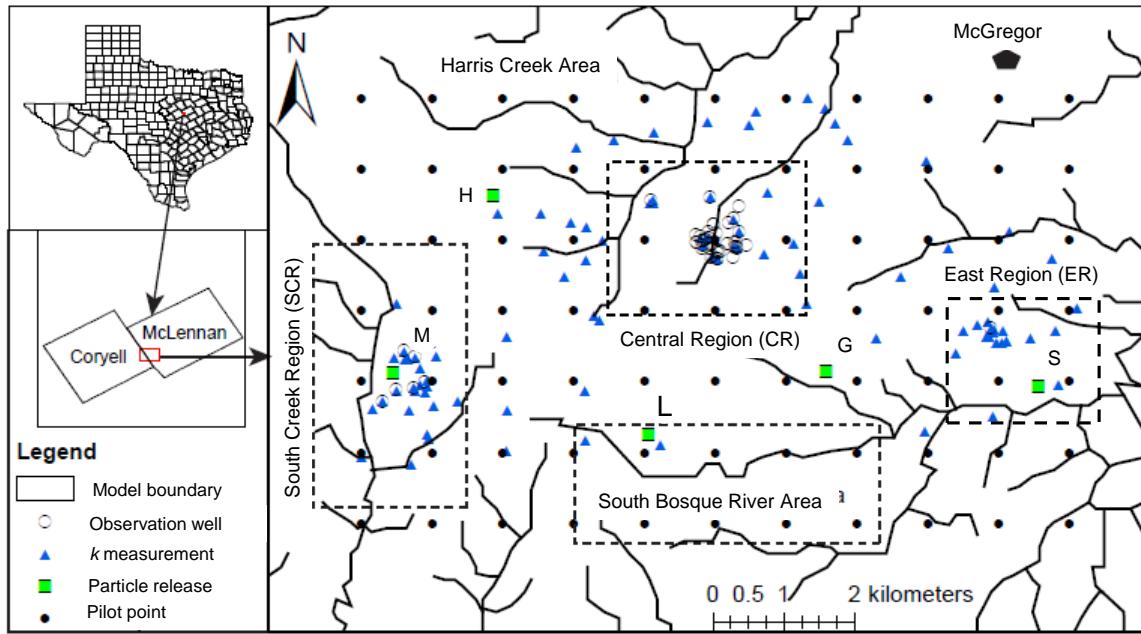


Figure 2.1: Study location including streams, 43 observation wells (open circles), 99 hydraulic conductivity measurements (blue triangles), 77 pilot points (filled circles), and particle-release locations (green squares) with associated NWIRP administrative designations G, H, L, M, and S.

table can mobilize dissolved perchlorate such that it enters the upper zone where it is more easily transported off site. Even though some dissolved perchlorate is transported off site, the source persists as residual perchlorate in the lower fractured zone awaiting remobilization during the next storm (see Figure 2.4).

Tracer (or contaminant) transport times from the NWIRP to surrounding streams, rivers, and lakes are the primary concern for stakeholders. Clark (2000) (Clark, 2000) built a piecewise-homogeneous MODFLOW model of the NWIRP, which was updated here to a pilot-point-based, heterogeneous MODFLOW model to predict flow fields while MODPATH estimated tracer transport times. MODFLOW is a ground-water modeling tools developed by United States Geological Survey and MODPATH (Pollock, 2016) is a particle-tracking code that uses boundary conditions, hydraulic heads, and parameters from MODFLOW to simulate flow streamlines. The conceptual model included the upper and lower fractured layers, topography, and heteroge-

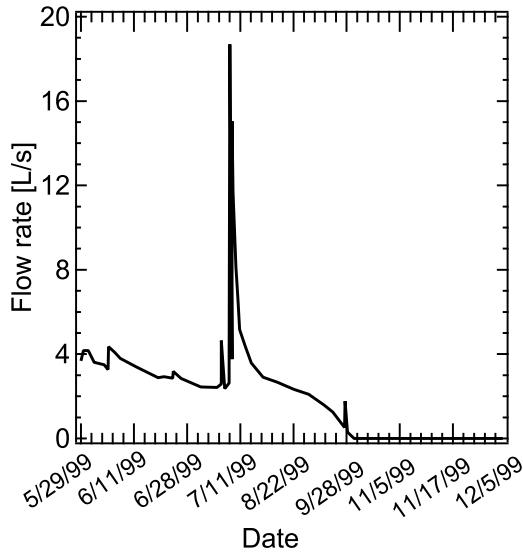


Figure 2.2: Flow hydrograph from an unnamed spring in the study area near McGregor, Texas with no contribution from surface runoff (Clark, 2000, Fig. 2). Flow was measured with a wire and a pressure transducer data logger. The sharp peak in discharge indicates an increase in saturated thickness and gradient due to recharge and water rising into the upper, more conductive layer.

neous hydraulic conductivities. The predictions of interest were the times for tracers from NWIRP administrative areas to exit the model domain as discharge to streams. Predictions were conditioned through calibration against measured water levels. Uncertainties in these predictions were quantified and the most important parameters and observations identified. Moreover, the Null-space Monte Carlo (NSMC) nonlinear uncertainty analysis technique was used to generate probability distributions of calibrated parameters and commensurate travel-time predictions (C. Moore, 2006; M. Tonkin & Doherty, 2009; M. Tonkin et al., 2007). Briefly, the NSMC technique uses subspace approaches like singular value decomposition (C. Moore & Doherty, 2005) to identify only those model parameters informed by the observation dataset (C. Moore & Doherty, 2006). This facilitates inversion of over-parametrized models by only calibrating those variables about which the dataset has information while re-

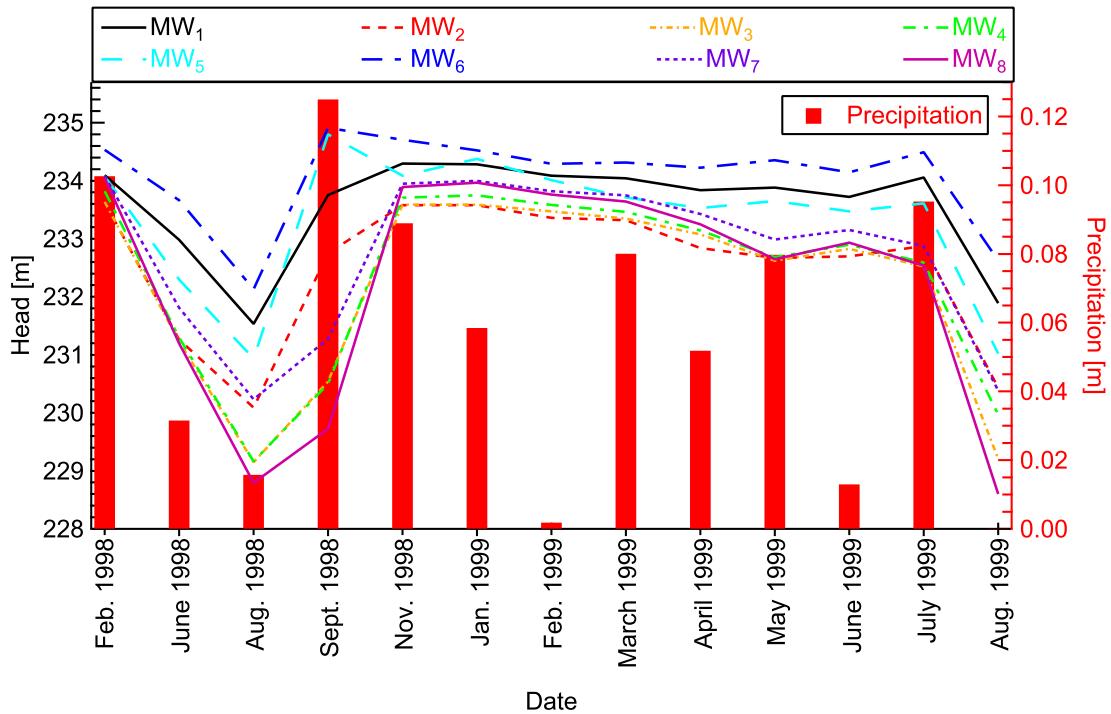


Figure 2.3: Hydraulic heads in eight observation wells near administrative area M demonstrating the correlation with precipitation. Precipitation data: (Waco Regional Airport, 2016).

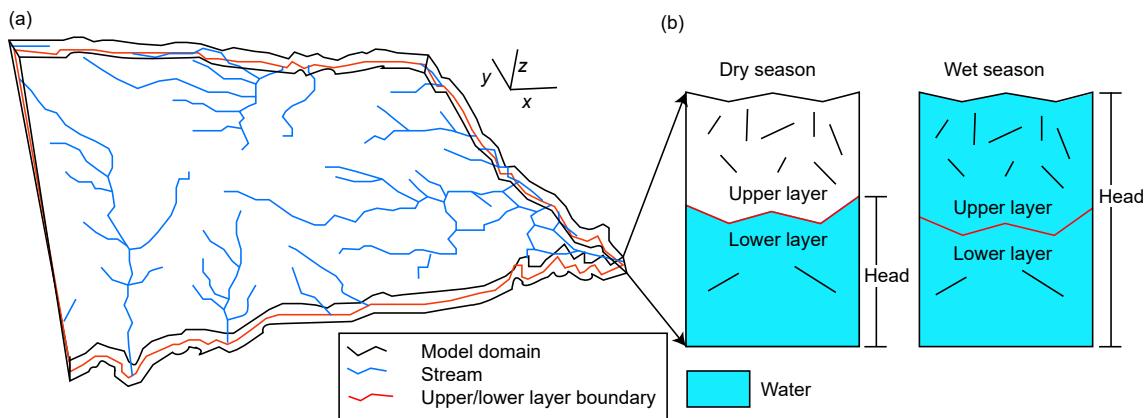


Figure 2.4: (a) Schematics of the model domain with undulating layers and streams. (b) Cross sections during dry seasons, contaminants travel slower in the less-fractured, lower layer and during the wet season, they travel faster in the more fractured, upper layer.

gating the rest to their user-preferred initial guesses through Tikhonov regularization (M. J. Tonkin & Doherty, 2005).

Theoretical Background

The level of parametrization of an environmental model should be commensurate with the quality and quantity of data used in its calibration to ensure confidence in the range of predictive possibilities (Hunt, Doherty, & Tonkin, 2007; C. Moore & Doherty, 2006). Calibration is constrained by the information content of the calibration data set (plus expert judgment) and linear predictive uncertainty can be assessed even before a calibration exercise. Parameter uncertainty and identifiability along with observation worth can be quantified (Doherty, 2016; James, Doherty, & Eddebbah, 2009). Post-calibration, the NSMC method facilitates a nonlinear assessment of parameter and prediction uncertainties (Doherty, 2016), but even with the use of super parameters (linear combinations of estimable parameters) to reduce the number of model calls, the approach can still be computationally intensive.

Predictive uncertainty analyses can be undertaken with a calibrated model using methods based on the propagation of variance (Doherty, 2016), which acknowledges that historic observation can be replicated with many non-unique parameter combinations. Predictive uncertainty reduction is calculated based on the pre- and post-calibration parameter uncertainties, where pre-calibration parameter variances (uncertainties) are specified according to measurements and expert judgment while post-calibration uncertainties are revealed through the calibration process. In addition, each parameter contributes to uncertainty in model predictions. Reduction in predictive uncertainty is predicated upon enhanced knowledge of the parameter space. Parameter uncertainty can be divided into solution- and null-space components. Solution-space uncertainty, usually the smaller of the two, is due to uncertainty in the calibration data (i.e., measurement error). Null-space uncertainty is due

to shortcomings in the data or model that preclude precise identification of the parameter (i.e., many parameter combinations can calibrate the model about equally well). The mathematical process of distinguishing solution- from null-space uncertainty is achieved through singular value decomposition (SVD), which is conducted with straightforward mathematical vector and matrix manipulations (Doherty, 2016; James et al., 2009).

Measurement errors (observation noise) can never be eliminated and these impact predictive uncertainties. The calibration process minimizes the weighted-sum-of-squares differences, the objective function, between site observations and their corresponding model predictions. Both quantitative (observation noise, measurement accuracy, number of measurements comprising an observation, etc.) and qualitative (expert judgment) metrics should be used to specify weights in the objective function.

Identifiability is a metric indicating the calibration data's ability to constrain a model parameter (Doherty & Hunt, 2009). Quantitatively, it is the direction cosine between a parameter and its projection onto solution-space uncertainty. Identifiability can be used in both model design and implementation to assess whether a model needs more calibration data to reduce parameter uncertainty while also quantifying the uncertainties in predictions that depend specifically upon that parameter.

Observation worth is quantified based on the reduction in uncertainty in a parameter or prediction that is accrued through the acquisition of that datum (Lotti & Doherty, 2016). Reduction in these uncertainties below their pre-calibration level is a measure of the worth of an observation (or observation group) with respect to that parameter or prediction.

The NSMC technique generates multiple, unique parameter fields that satisfy both the model-to-measurement misfit (i.e., a sufficiently low objective function) and parameter-reality constraints (i.e., parameters cannot be assigned unrealistic values) and it quantifies post-calibration parameter and prediction uncertainties (Welter,

White, Hunt, & Doherty, 2015). It generates a suite of equally likely and realistic parameter fields that are used to make predictions. Generating parameter fields involves three steps: (1) generating random parameter fields according to pre-calibration uncertainty, (2) perturbing pre-calibrated parameters by adding null-space uncertainty, and (3) a brief model calibration (usually three optimizations) using fewer parameters or (super parameters), which are linear combinations of those parameters that have their pre-calibration uncertainty reduced by more than an insignificant amount. Uncertainty in a prediction can thereby be assessed through construction of an empirical probability density function (PDF) assembled by running the model using each NSMC parameter field realization to generate PDFs of predictions.

Methods

Conceptual Model

The conceptual model was built using four digital elevation maps (DEMs) from the Texas Natural Resources Information System website (TNRIS, 1999) to create the model topography. GIS (ESRI, 1996) and SURFER (Golden Software, 1997) were used to mosaic and grid the DEM data (Clark, 2000). The conceptual model also included local rivers, streams, creeks, and spring as shown in Figure 2.4 (a). The model comprised two 4-m-thick layers representing the upper, weathered layer and the lower, less-permeable limestone. The NWIRP model domain and terrain-following layers were adjusted according to the topographic elevation (see Figure 2.4 (a)). The top of the upper model layer corresponded to the water level, which was initially assumed 2 m below the land surface. The bottom of the model was established uniformly 10 m below surface (Gburek, Folmar, & Urban, 1999; Verma, Rao, & Rao, 1980). Average precipitation from 1960 to 2015 was 2.7×10^{-8} m/s. A base-flow study conducted in a nearby similar geologic setting indicated that 7% of total precipitation infiltrated to the aquifer (Cannata, 1988; Myrick, 1989), so recharge was 1.8×10^{-9} m/s.

Model Development

Because the rock is densely fractured (Figure 2.5), it was approximated as an equivalent porous medium. Therefore, this effort started by developing MODFLOW and MODPATH models of the NWIRP site (Figure 2.1), calibrating to available data, assessing observation worth and parameter identifiability, as well as quantifying uncertainties in parameters and tracer travel-time predictions. Next, results from the NSMC approach were used to generate PDFs of pilot points (hydraulic conductivities) and travel times. The model had two layers, 126 columns, and 97 rows with $100 \times 100 \text{ m}^2$ cells. Recharge through precipitation was specified at the top of the model while the lateral sides along with the bottom of the model were specified as no-flow boundaries. Because all streams in the domain are gaining reaches, there was no need to specify streambed properties distinct from the hydraulic conductivities. No-flow boundaries can affect results near the edges of the model, so the NWIRP regions of interest were always at least 1.5 km from the model edges. The model was executed with MODFLOW-NWT (Niswonger, Panday, & Motomu, 2011), which admits the drying and re-wetting nonlinearities of an unconfined aquifer (J. Hughes, Langevin, Chartier, & White, 2012; Niswonger et al., 2011). Single particles were released at the midpoint of each layer from the five administrative areas and tracked until they exited the model domain at streams.

Parameters

Groundwater flow is fundamentally governed by the distribution of hydraulic conductivities. In conventional calibration methods, property uniformity or pilot-point distributions are used as the basis for spatial parameter distribution (Doherty, 2003). In the absence of data, piecewise-homogeneous zones are often specified. If geologic zones are not piecewise-uniform, pilot points are distributed throughout such zones. Pilot-point property values were estimated during this calibration exercise and



Figure 2.5: Picture of the fractured formation in the study area (scale is 0.5 m^2).

the hydraulic conductivities at model cells were assigned according to a kriging algorithm (Doherty, 2003). Pilot points facilitate a smooth but realistic distribution of hydraulic properties over a geologic unit, which cannot be achieved using piecewise-uniform methods. The upper model layer had only a single hydraulic conductivity measurement, but one parameter over such a large region would give false confidence in the solution because it would yield unrealistic homogeneity (Parker, 1977). Instead, a total of 77 pilot points (Figure 2.1) were used in each layer such that hydraulic conductivity fields were developed with heterogeneity commensurate with the information available in the observation data set. The initial value of the horizontal hydraulic conductivity in the upper layer, k_u , was $3.048 \times 10^{-3} \text{ m/s}$ for all 77 pilot points with a 1.5% porosity (Cannata, 1988). Just like the upper layer, a pilot-point-based heterogeneous distribution of hydraulic conductivities were also assigned to the lower layer. Ensafe (1999) (Ensafe Inc., 1999) conducted 99 slug tests measuring horizontal hydraulic conductivities in the lower layer, k_l , ranging from 10^{-8} to 10^{-4} m/s with mean 10^{-7} m/s . These hydraulic-conductivity measurements were used in an expo-

nential variogram with specified range and sill (variance) (Deutsch & Journel, 1992). The range and variance of log of hydraulic-conductivity measurements were 700 m and 1.52, respectively, and using the 99 measured hydraulic conductivities, they were kriged (interpolated) onto each model cells. The vertical anisotropies and porosity of the lower layer were specified as one tenth of horizontal hydraulic conductivities and 0.5% (Cannata, 1988), respectively.

The calculated range and sill were used to generate kriging factors for pilot points in the lower layer using the PPK2FAC utility in the PEST suit. Later, these factors were used to interpolate k onto the model grid using the FAC2REAL utility in the PEST suit. Because of exposure to weathering and erosional process, the upper layer is more heterogeneous even though it comprises similar rock types, so a larger variance is appropriate. Thus, a variance of 3.04 (twice that of the lower layer) was assigned to the upper layer with the same 700-m range as the lower layer. Parameter uncertainties and observation worth were calculated based on propagation of variance. Initially, a pre-calibration covariance matrix was calculated for the pilot points using the PPCOV (Doherty, 2016). The diagonal elements of the covariance matrix were the variance while off-diagonal elements were non-zero numbers based on 99 measured hydraulic conductivities and their geospatial characteristics. Later, this covariance matrix was used to generate pre-calibration pilot point realizations.

A total of 156 parameters were adjusted during calibration. Parameters were subdivided into three groups: (1) 77 pilot-point-based horizontal hydraulic conductivities for the upper ($k_{u1}-k_{u77}$), (2) and lower ($k_{l1}-k_{l77}$) layers, and (3) horizontal anisotropies for the upper, h_u , and lower, h_l , layers. Each hydraulic conductivity was assigned a pre-calibration lognormal probability distribution with mean of 3.048×10^{-3} m/s and 10^{-7} m/s for the upper and lower layers, respectively.

Calibration Data and Predictions of Interest

Because of model complexity and under-determinacy, regularization was used during calibration (Doherty, 2016) to reduce bias and to decrease the required number of model calls. A total of 156 log-transformed regularized parameters were calibrated. The calibration was performed against 43 steady-state heads (average water levels if multiple measurements were available) at the monitoring wells indicated with open circles in Figure 2.1. Each observation was assumed equally important (equal weight). Predictions of interest were travel times for 10 particles released at the midpoints of the upper and lower layers in the five administrative hazardous storage sites in Figure 2.1 (T. L. Moore & McSpadden, 2009). Travel times for the particles were simulated and their uncertainties were quantified.

Calibration and NSMC

The work flow for calibration and uncertainty analyses was: (1) parametrization and calibration of the NWIRP MODFLOW model; (2) parameter identifiabilities were calculated along with observation worth; (3) the effects of additional monitoring wells on travel-time predictions and observation worth were estimated; (4) uncertainties in travel-time predictions were then explored upon consideration of the new hypothetical observations; and (5) post-calibration uncertainties of pilot points and particle travel times were assessed using the NSMC technique.

Calibration estimated 156 parameters (154 pilot points and two horizontal anisotropies) with only 43 observations, which made this an ill-posed problem; not all parameters could be uniquely estimated (Moeck, Molson, & Schirmer, 2019; C. Moore & Doherty, 2005). Using SVD, it was determined that 25 unique linear combinations of parameters (super parameters) could be reasonably identified. In other words, the modeler presents all parameters (here 113 more than could possibly be identified by the 43 data points) to the calibration and PEST's subspace-regularization capa-

bilities determine which parameters can be uniquely identified by the dataset and focuses only on those. Here, the dataset informed 25 parameters, which can be specified as linear combinations if there are strong correlations between some of them. For over-parametrized models, an NSMC analysis affords a more accurate, nonlinear assessment of predictive uncertainty. The NSMC approach was executed in three steps. First, the `RANDPAR` utility in the PEST suite generated a total of 1,000 pilot-point realizations (the mean of each pilot point stabilized by 1,000 realizations) using a log-normal distribution with mean (from pump tests) and the pre-calibration covariance matrix. Second, each pre-calibrated parameter realization was perturbed by adding null-space uncertainty using the `PNULPAR` utility in the PEST suite. `PNULPAR` calculated the orthogonal differences between calibrated parameters and those produced with `RANDPAR`. Then, these differences were added to each realization of calibrated parameters to generate 1,000 realizations of calibrated parameters with perturbations in the null space. For each of these null-space-projected parameter-field realization, three iterations of a PEST calibration were undertaken (`NOPTMAX = 3`) using 25 super parameters. Each resulted in a calibration-constrained parameter-field realization (NSMC realization) with a corresponding objective function value, Φ , which indicated how closely the simulated heads from that model run matched their corresponding observation. Only those NSMC realizations with objective functions less than 150% of the calibrated objective function were retained to form distributions of parameters and corresponding particle transport times.

Results

Post-calibration Linear Uncertainty Analyses

Figure 2.6 compares measured (y axis) and calibrated (x axis) heads. There was a slight bias toward underprediction (-0.1 m) while the mean absolute error was 0.7 m

and the root-mean-squared error was 0.8 m, all of which indicated a well-calibrated model.

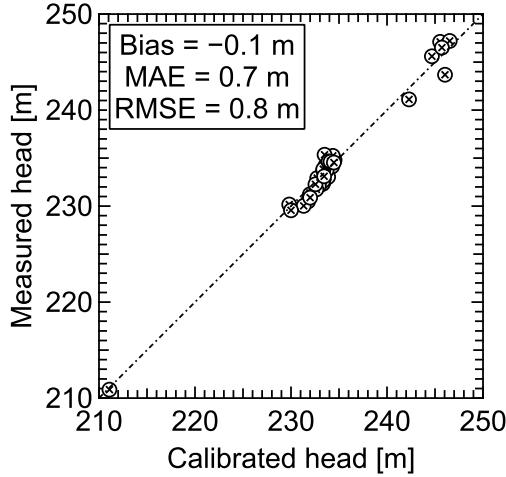


Figure 2.6: Cross-plot of measured and calibrated heads.

The linear uncertainty analysis identified reductions in parameter uncertainty (i.e., pilot-point hydraulic conductivities, so parameter uncertainty always refers to the reduction in the *a priori* hydraulic conductivity variance while prediction uncertainty refers to reduction in particle travel times realized through the reductions in hydraulic-conductivity uncertainty) subject to the information content in the calibration dataset. These post-calibration parameter uncertainties reflect the degrees to which the observations reduced the pre-calibration parameter uncertainties as indicated in Figure 2.7. The size and color of the circles indicate the percent reduction in pre-calibration uncertainty upon application of the observation data set. Pilot-point uncertainty reductions ranged from 0.4 to 92% with greater reductions nearer to observation wells in the upper layer (Figure 2.7a). The Central (CR), Station Creek (SCR), and East Regions (ER) of the model, indicated on Figure 2.1, contain 36, six, and one observation well(s), respectively. In the upper layer, observation wells in the CR significantly informed the six nearby pilot points by reducing their uncertainties from 15 to 92%. Similarly, the six wells in the SCR decreased uncertainties

in four nearby pilot points from 25 to 90% while the single well in the ER notably decreased uncertainty in the nearest pilot point by 50%. Uncertainty reduction for pilot points in the lower layer ranged from 0.3 to 30%. Observation wells in the CR reduced uncertainties from 1 to 30%; the uncertainty reductions were relatively lower in this layer because of their smaller (by half) variances 1.52 ($1.78 \times 10^{-10} \text{ m}^2\text{s}^{-2}$). The observation wells in SCR informed three nearby pilot points and reduced their uncertainties by $>10\%$ (Figure 2.7b). All told, uncertainty reductions for 120 of the pilot points were $<10\%$, typically for pilot points distant from observation wells. Overall, given that there were 43 unequally distributed monitoring wells and low initial uncertainties in the lower-layer pilot points, it was not surprising that only 36 pilot points had their uncertainties reduced by $>10\%$. Moreover, this is in accord with the 25 super-parameters.

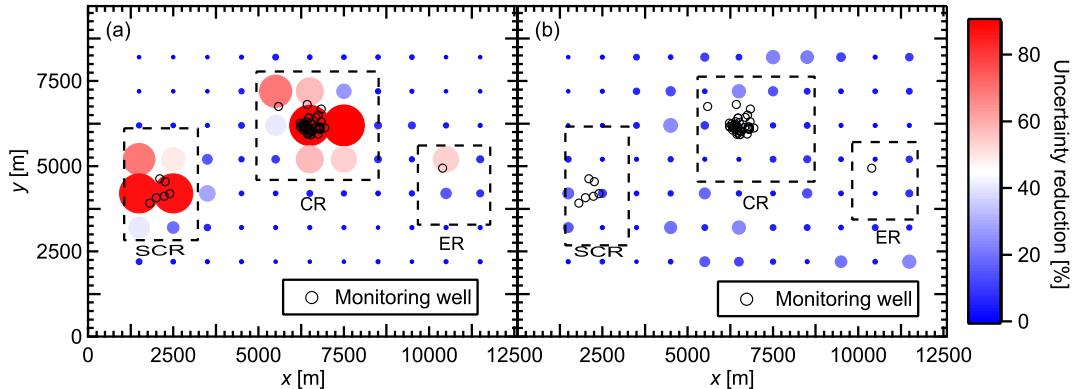


Figure 2.7: Reductions in pre-calibration uncertainties for the (a) upper- and (b) lower-layer pilot points upon application of the observation data set (43 head measurements).

Parameter identifiability apportions parameter uncertainties into solution and null spaces. An identifiability of zero means that the data set says nothing about that parameter while an identifiability of one means that uncertainty in that parameter is solely due to measurement error. Of the 156 parameters, seven had high (>0.5), 12 moderate (0.1 – 0.5), and 137 low (<0.1) identifiabilities (Figure 2.8).

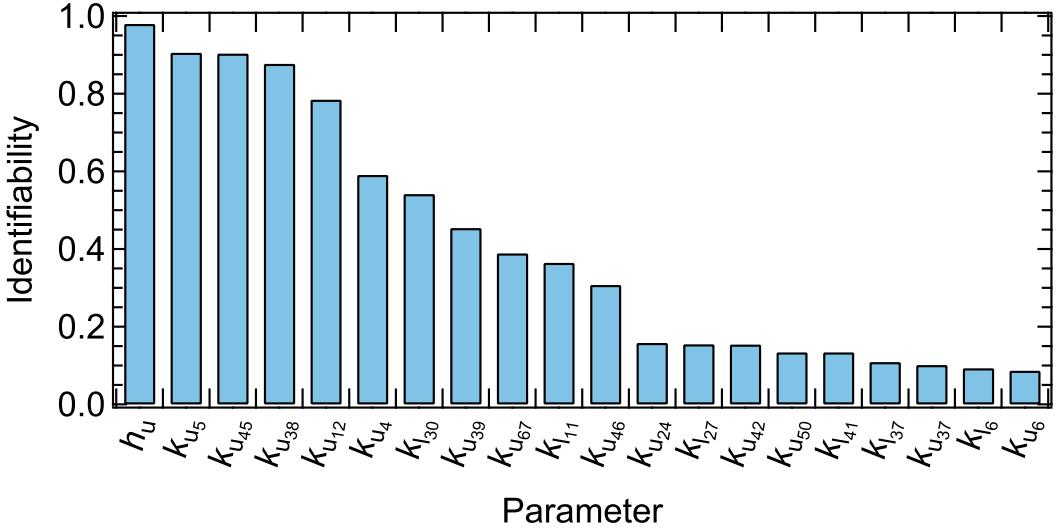


Figure 2.8: The 20 most identifiable parameters.

Initial travel-time variances were calculated by running the 1,000 uncalibrated hydraulic-conductivity-field realizations and these uncertainties were reduced by up to 92% (i.e., standard deviation in travel times reduced from 3.15 to 0.25 years) when the model was run with the calibrated hydraulic-conductivity-field realizations (discussed below in the NSMC section). Later, an observation-worth analysis was performed to assess the contribution of monitoring wells toward reducing travel-time uncertainties. Normalized contributions to uncertainty reduction from each monitoring well are indicated in Figure 2.9(a). The contributions of observations toward decreasing travel-time uncertainties depended upon the degree to which parameters were constrained along the particle paths and the proximity of the observations to the particle paths. Monitoring wells in the SCR reduced uncertainties for particles released at M and L while one monitoring well in the CR reduced uncertainties for particles released at H and one monitoring well in the ER reduced uncertainties for particles from S. Particles released at G did not travel through a region with monitoring wells, so their uncertainties were not reduced. Overall, only four monitoring wells (large open circles in Figure 2.9(a)) significantly contributed to decreasing travel-time uncertain-

ties indicating that the existing monitoring well network does not effectively inform contaminant transport times.

Additional targeted well installations would greatly reduce travel-time uncertainties. To this end, an observation-worth analysis was performed by adding 11 hypothetical monitoring wells (Figure 2.9(b)). The analysis was conducted by placing wells at every third model cell throughout the model domain and selecting optimal locations. The 11 hypothetical wells yielding the greatest reductions in uncertainty were “installed” down gradient of particles released at administrative area M and H and in the vicinity of hazardous-materials storage sites L, G, and S where no wells exist. The normalized contributions of monitoring plus hypothetical wells in reducing travel-time uncertainties are indicated in Figure 2.9(b). Because the 11 hypothetical wells were optimally located, they significantly reduced travel-time uncertainties. Interestingly, upon adding the 11 hypothetical wells, the contribution of an existing well toward reducing travel-time uncertainty greatly increased (largest red circle) because it gained important gradient information.

Post-calibration sensitivities of travel-time predictions to observations were calculated and the degree to which observations reduced uncertainties in travel times are indicated in Figure 2.9(c). M_u and M_l had six nearby monitoring wells and these particles traveled through a well-constrained region of the model and happened to have the shortest travel distances; hence uncertainties were notably reduced by the existing wells. The addition of the 11 hypothetical wells further constrained parameters (and corresponding predictions) such that travel-time uncertainties for particles released at H, G, L, and S decreased by factors from 1.04 to 4.3 (red bars) compared to the existing well network.

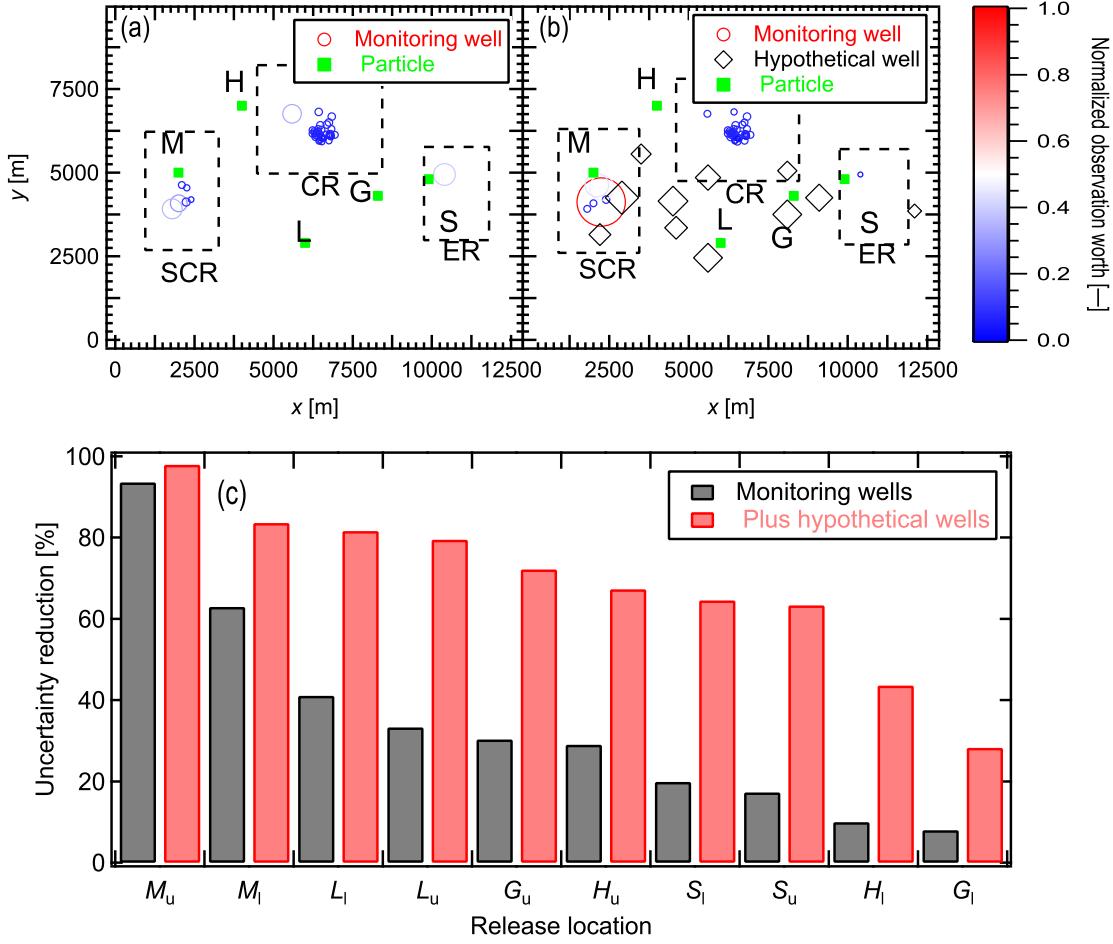


Figure 2.9: Normalized observation worth of (a) monitoring wells (b) plus 11 hypothetical wells, symbol color and size indicate the observation worth; and (c) uncertainty reductions in travel times due to the existing wells (gray) and as augmented by 11 hypothetical wells (red).

NSMC

Of the 1,000 parameter-field realizations generated, 882 had objective functions less than 1.5 times the calibrated (minimum) objective function after three optimization iterations of a NSMC calibration. These calibration-constrained NSMC realizations and the corresponding 882 realizations from RANDPAR were used to compute parameter variances to assess the decreases in uncertainty due to the information content in the calibration data set. Figure 2.10 compares variances of pre-calibration

and calibration-constrained parameter distributions. NSMC reduced uncertainties, at least slightly, for the majority of parameters near existing monitoring wells. For both layers, NSMC reduced notably uncertainties for highly and moderately identifiable parameters (see Figure 2.10 (a)–(b)).

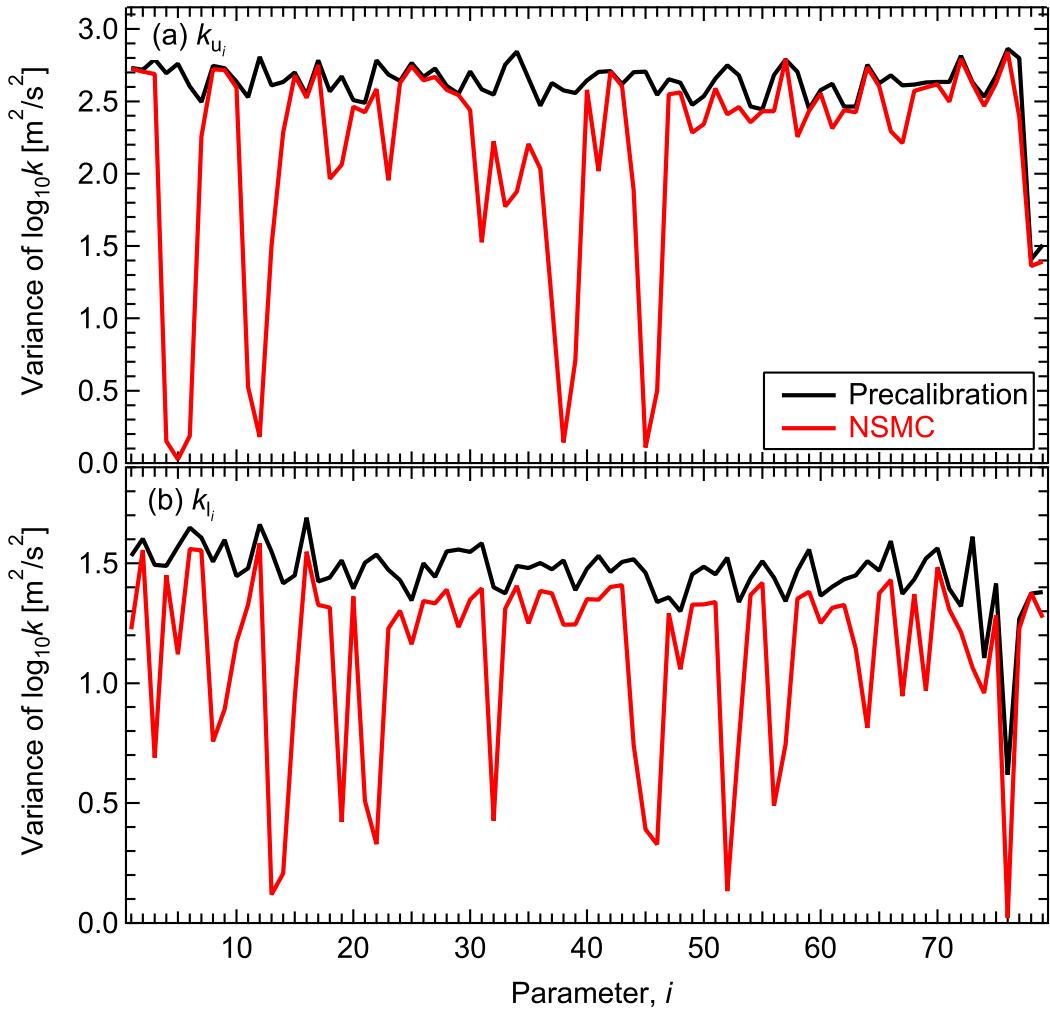


Figure 2.10: Variances of pilot point, $\log_{10}(k)$, for the (a) upper and (b) lower layers for the pre-calibration (black) and NSMC (red) parameter realizations.

Parameter realizations were also used to generate PDFs and Figure 2.11 provides six examples. For highly identifiable parameters (e.g., k_{u_5} and $k_{u_{45}}$) in the top row of Figure 2.11, distributions were significantly narrower for NSMC than from pre-calibration reflecting the information obtained from the null-space projection

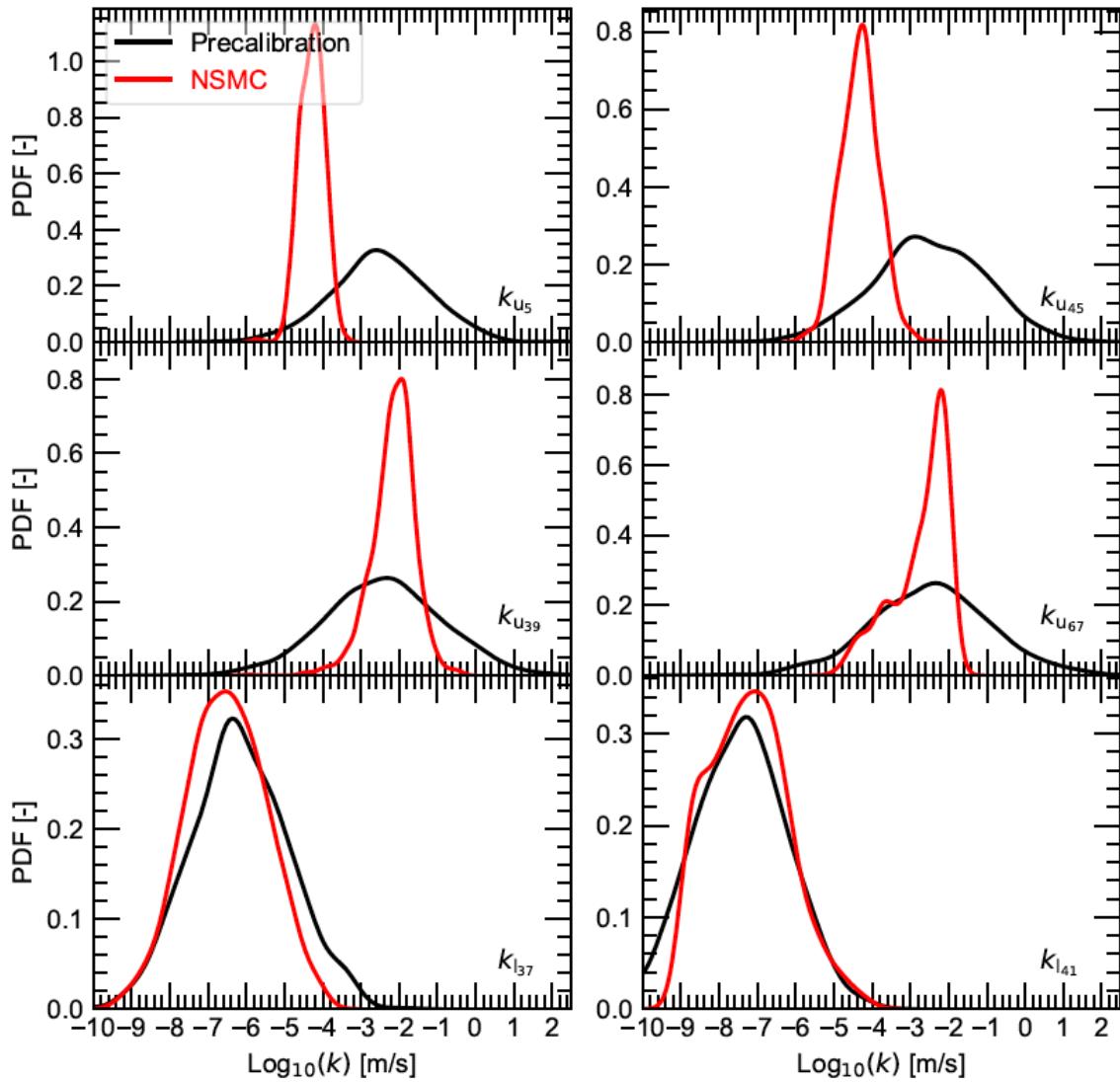


Figure 2.11: PDFs of highly identifiable parameters k_{u5} and k_{u45} (top row), moderately identifiable parameters k_{u39} and k_{u67} (middle row), and minimally identifiable parameters k_{l37} and k_{l41} (bottom row).

and three additional optimization iterations. Distributions of the moderately identifiable parameters (e.g., k_{u39} and k_{u67}) in the middle row of Figure 2.11 were also narrower than their pre-calibration equivalents. Finally, the three optimization iterations slightly reduced uncertainties for minimally identifiable parameters (e.g., k_{l37} and k_{l41}) in the bottom row of Figure 2.11, because those parameters were not informed by existing monitoring wells.

The field data from the monitoring wells were collected by a third party and the model was created after the investigation was finished to try to show how contaminates may have migrated off-site and to demonstrate the need to incorporate the NSMC technique earlier in the monitoring process. There were no opportunities to collect additional data nor drill new wells.

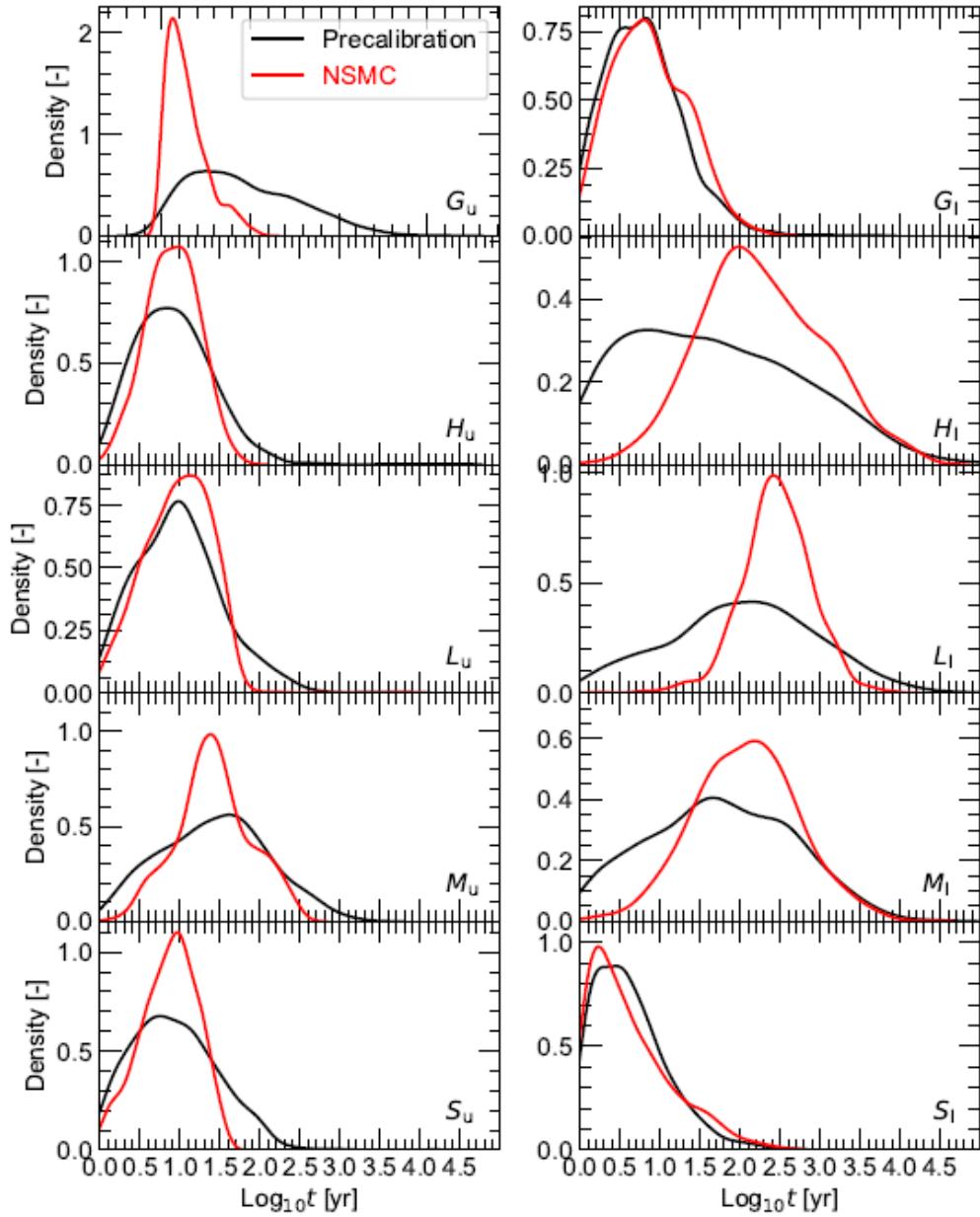


Figure 2.12: PDFs of particle travel times.

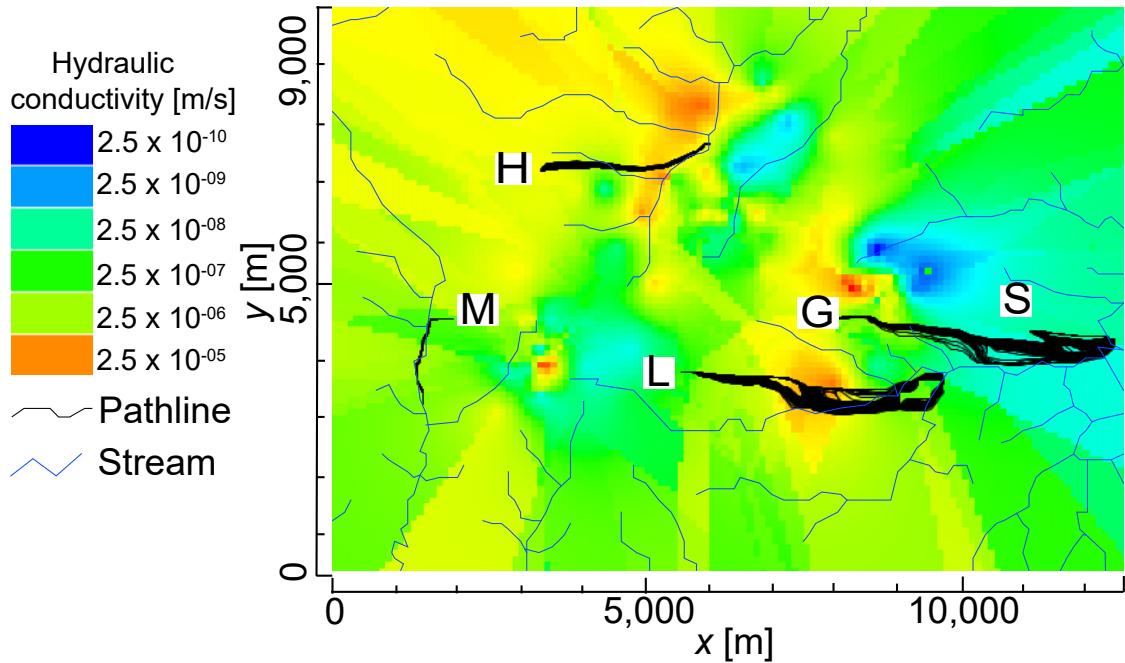


Figure 2.13: Particle paths for 250 realizations with background color a representative hydraulic conductivity distribution.

Travel times for particles to reach exit points at surrounding streams were calculated for each of the 882 pre-calibration and NSMC calibration-constrained pilot-point parameter fields and Figure 2.12 shows distributions of log-transformed travel times. Figure 2.13 shows 250 of 889 particle tracks released from each administrative designation with a representative hydraulic conductivity field. A particle's travel time depended on distance traveled and hydraulic conductivities along its path. Passing through even a single low-hydraulic-conductivity cells significantly decreased that particle's travel time. The combination of one or more low-hydraulic conductivities along a particle's path in conjunction with lognormally distributed hydraulic conductivities (long tails toward low values) yielded some realizations with exceptionally long travel times, hence it was more appropriate to compare median travel times (Table 2.1).

Particles released at the same location for pre-calibration and NSMC parameter realizations had similar path lengths in both layers, but median travel times in the upper layer were significantly shorter than those through the lower layer. Consistent

with the conceptual model, hydraulic conductivity ranges were two to six order of magnitude higher in the upper layer than the lower layer resulting in the travel-time disparities. However, all simulations indicated that particles released in the upper layer reached surrounding streams within a year (consistent with site observations) while particles released in the lower layer took one to five orders of magnitude longer.

Table 2.1: Median travel times for the 882 pre-calibration and NSMC parameter fields used for particle tracking.

Particle	Pre-calibration [yr]	NSMC [yr]
G_u	0.29	\uparrow 0.65
G_l	5.51	\uparrow 6.65
H_u	0.14	\downarrow 0.12
H_l	39.01	\uparrow 166.34
L_u	0.11	= 0.11
L_l	109.92	\uparrow 303.23
M_u	0.03	\uparrow 0.04
M_l	54.67	\uparrow 122.98
S_u	0.13	\downarrow 0.12
S_l	0.96	\uparrow 2.36

Although travel times through the lower layer were longer when using pre-calibration parameter realizations than their NSMC counterparts, there was no consistent trend for the upper layer. Nevertheless, corresponding travel-time uncertainties (variances of log-transformed travel times) always decreased from pre-calibration to NSMC (visually evident in Figure 2.12), which is consistent with the decreases in parameter variances (see Figure 2.10). Although the NSMC travel times through the lower layer indicated that there may be more time to remediate the contamination than might have been initially expected, the short travel times through the upper layer offered no such consolation.

Conclusions

Based on MODFLOW and MODPATH simulations of the NWIRP site, calibration and model interrogation quantified: parameter and predictive uncertainties, parameter identifiabilities, and observation worth for both existing and hypothetical monitoring wells. Using a linear analysis, pre-calibration parameter uncertainties were reduced up to 92% and 36 of 156 parameters exceeded a 10% reduction when constrained by the calibration data set. An identifiability study revealed that seven parameters were highly identifiable (>0.5) while 12 parameters had identifiabilities between 0.1 and 0.5. Travel-time uncertainties were reduced up to 92%. Using a non-linear analysis, pre-calibration travel-time uncertainties were reduced by $>50\%$ for two particles released at site M and between 5 and 40% for the remaining sites. An observation-worth analysis showed that the existing monitoring well network does not strongly constrain travel times. Targeted data collection, especially at the locations shown in Figure 2.5, could reduce travel-time uncertainties for all particles by factors from 1.04 to 4.3.

This study generated pre- and post-calibration parameter distributions along with corresponding travel-time PDFs. The decreases in width of these distributions (variances) reflected the information content in the calibration data set. This study also predicted travel times for conservative tracers to reach nearby streams and revealed that conservative tracers exited the flow system through the more conductive upper layer within a year.

Any seriously contaminated site like NWIRP should undergo a rapid and detailed modeling study before further data collection and, of course, before making remediation decisions. For example, the authority collected clustered water-level measurements, which could have been optimized if a comprehensive study was conducted before drilling wells. In addition, no base-flow data were collected even though a single base-flow measurement significantly improves uncertainty quantification (Hunt,

Fienen, & White, 2019). This study also demonstrated that if a contaminant reached the upper layer (for example during storm events that raise the water table), it will travel much faster to surrounding streams. The predictive-uncertainty and observation-worth analyses determined the most important parameters and observations contributing to the greatest decreases in predicted travel-time uncertainties. Ultimately, the 43 poorly distributed water-level measurements over such a large model domain and the absence of base-flow data were notable shortcomings. This analysis can support decision making by identifying where additional wells should be located to achieve the greatest reductions in predictive uncertainty.

Looking to the future, transient modeling would be appropriate for this system, but it could not be undertaken because of a lack of water-level time-series data. Although beyond the scope of this study, it would be reasonable to use these calibrated parameters in a transient model to assess other aspects of contaminant transport subject to storm or flood events.

CHAPTER THREE

Manuscript 2: PFLOTRAN-SIP

Abstract

Spectral induced polarization (SIP) is a non-intrusive geophysical method that is widely used to detect sulfide minerals, clay minerals, metallic objects, municipal wastes, hydrocarbons, and salinity intrusion. However, SIP is a static method that cannot measure the dynamics of flow and solute/species transport in the subsurface. To capture these dynamics, the data collected with the SIP technique need to be coupled with fluid flow and reactive-transport models. To my knowledge, currently, there is no simulator in the open-source literature that couples fluid flow, solute transport, and SIP process models to analyze geoelectrical signatures in a large-scale system. A massively parallel simulation framework (PFLOTRAN-SIP) was built to couple SIP data to fluid flow and solute transport processes. This framework, PFLOTRAN-E4D, couples PFLOTRAN (a massively parallel multi-physics simulator for subsurface flow and transport) and E4D (a massively parallel geoelectrical simulator), without sacrificing computational performance. PFLOTRAN solves the coupled flow and solute transport process models to estimate solute concentrations, which were used in Archie's model to compute bulk electrical conductivities at near-zero frequency. These bulk electrical conductivities were modified using the Cole-Cole model to account for frequency dependence. Using the estimated frequency-dependent bulk conductivities, E4D simulated the real and complex electrical potential signals for selected frequencies for SIP. The PFLOTRAN-SIP framework was demonstrated through a synthetic tracer-transport model simulating tracer concentration and electrical impedances for four frequencies. Later, SIP inversion estimated bulk electrical conductivities by matching electrical impedances for each specified frequency. The estimated bulk electrical conductivi-

ties were consistent with the simulated tracer concentrations from the PFLOTRAN-SIP forward model. This framework is useful for practitioners of environmental hydrogeophysics and biogeophysics to monitor chemical, nuclear, and tracer transport sites as well as to detect sulfide minerals, metallic objects, municipal wastes, hydrocarbons, and salinity intrusion.

Introduction

Engineered subsurface systems are dynamic due to natural and anthropogenic activities that alter porosity, permeability, fluid saturation, and geochemical properties over time (Council, 2000). Geophysical techniques such as seismic (deep or near-surface seismic) and potential-based methods (electromagnetic, magnetic, electrical resistivity tomography [ERT], spectral induced polarization [SIP]) characterize changes in the subsurface (Kearey et al., 2013; Revil et al., 2011; Snieder et al., 2007). Among these, ERT and SIP map the distribution of bulk electrical conductivity (i.e., the reciprocal of resistivity) due to changes in subsurface fluid flow, temperature, deformation, and reactive transport (Byrdina et al., 2012; Carpenter, 2017; Gresse et al., 2017; Kaselow & Shapiro, 2004; Robinson et al., 2015). Because structural, topological, and geochemical properties (e.g., pore structures, fracture networks, electron donor, etc.) influence bulk electrical conductivity (Revil et al., 2011; Snieder et al., 2007), ERT and SIP are applied in environmental and energy industries to characterize subsurface interactions. Hence, coupling ERT and/or SIP process models to flow and reactive-transport process models can enhance the interrogation of engineered subsurface systems.

ERT's data-collection component measures the electric potentials resulting from an applied direct current (DC), while the data-processing component inverts these measured potentials to map the spatial distribution of bulk electrical conductivity (Byrdina et al., 2012; Revil et al., 2011; Revil, Finizola, Sortino, & Ripepe, 2004).

Because ERT injects DC (near-zero frequency), it cannot interrogate the polarization features of geologic materials, heavy metals, and induced-polarization minerals (e.g., clay minerals, hydrothermal-alteration products, pyrite, finely disseminated sulfide minerals, etc.) (He, Jiang, Liu, & Cui, 2005; Revil et al., 2011; Yan, Xiang, Li, Liu, & Wang, 2014). However, by injecting alternating currents (AC), the induced polarization (IP) method can measure “chargeability” in the time domain or “phase shift” in the frequency domain, which is the phase angle (phase lag) between the applied current and induced voltage of polarized geologic materials (Sumner, 1976; Vaudelet, Revil, Schmutz, Franceschi, & Bègassat, 2011). The IP method measures the energy storage capacity of certain minerals and can be used to detect hydrocarbons (Luo & Zhang, 1998), contaminant plumes (Morgan et al., 1999; Olhoeft, 1986; Vanhala, 1997), municipal waste, green waste (agricultural and biodegradable wastes) (Aristodemou & Thomas-Betts, 2000), sulfide minerals (Butler, 2005; Yan et al., 2014), and hydrothermal products (Butler, 2005; Yan et al., 2014). IP is a single- or double-frequency method that generally fails to distinguish between a true IP response (e.g., polarized geologic materials) and noise (e.g., electromagnetic interference) (Butler, 2005; Luo & Zhang, 1998). The SIP method extends IP to measure geoelectrical signatures across a range of user-specified frequencies to facilitate comprehensive data collection to identify sources of a true IP response. It also has the potential to characterize subsurface structures and processes (Luo & Zhang, 1998).

SIP is a comprehensive but static method to extract subsurface polarization signals. It cannot measure evolving dynamics of subsurface processes (e.g., flow, deformation, reactive transport, etc.) (Johnson et al., 2017). Moreover, it cannot detect all contaminants or chemical reactions occurring in a system (Vaudelet, Revil, Schmutz, Franceschi, & Bègassat, 2011). However, these shortcomings can be overcome by coupling the SIP method with flow and reactive-transport models. Specifically, the process models related to fluid flow and reactive-transport simulate the spatio-temporal

distribution of conducting fluids in rocks, fluid content in the pores, and fluid chemistry. The concentration of chemical species is transferred to the SIP method, which simulates electrical potentials due to polarization. The electrical conductivity from the SIP method contains information on the spatial distribution of conducting fluids and fluid chemistry. In addition, the SIP method inverts for frequency-dependent electrical conductivity based on measured/simulated electrical-impedance and phase-shift data, which facilitates detection, extraction, and understanding of the evolution of hydrogeophysical and biogeophysical signatures at both lab and field scales (Atekwana & Slater, 2009; Kenma, Vandeborught, Kulessa, & Vereecken, 2002; Mellage et al., 2018; Slater et al., 2002).

Software to model and invert geoelectrical data include: `Res2DInv` (Loke, 2019; Loke, Acworth, & Dahlin, 2003; Loke & Barker, 1996), `AarhusInv` (Fiandaca, Ramm, Binley, Christiansen, & Auken, 2013), `BERT` (Günther, Rücker, & Spitzer, 2006; Rücker, Günther, & Spitzer, 2006), `EarthImager3D` (EarthImager 3D, 2008), `E4D` (Johnson, Versteeg, Ward, Day-Lewis, & Revil, 2010), `pyGIMLi` (Rücker, Günther, & Wagner, 2017), and `ZondRes3D` (ZONDRES3D, 2017). These software packages can also image frequency-dependent electrical conductivities but cannot capture dynamic subsurface processes. Moreover, when imaging fluid flow in the subsurface, they are not coupled with flow and reactive-transport models, which would improve image quality. To overcome these problems, Johnson et al. (Johnson et al., 2017) developed the massively parallel `PFLOTTRAN-E4D` simulator, which couples `PFLOTTRAN` (Hammond, Lichtner, & Mills, 2014), a subsurface flow and reactive-transport code, to `E4D`, a finite element code for simulating and inverting geoelectrical data. However, `PFLOTTRAN-E4D` does not account for induced polarization. To capture dynamics of subsurface processes and the true sources of induced polarization, a computationally efficient framework is needed to couple fluid flow and solute transport with the SIP process model. *This work extended the capabilities of PFLOTTRAN-E4D to include SIP in a framework called*

PFLOTTRAN-SIP. Here, in a medium with polarization properties, PFLOTTRAN-SIP was demonstrated with a representative tracer-transport model to provide more information than the ERT model.

PFLOTTRAN-SIP: Process Models and Coupling Framework

The PFLOTTRAN-SIP framework couples flow and reactive-transport process models in PFLOTTRAN (Hammond, Lichtner, Lu, & Mills, 2012; Hammond et al., 2014; Lichtner, Hammond, et al., 2019a, 2019b) with SIP process models in E4D (Johnson et al., 2010; Pacific Northwest National Laboratories, 2014; T. C. Johnson, 2014) to characterize fluid-driven electrical impedance signatures across multiple frequencies. At each time-step, simulation outputs from PFLOTTRAN (fluid saturation, tracer concentration, etc.) were supplied to Archie's model (Archie, 1942) to calculate fluid-dependent bulk electrical conductivities for E4D simulations. These estimated bulk electrical conductivities were decomposed into real and imaginary components for user-defined frequencies using the Cole-Cole model (Cole & Cole, 1941; Tarasov & Titov, 2013), which is an empirical description of frequency-dependent behavior of bulk electrical conductivities. These processes were repeated until the entire transient simulation was completed.

E4D is an open-source, massively parallel, finite-element code for simulating and inverting three-dimensional time-lapsed ERT and SIP data (Johnson & Thomle, 2017; Johnson et al., 2010; Pacific Northwest National Laboratories, 2014; T. C. Johnson, 2014). The process models in E4D for ERT and SIP assume that displacement currents are negligible and current density can be described by Ohm's constitutive model (Johnson & Thomle, 2017). These assumptions result in a Poisson equation relating induced current to the potential field that determines the electrical potential field:

$$-\operatorname{div} [\sigma(\mathbf{x}) \operatorname{grad} [\Phi_\sigma(\mathbf{x})]] = \mathcal{I}\delta(\mathbf{x} - \mathbf{x}_0), \quad (3.1)$$

where σ [S m^{-1}] is the effective electrical conductivity, \mathcal{I} [A] the current injected, and $\Phi_\sigma(\mathbf{x})$ [V] the electrical potential all at position-vector \mathbf{x} [m] while $\delta(\cdot)$ is the Dirac delta function.

Equation (3.1) models the DC effect, which is required in ERT forward/inverse modeling; however, it does not account for induced polarization under alternating currents. Induced polarization under alternating current results in a secondary potential that needs to be accounted for in the SIP forward/inverse modeling. This requires modification of Equation (3.1) to solve for the total electrical potential field under IP effects:

$$-\operatorname{div}[(1 - \eta(\mathbf{x}))\sigma(\mathbf{x})\operatorname{grad}[\Phi_\eta(\mathbf{x})]] = \mathcal{I}\delta(\mathbf{x} - \mathbf{x}_0), \quad (3.2)$$

where Φ_η [V] is the total electrical potential field, which includes IP effects from a polarized material with chargeability distribution $\eta(\mathbf{r})$ [milliradians] (H. O. Seigel, 1959). The secondary potential resulting from the IP effect is (Oldenburg & Li, 1994):

$$\Phi_s = \Phi_\eta - \Phi_\sigma, \quad (3.3)$$

and the apparent chargeability is (H. O. Seigel, 1959):

$$\eta_a = \frac{\Phi_\eta - \Phi_\sigma}{\Phi_\eta}. \quad (3.4)$$

Secondary potential Φ_s and apparent chargeability η_a can be computed by solving Eqs. (3.1) and (3.2). These potentials Φ_η , Φ_σ , and Φ_s are time-domain signatures of induced polarization. Eq. (3.3) is in the time domain and is transformed into the frequency domain:

$$-\operatorname{div}[\sigma^*(\mathbf{x}, \omega)\operatorname{grad}[\Phi^*(\mathbf{x})]] = \mathcal{I}\delta(\mathbf{x} - \mathbf{x}_0), \quad (3.5)$$

where ω [Hz] is the frequency. $\sigma^*(\mathbf{x}, \omega)$ [S/m] and $\Phi^*(\mathbf{x})$ [V] are the frequency-dependent electrical conductivities and electrical potential, respectively. $\Phi^*(\mathbf{x})$ consists of real and complex electrical potentials corresponding to induced polarization.

Zero potential is enforced on boundaries of the domain (Johnson & Thomle, 2017, Section 3) to solve Eq. (3.5).

E4D simulates four-electrode configurations (e.g., Wenner array, dipole-dipole array) (Johnson et al., 2010). Current is injected from source to sink electrodes while measurements are recorded between the other two electrodes (Johnson et al., 2010; Kearey et al., 2013; Robinson et al., 2015). For ERT, the measured response is the potential difference (Voltage) between the two electrodes while SIP also includes the phase shift (radians). Based on the user-defined survey design, E4D simulates up to thousands of ERT/SIP measurements to compute the electrical potential distribution or to invert the lab/field/simulated data to identify the best-fit bulk electrical conductivity distribution throughout the model domain. As the governing equations are linear in Φ_σ and Φ_η , E4D solves Eq. (3.5) by superimposing pole solutions with different current sources that makes ERT or SIP forward modeling highly scalable (Johnson & Thomle, 2017; Johnson et al., 2010).

E4D solves the ERT and SIP process models in the frequency domain using a low-order finite element method (FEM). The output of the FEM solution for the ERT process model is electrical potential throughout the domain, which is real valued and frequency independent. Because the SIP process model is frequency dependent, the corresponding output of the FEM solution has both real and imaginary components of electrical potential. The complex-valued electrical potential (or equivalently the phase-shift distribution in the model domain) provides new information on IP in the subsurface, which is not capturable by ERT.

E4D uses the standard Galerkin weak formulation (T. J. R. Hughes, 2012) on an unstructured, low-order, tetrahedral, finite element mesh (Si, 2015) and it iteratively computes the total electrical potential field due to IP effects (Johnson & Thomle, 2017, Section 3). Equations for computing the real and imaginary components of the complex-valued electrical potential are decoupled, and the finite-element analysis is

performed in the real-number domain. First, E4D solves for the real component without considering IP effects. Second, the current source for the imaginary component is computed from the real component. Third, the imaginary component of the total electrical potential is calculated based on this computed current source. Fourth, the secondary current source arising from the imaginary component is computed. This secondary current source considers IP effects. Later, the real component is calculated based on this secondary current source. These steps are repeated until a convergence criterion is satisfied.

PFLOTRAN Process Models

PFLOTRAN solves a system of nonlinear partial differential equations describing multiphase, multicomponent reactive flow and transport using the finite-volume method (FVM) (Hammond et al., 2012, 2014; Lichtner et al., 2015). In this paper, I consider only single-phase fluid flow and solute transport when predicting the spatio-temporal distribution of solute concentrations. Mass conservation for single-phase, variably saturated flow is:

$$\frac{\partial \phi s \rho}{\partial t} + \text{div} [\rho \mathbf{q}] = Q_w, \quad (3.6)$$

where ρ [kg/m³] is the fluid density, ϕ [-] is the porosity, s [-] is the saturation, t [s] is time, \mathbf{q} [m/s] is the Darcy flux, and Q_w [kg/m³/s] is the volumetric source/sink term. Darcy flux is:

$$\mathbf{q} = -\frac{k k_r(s)}{\mu} \text{grad} [p - \rho g z], \quad (3.7)$$

where k [m²] is the intrinsic permeability, k_r [-] is the relative permeability, μ [pas] is dynamic viscosity, p [Pa] is pressure, g [m/s] is gravity, and z [m] is the vertical component of \mathbf{x} . The source/sink term is:

$$Q_w = \frac{q_M}{W_w} \delta (\mathbf{x} - \mathbf{x}_Q), \quad (3.8)$$

where q_M [kg/m³/s] is the mass flow rate, W_w [kg/kmol] is the formula weight of water, and \mathbf{x}_Q [m] denotes the location of the source/sink. The governing equation

for tracer transport is:

$$\frac{\partial \phi c}{\partial t} + \operatorname{div} [c \mathbf{q} - \phi s \tau D \operatorname{grad} []c] = Q_c, \quad (3.9)$$

where c [molality] is the solute concentration, D [m^2/s] is the diffusion/dispersion coefficient, τ [–] is tortuosity (related to the path length of the fluid flow), and Q_c [molality/s] is the solute source/sink term. Dirichlet, Neumann, or Robin boundary conditions are specified when solving Eqs. (3.6)–(3.9).

Coupled governing Eqs. (3.6)–(3.9) are solved with a two-point flux FVM in space and a fully implicit backward Euler method in time using a Newton–Krylov solver (Balay et al., 2017; Hammond et al., 2012). PFLOTRAN’s process model tree shown in Fig. 3.1 has master process \mathcal{A} and pointers to child process \mathcal{B} and peer process \mathcal{C} . Here, the flow model is master process \mathcal{A} while \mathcal{B} and \mathcal{C} are the solute transport and E4D/SIP models, respectively. The time step for the flow model may be different from solute-transport model. Transfer of information between \mathcal{A} (e.g., flow) and \mathcal{B} (e.g., solute transport) takes place before and after each of \mathcal{A} ’s time steps. Synchronization of \mathcal{A} and \mathcal{C} (e.g., ERT or SIP) occurs at specified times. Execution starts with the master-process model \mathcal{A} , which can take as many adaptive time steps as needed to reach the synchronization point. \mathcal{B} and \mathcal{C} proceed according to their time steps ($\leq \mathcal{A}$ ’s) to reach the synchronization point. When \mathcal{A} , \mathcal{B} , and \mathcal{C} all reach the synchronization point, variables and parameters (e.g., saturation, solute concentration, porosity, etc.) are updated between \mathcal{A} and \mathcal{C} .

PFLOTRAN-SIP Coupling

The coupling involves six steps: (1) PFLOTRAN’s flow model calculates fluid pressure, saturation, and velocity; (2) using those simulated outputs, the transport model calculates solute concentrations; (3) solute concentrations in each PFLOTRAN mesh cell are used to calculate DC electrical conductivities for E4D based on Archie’s model; (4) the Cole-Cole model calculates frequency-dependent electrical conductivi-

ties; (5) real and complex electrical conductivities are interpolated onto the E4D mesh; and (6) the SIP model solves the forward problem to calculate electrical impedances and phase shifts.

PFLOTRAN and E4D use message passing interface calls for inter-process communication. Based on user specification, PFLOTRAN divides the computing resources between PFLOTRAN and E4D at the initial step. PFLOTRAN and E4D read their corresponding input files and complete pre-simulation steps. These include setup of the flow model, the solute transport model, the SIP model, and mesh interpolation matrix. Mesh interpolation is needed for two reasons: (1) the meshes of PFLOTRAN and E4D are different and (2) the solution procedure of PFLOTRAN is based on the FVM while E4D's solution procedure is based on the FVM. As a result, the state variables (e.g., solute concentration, fluid saturation) computed at the cell center by PFLOTRAN need to be accurately transferred from the PFLOTRAN mesh to the E4D mesh to calculate electrical conductivities. Generation of the mesh interpolation matrix is described in Sec. 3. Algorithm 3 and Fig. 3.2 summarize the coupling of PFLOTRAN and E4D SIP models.

Petrophysical Transformation

To simulate SIP signals during fluid flow and solute transport, a mathematical relationship linking fluid flow state variables and bulk electrical conductivities is required. Archie's model (Archie, 1942; Glover, 2010; Shah & Singh, 2005) is a petrophysical transformation relating state variables simulated by PFLOTRAN to bulk electrical conductivities:

$$\sigma_b(\mathbf{x}) = \frac{1}{\tau} \phi^\alpha s_f^\beta \sigma_f, \quad (3.10)$$

where $\sigma_b(\mathbf{x})$ [S/m] is the bulk electrical conductivity at near-zero frequency ($\omega \sim 0$), α [-] is the cementation exponent (1.8 to 2.0 for sandstone), s_f [-] is the solute

concentration simulated by PFLOTTRAN, β [–] is the saturation exponent (close to 2.0), and σ_f [S/m] is the fluid electrical conductivity.

To account for frequency dependence, Eq. (3.10) was modified using the Cole-Cole model (Cole & Cole, 1941, 1942; Dias, 2000; Revil, Florsch, & Camerlynck, 2014; Tarasov & Titov, 2013):

$$\sigma^*(\mathbf{x}, \omega) = \sigma_b(\mathbf{x}) \left[1 + \eta_a \left(\frac{(i\omega\mathcal{T})^\gamma}{1 + (1 - \eta_a)(i\omega\mathcal{T})^\gamma} \right) \right], \quad (3.11)$$

where $i^2 = -1$, γ [–] is a shape parameter, and \mathcal{T} [s] is the characteristic relaxation time constant (time for the imaginary electrical component to reach equilibrium after perturbation) related to characteristic pore or grain size.

Mesh Interpolation

Once the frequency-dependent real and imaginary components of bulk electrical conductivities were calculated on PFLOTTRAN mesh, they were interpolated onto the E4D mesh. The conductivity at any intermediate point in a PFLOTTRAN mesh cell was approximated using tri-linear interpolation. These approximated values were computed using values at the PFLOTTRAN cell centers surrounding the point (see Fig. 3.3) using (Johnson et al., 2017):

$$\sigma_i^c(\mathbf{x}, \omega) = \frac{1}{V_i} \int_{v_i} \sigma^c(\mathbf{x}, \omega) dV \approx \frac{1}{n_k} \sum_{k=1}^{n_k} \sigma_{i,k}^c(\mathbf{x}, \omega), \quad (3.12)$$

where V_i [m^3] is the volume of the i^{th} element of E4D mesh, σ^c [$S m^{-1}$] is the bulk electrical conductivity in the PFLOTTRAN mesh element, n_k [–] is the number of subdivisions by which i^{th} E4D element is divided for integral approximation, and $\sigma_{i,k}^c$ [$S m^{-1}$] is the bulk conductivity within the i^{th} element and subdivision k . The value of $\sigma_{i,k}^c$ is:

$$\sigma_{i,k}^c(\mathbf{x}, \omega) = \sum_{j=1}^{n_c} W_{i,k,j} \sigma_j^c(\mathbf{x}, \omega), \quad (3.13)$$

where n_c [–] is the total number of mesh elements in the PFLOTTRAN mesh, σ_j^c is the bulk conductivity of the j^{th} element in the PFLOTTRAN mesh, and $W_{i,k,j}$ is the linearly

interpolated weight for σ_j^c to determine the value of sub-element k in the i^{th} E4D mesh element.

The preceding equations interpolated data onto the E4D mesh based on the computed values of $\sigma(\mathbf{x}, \omega)$ in the PFLOTTRAN mesh (Johnson et al., 2017; Johnson & Thomle, 2017):

$$\sigma_i^e = \frac{1}{n_k} \sum_{k=1}^{n_k} \sum_{j=1}^{n_c} W_{i,k,j} \sigma_j^c. \quad (3.14)$$

Numerical Model Setup

PFLOTTRAN Model Setup

The domain was $500 \times 500 \times 500 \text{ m}^3$ and consisted of three layers as shown in Fig. 3.4. The PFLOTTRAN mesh had a total of 125,000 finite volume cells of equal size. The upper layer was $500 \times 500 \times 350 \text{ m}^3$ and extended from $z = 0$ to -350 m as a highly conductive material with a permeability of $7.38 \times 10^{-13} \text{ m}^2$. Fluid was water, and these rock properties (e.g., permeability, porosity, diffusion coefficient, etc.) are representative of sandstone. The middle layer was less permeable (permeability was $1.05 \times 10^{-22} \text{ m}^2$), with size $500 \times 500 \times 50 \text{ m}^3$ extending from $z = -350$ to -400 m . This permeability is representative of rocks such as shale or granite. The low-permeability layer, however, included a small-volume, highly permeable ($7.38 \times 10^{-13} \text{ m}^2$) material between $x = 300$ and 350 m , $y = 0$ and 500 m , and $z = -400$ and -450 m . The bottom layer was also a highly conductive region with a permeability of $7.38 \times 10^{-13} \text{ m}^2$ and dimensions of $500 \times 500 \times 100 \text{ m}^3$.

A solute (conservative tracer) at 10 mol m^{-3} was placed below the low permeable zone as shown in Fig. 3.4 as the purple $50 \times 500 \times 50 \text{ m}^3$ block. The initial and boundary conditions for the model included: pressure of 1 atm at the top with a hydrostatic pressure gradient from top to bottom. The right face ($x = 0$) was assigned 2 atm to drive fluid from right to left. For solute transport, the boundary conditions were zero-concentration Dirichlet inflow at the right face and zero diffusive gradient outflow at

the left face that allowed only advective outflow. The remaining faces were specified as zero-solute flux boundaries.

For low- and high-permeability zones, tortuosity was 1.0 while porosities were 0.3 and 0.25, respectively. Solute diffusivity was $10^{-9} \text{ m}^2 \text{ s}^{-1}$. The Newton solver (20-iteration maximum) was applied for flow and solute transport. For the flow solver, relative and absolute tolerances were 10^{-50} with a relative update tolerance of 10^{-60} while for solute transport solver, relative and absolute tolerances were 10^{-4} with a relative update tolerance of 10^{-60} . The simulation was run for one year with an initial time step of 10^{-8} years, which was allowed to accelerate by a factor of 8.

SIP Model Setup

Although the domain dimensions for SIP simulations were identical to the PFLOTTRAN simulation, there was only a single layer. The corresponding E4D mesh for the simulation had 86,780 nodes and 609,562 tetrahedral elements. Zero potentials were enforced on the boundaries. A total of 80 point electrodes were placed in the domain, all located at $z = -425 \text{ m}$ arranged in 5 lines along the x -axis, with each line comprising 16 electrodes. The electrode coordinates started at $(40, 50, -425)$ and ended at $(460, 450, -425)$ with a 28 m separation between lines. Electrode measurement configurations included a combination of Wenner and dipole-dipole arrays.

A current of 1A was injected between one pair of electrodes and the potential difference was measured between another pair of electrodes. There are various advantages of using separate source and potential electrode pairs. For example, such a measurement system can eliminate any inaccuracies caused by the injecting circuit impedance (the contact impedance between the probe and the medium, which can be high). Using the prescribed measurement configuration, a total of 1,062 simulated measurements were collected to capture electrical impedance and phase shift.

The electrical conductivity of the fluid at $\omega = 0$ Hz was 2×10^{-3} S/m. The cementation and saturation exponents were 0.564 and 0.576, respectively. The characteristics-relaxation time was 0.061 s, all representative of sandstone (Titov, Komarov, Tarasov, & Levitski, 2002). SIP analysis was performed for five different frequencies: 0.1, 1.0, 10, 100, and 1,000 Hz. Forward model simulations were performed using 61 processors, where 20 processors were assigned for PFLOTTRAN and 41 for E4D. Out of those 41 processors, 40 performed SIP simulations for different measurement configurations, and the remaining processor gathered the simulated data.

SIP Inversion of Electrical Conductivity

For verification, E4D's inversion module was used to estimate frequency-dependent electrical conductivity based on the simulated electrical impedance and phase-shift data. This estimated conductivity was compared with the simulated conductivity generated by the PFLOTTRAN-SIP framework. The employed inversion process was blind (i.e., I did not provide prior constraints on the conductivity). This can be improved by providing detailed conductivity information to E4D's inversion module. The SIP inversion employs an unstructured mesh, which consisted of 51,124 nodes with 316,183 mesh elements. Low-order mesh elements were generated to make inversion process simple and computationally efficient because high-order mesh elements did not improve the outcome (T. C. Johnson, 2014). The simulated measurements by PFLOTTRAN-SIP were the data supplied to the inversion process as observations.

The E4D inversion is based on minimizing the following objective function to estimate the frequency-dependent electrical conductivity distribution, σ_{est} :

$$\Phi = \Phi_d [\mathbf{W}_d(\Phi_{\text{obs}} - \Phi_{\text{pred}})] + \zeta \Phi_m [\mathbf{W}_m(\sigma_{\text{est}} - \sigma_{\text{ref}})], \quad (3.15)$$

where Φ_d is an operator that provides a scalar measure of the misfit between observed and simulated data (e.g., electrical impedance and phase shift) based on the user-specified norm (e.g., Euclidean norm), Φ_m is another operator that provides the

scalar measure of the difference between σ_{est} [S m^{-1}] and constraints placed upon the structure of σ_{ref} [S m^{-1}], ζ is the regularization parameter, \mathbf{W}_d is the data-weighting matrix, and \mathbf{W}_m is the model-weighting matrix. σ_{est} and σ_{ref} are the estimated and reference frequency-dependent electrical conductivities. The user specified bounds on the frequency-dependent conductivity in each mesh cell were 0.00001 and 1.0. The Φ_{obs} and Φ_{pred} were the observed and simulated data, respectively. Eq. (3.15) is solved using the iteratively reweighted least square method (Scales & Gerszenkorn, 1988). Further details on the parallel inverse modeling algorithm and its implementation in E4D are available (Johnson et al., 2010).

The ζ value was 100 at the beginning of the inversion and decreased as the non-linear iteration progressed. Before ζ was reduced, the minimum fractional decrease in the objective function, Φ , between iterations had to be less than 0.25 upon which ζ was reduced to 0.5. The convergence of the SIP inversion procedure was based on the χ^2 value of the current iteration after data culling, computed as:

$$\chi^2 = \frac{\Phi_d}{n_d - n_c}, \quad (3.16)$$

where the data residual is the difference between observed and estimated values divided by the standard deviation for that measurement. n_d is the total number of survey measurements and n_c is the number of measurements selected from the total number of measurements during the current iteration. SIP inversion converged after 48 iterations when χ^2 reached 60 using the example model from Sec. 3.

Results

The one-year PFLOTTRAN-SIP model simulations were completed in two minutes. The computation was performed on 61 Intel® Xeon® CPU E5-2695 V4 @ 2.1GHz processors. Fig. 3.5 shows the tracer concentration at the end of simulation. In one year, the pressure gradient drove the tracer about 100 m from its initial location in the y -direction and also moved it upward about 20 m.

The SIP module in PFLOTTRAN-E4D simulated real and complex electrical impedances at 0.1, 1, 10, 100, and 1,000 Hz. Because of minimal differences between 1 and 10 Hz, only results for 0.1, 10, 100, and 1,000 Hz are discussed. This indicated that some frequencies may be redundant because they yielded similar impedances. Sensitivity analysis can be performed to identify redundant frequencies; however, this was beyond the scope of this paper. Fig. 3.6 shows the real and complex potentials due to changes in tracer concentration for the various frequencies. Also, this figure provides information on the change in electrical potential at different frequencies for a single measurement configuration, indicating the maximum tracer concentration. This 80-node measurement configuration was selected because tracer concentration was most visible. The response clearly shows the polarization feature of the tracer. The gradient of the real electrical potential was high near $x = 300$ m (top row of Fig. 3.6) because the tracer concentration was maximum. From Fig. 3.6, it is evident that the real potential response for 0.1 Hz is different from the responses at 10, 100, and 1,000 Hz. The root-mean-square error (RMSE) between these responses was approximately 15% of the maximum real potential value indicating that frequency has an impact on the real potential distribution.

The bottom row of Fig. 3.6 shows complex electrical potential responses where the polarity was switched (colors interchanged). Unlike the real electrical potential, each complex electrical potential was notably different suggesting that it was frequency dependent. The corresponding RMSE between responses was $\sim 85\%$ of the maximum complex potential value. Such high variation was expected as the complex electrical potential depends on frequency, chargeability, and relaxation time, although the last two were constant in this study. Because the response of the complex potential was clearly visible in the simulation, this indicated that the PFLOTTRAN-SIP framework can effectively simulate polarized geologic materials.

Fig. 3.7 shows the simulated and estimated frequency-dependent electrical conductivities using the PFLOTTRAN-SIP framework with the SIP inversion module in E4D. The true (PFLOTTRAN simulated) and estimated (inversion of survey data) real electrical conductivities are plotted in Fig. 3.7(a)-(d) and Fig. 3.7(e)-(h), respectively. SIP inversion was performed using the simulated electrical impedance, and phase-shift data obtained from PFLOTTRAN-SIP model runs after one year. The computational time required to perform SIP inversion was approximately two hours on 41 Intel® Xeon®d CPU E5-2695 V4@ 2.1GHz processors. Estimated electrical conductivities showed high contrast around the high tracer distribution/simulated conductivities, although they were more diffuse than the true/simulated distribution (Fig. 3.7(a-h)). However, estimated conductivities at 1,000 Hz were more accurate than frequencies $<1,000$ Hz with the same constraints. Later, real conductivity values were used in Eq. 3.11 to provide initial guesses for complex conductivities for SIP inversion. Estimated complex electrical conductivity distributions are shown in Fig. 3.7(i)-(l). Similar to estimated real conductivities, complex conductivities computed from SIP inversion were also diffuse. The inversion process can be improved by providing prior information and structural constraints on electrical conductivities. However, both estimated conductivities were generally consistent with the tracer distribution, which showed that the SIP inversion module can simulate electrical impedance and phase-shift data. To summarize, SIP provides a major benefit, which ERT lacks. SIP provides greater information content than ERT. This is because the SIP survey yields multiple datasets at different frequencies, which help to overcome false positives. For example, from Fig. 3.7 it is evident that the SIP inversion analyses at different frequencies are indicating the same tracer region, (not a false positive). With an ERT survey, it may be difficult to delineate a false positive from a true positive because ERT generates a single dataset.

Fig. 3.8(a)-(c) show simulated outputs of tracer concentrations, real potentials, and complex potentials for the 80-electrode measurement configuration at frequencies of 0.1, 10, 100, and 1,000Hz. The location of maximum tracer concentration was around $x = 300$ m. The locations of current and potential measurement electrodes were at ($x = 208, 236, 264$, and 292 m, $y = 250$ m, and $z = -420$ m) (Fig. 3.8(a)). Note that electrodes were not placed at the location of maximum concentration because of the lack of a prior knowledge of tracer fate (which is the case in real-world applications). Because the measurement electrodes were offset from the maximum tracer concentration, this resulted in an offset of peaks between tracer distribution and geoelectrical signals. However, the measured potentials provided meaningful information on the bounds of the tracer distribution as well as revealing the significance of higher frequencies obtained from a combination of the electrical impedance and phase shift.

Fig. 3.8 (b) and (c) showed that the absolute real potential and complex potential decreased as frequency increased. For SIP simulations, E4D first solves the real potential Φ_r . That is, $-\text{div} [\sigma_r \text{grad} [\Phi_r]] = I$, where σ_r is the real component of $\sigma^*(\mathbf{x}, \omega)$ and the real potential, Φ_r is inversely proportional to σ_r . Also, σ_r increases as ω increases; hence the absolute value of the real potential distribution (as shown in Fig. 3.8 (b)) decreases as ω increases. After σ_r is evaluated, E4D computed the complex potential by solving $\text{div} [\sigma_r \text{grad} [\Phi_c]] = -\text{div} [\sigma_c \text{grad} [\Phi_r]]$ where σ_c is the imaginary part of $\sigma^*(\mathbf{x}, \omega)$ and Φ_c is the imaginary potential. Thus, Φ_c is proportional to σ_c . Also, σ_c decreases as ω increases; hence the absolute value of the complex potential distribution (as shown in Fig. 3.8 (c)) decreased as ω increased.

Fig. 3.8 (d) shows phase-shift data distribution along the same line for tracer distribution, real and imaginary potential distribution. Mathematically, the phase shift is the inverse tangent of the ratio between complex and real potential responses. Physically, it is the shift between voltage and current signals that is largely governed

by the polarization characteristics of the subsurface. In this study, phase shift leveraged signals from both real and complex potential responses to improve interpretation of survey data. From Fig. 3.8, there was a change in phase shift where tracer transport was predominant. Moreover, the 1,000 Hz frequency bounded the tracer zone better than lower frequencies that cannot be distinguished with ERT. This change in phase helped constrain the polarized region or bound the interface between tracer-filled and tracer-free fluids. Obtaining the region of interest using these constraints, further geoelectrical interrogation can be performed with this volume. Hence, through phase-shift signatures across multiple frequencies, the PFLOTTRAN-SIP framework will facilitate identification of polarized or geochemically altered zones.

Discussion

IP arises from solute transport and accumulation of ions/electrons in polarized materials (e.g., those with different grain types, colloids, biological materials, phase-separated polymers, blends, and crystalline minerals, etc.) that are subject to an external electric field. Five mechanisms govern the IP phenomena at frequencies < 1 MHz: (1) Maxwell-Wagner polarization, which occurs at high frequencies (Alvarez, 1973; Chelidze & Gueguen, 1999; Y. Chen & Or, 2006; Lesmes & Morgan, 2001); (2) polarization of the inner part of the interface between minerals and water (de Lima & Sharma, 1992; Leroy & Revil, 2009; Revil, 2012; Vaudelet, Revil, Schmutz, Franceschi, & Bégassat, 2011); (3) polarization of the outer part of the interface between minerals and water (de Lima & Sharma, 1992; Dukhin, Shilov, & Bikerman, 1974); (4) membrane polarization for multi-phase systems (Marshall & Madden, 1959; Titov et al., 2002; Vinegar & Waxman, 1984); and (5) electrode polarization observed in the presence of disseminated conductive minerals such as sulfide minerals and pyrite (Merriam, 2007; H. Seigel, Nabighian, Parasnis, & Vozoff, 2007; Wong, 1979).

This PFLOTTRAN-SIP simulations were geared toward IP mechanisms (1), (4), and (5). To simulate the mechanisms mentioned in (2) and (3), Eq. (3.11) must be replaced with conductivity models that account for interface polarization with consideration of effective pore size, electrical formation factor, distribution of relaxation times, and sorption mechanisms (Revil & Florsch, 2010; Vaudelet, Revil, Schmutz, Franceschi, & Bègassat, 2011). Note that the Cole-Cole model given by Eq. (3.11) neglects the effects of polarization at interfaces or sorption onto mineral surfaces. The PFLOTTRAN-SIP framework can easily account for such modifications in conductivity, but this is a future endeavor.

Conclusions

This work developed a computational framework to couple PFLOTTRAN and E4D to model electrical-impedance and phase-shift data for SIP due to changes in subsurface processes. PFLOTTRAN and E4D are massively parallel codes that simulate processes related to fluid flow, reactive transport, and SIP. A mathematical relationship based on Archie's and Cole-Cole models linked flow and solute-transport state variables at various frequencies. A reservoir-scale tracer transport model demonstrated the proposed PFLOTTRAN-SIP framework where fluid flow and tracer concentration evolution were simulated over one year. Then, I simulated electrical potentials for various electrode configurations at different frequencies. These simulations showed that contrast in real potential was minimal even as the frequency varied. However, there was a significant change in contrast of complex potentials across frequencies. Phase shift (combination of real and complex potentials) helped identify the region where tracer concentration was high. This analysis showed that SIP has two major advantages over ERT. First, SIP provided frequency-dependent electrical impedance data. Second, phase-shift signatures obtained from SIP analysis identified and constrained geochemically altered zones. Combining frequency-dependent real potential, complex

potential, and phase responses from a SIP survey/simulation paints a more detailed picture of the subsurface with an enhanced ability to detect contaminants/tracers. Moreover, coupling fluid flow, reactive transport, and SIP models can better detect contaminants compared to either the ERT or SIP method alone. For instance, through the specified numerical example, solute transport simulations provided insight into the tracer distribution. This information was used to customize SIP inversion to estimate frequency-dependent electrical conductivities, which yielded an improved image of tracer concentration at different frequencies. Although this work focused on simulating tracer transport, it could also be applied to detect hydrocarbon flow, changes in the subsurface due to geochemical reactions, sulfide minerals, metallic objects, municipal wastes, and salinity intrusion.

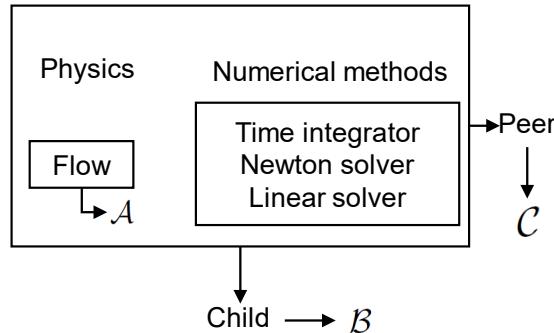


Figure 3.1: Peer and child process model class of PFLOTRAN (redrawn from reference (Johnson et al., 2017)).

Algorithm 1 Overview of the proposed PFLOTRAN-SIP framework for simulating electrical impedance data

- 1: INPUT: Initial and boundary conditions for fluid flow and solute transport models in PFLOTRAN, fluid density, porosity, saturation, volumetric source/sink with its location, intrinsic and relative permeabilities, dynamic viscosity, mass flow rate, diffusion/dispersion coefficients, tortuosity, solute source/sink with its location, Archie's and Cole-Cole model parameters, total simulation time, time-step for PFLOTRAN, interrogation frequencies, electrode locations and measurement configuration, number of processors for PFLOTRAN and E4D, and meshes for PFLOTRAN and E4D.
 - 2: Solve Eqs. (3.6)–(3.8) for fluid pressure, fluid saturation, and fluid velocity.
 - 3: Solve Eq. (3.9) to calculate the spatio-temporal distribution of solute concentration.
 - 4: Transfer solute concentration from PFLOTRAN to the E4D master processor to perform SIP simulations at specific times.
 - 5: Receive numerical model setup information from PFLOTRAN input files to perform mesh interpolation for SIP simulations.
 - 6: Broadcast run information and distribute mesh assignments to E4D slave processors.
 - 7: Calculate the mesh interpolation matrix Eq. (3.14) to interpolate PFLOTRAN simulation outputs (e.g., solute concentrations) onto the E4D mesh for SIP simulations.
 - 8: Calculate electrical conductivities using Archie's model Eq. (3.10).
 - 9: Calculate frequency-dependent electrical conductivities using the Cole-Cole model Eq. (3.11).
 - 10: Pass real and complex conductivities calculated at different frequencies to the E4D master processor to perform SIP simulations.
 - 11: Broadcast real and complex conductivities to E4D slave processors to compute pole solutions for electrode configurations.
 - 12: Solve Eq. (3.5) to compute real and complex electrical potential at different frequencies and solute concentrations at specified times.
-

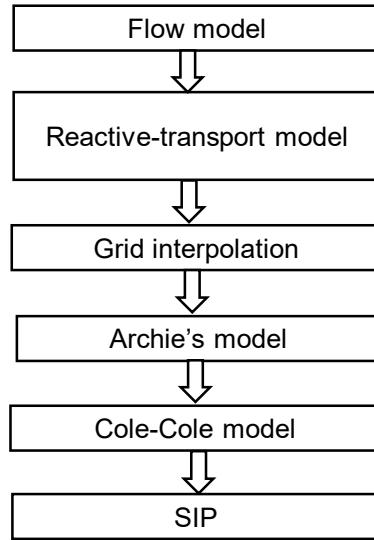


Figure 3.2: Steps involved in coupling fluid flow, solute transport, and SIP process models in the PFLOTRAN-SIP framework; further details are available (Johnson & Thomle, 2017; Lichtner et al., 2015).

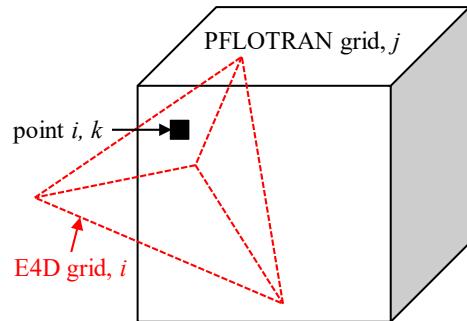


Figure 3.3: Schematic of interpolation of state variables (e.g., solute concentration) on the PFLOTRAN mesh (cube) on to E4D mesh (tetrahedron), redrawn from (Johnson et al., 2017).

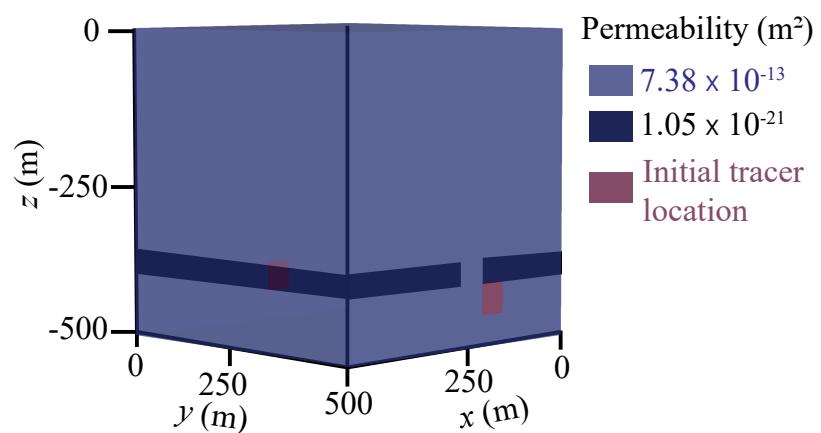


Figure 3.4: Schematic of the initial boundary value problem.

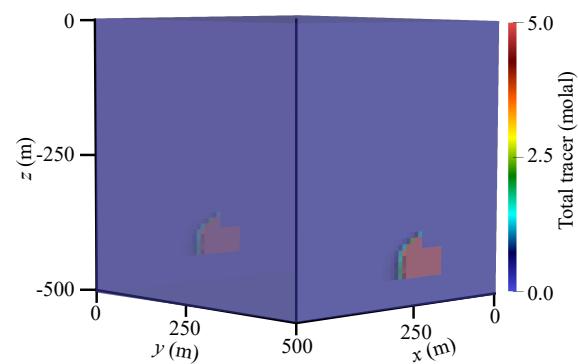


Figure 3.5: Spatial distribution of tracer concentrations after one year.

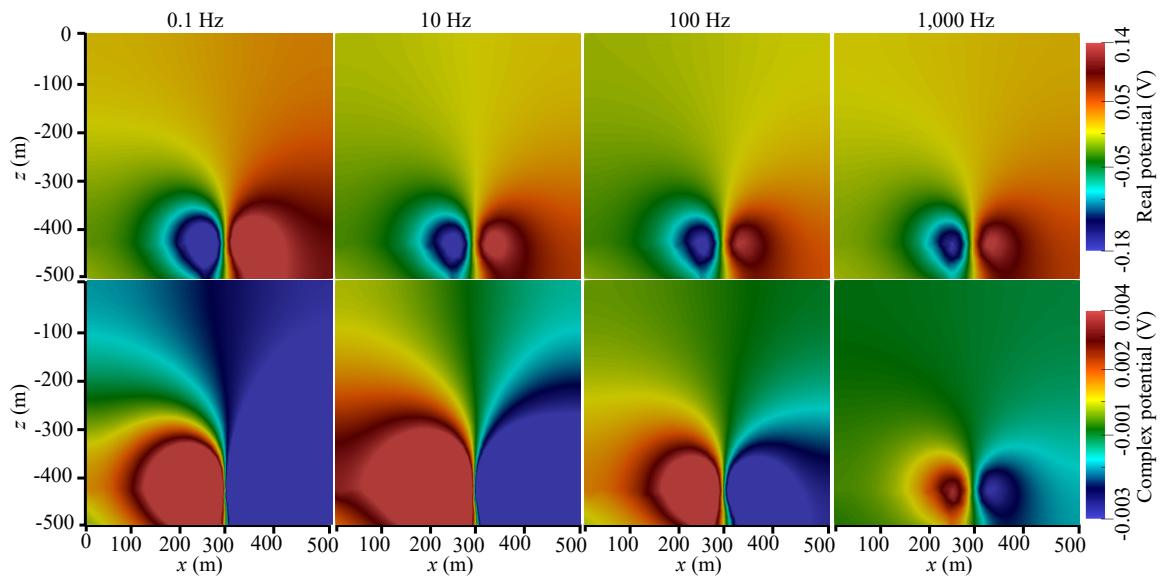


Figure 3.6: Slices of simulated real (top) and complex (bottom) electrical potentials/impedances at $y = 250$ m for a single electrode-measurement configuration after one year.

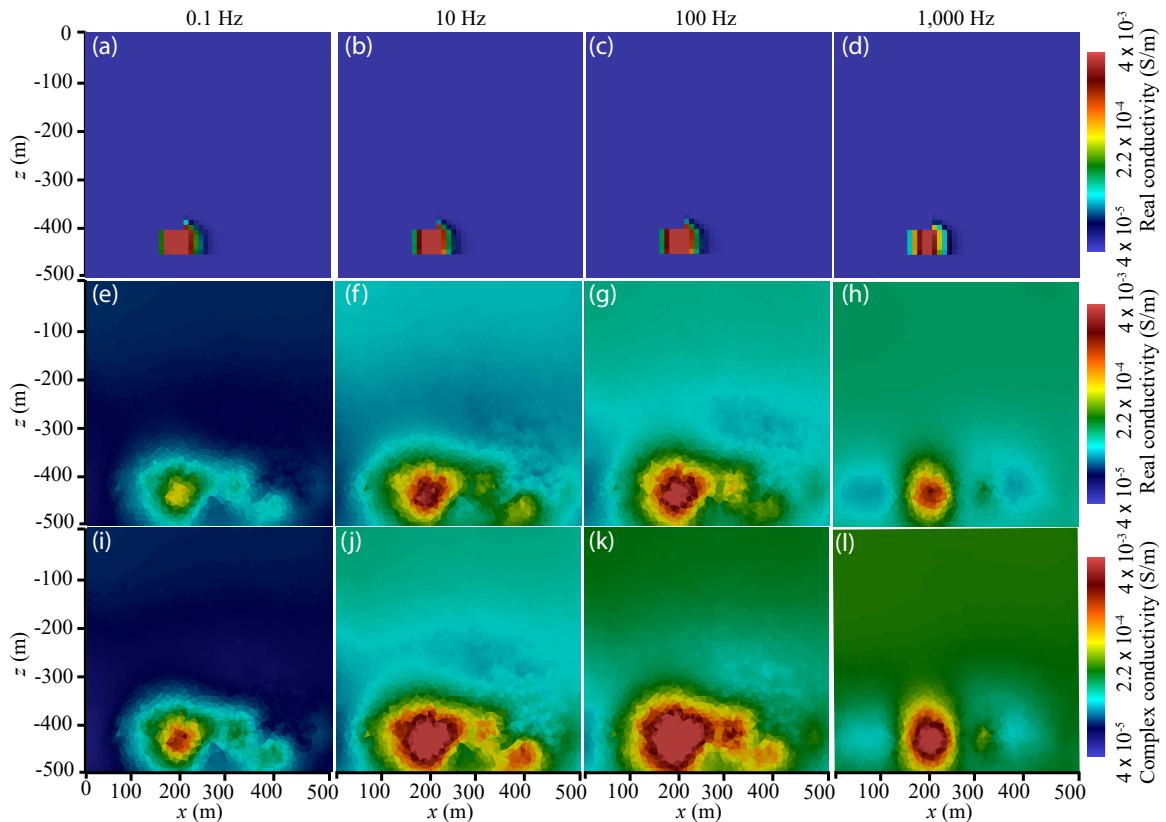


Figure 3.7: Simulated and estimated frequency-dependent electrical conductivities at $y = 250$ m after one year (a)-(d) True-electrical conductivities from the PFLOTTRAN-SIP framework, (e)-(h) estimated bulk-real conductivities from SIP inversion, and (i)-(l) estimated bulk complex electrical conductivities from SIP inversion.

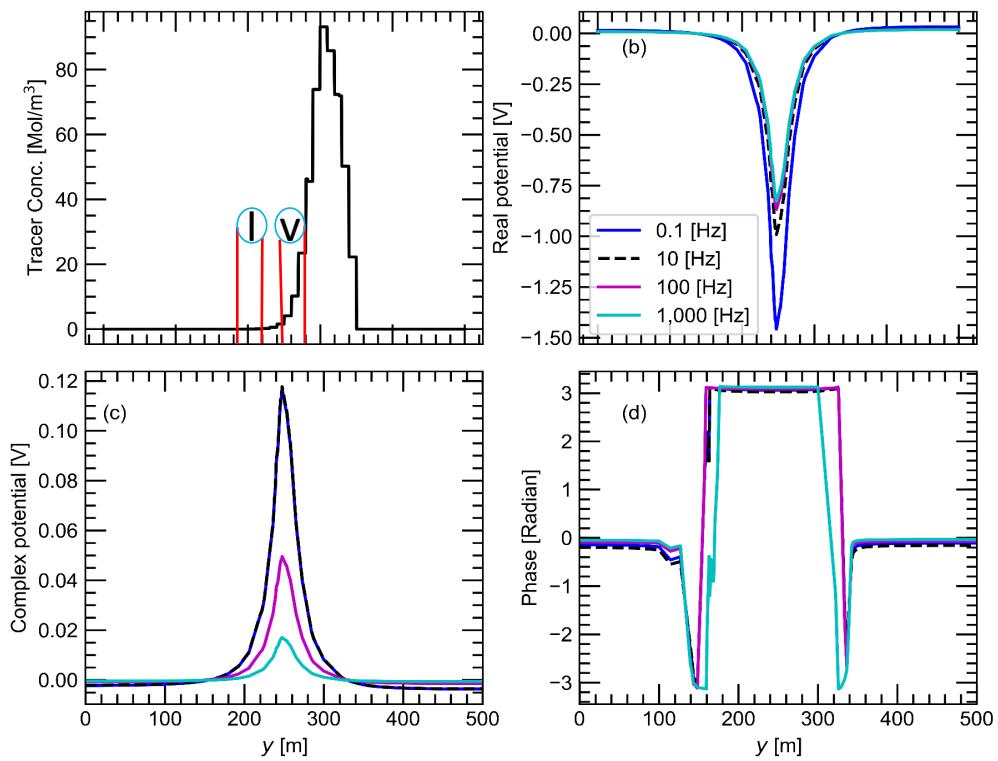


Figure 3.8: Distribution of (a) tracer concentration where I and V represents current and potential electrodes, respectively, (b) real potential, (c) complex potential, and (d) phase shift along the y -axis at $x = 250$ m and $z = -425$ m.

CHAPTER FOUR

Manuscript 3: ML to Reactive-transport Data

Abstract

Mixing phenomena are important mechanisms controlling flow, species transport, and reaction processes in fluids and porous media. Accurate predictions of reactive mixing are critical for many Earth and environmental science problems such as contaminant fate and remediation, macroalgae growth, and plankton biomass evolution. To investigate mixing dynamics over time under different scenarios (e.g., anisotropy, fluctuating velocity fields), a high-fidelity, finite-element-based numerical model is built to solve the fast, irreversible bimolecular reaction-diffusion equations to simulate a range of reactive-mixing scenarios. A total of 2,315 simulations are performed using different sets of model input parameters comprising various spatial scales of vortex structures in the velocity field, time-scales associated with velocity oscillations, the perturbation parameter for the vortex-based velocity, anisotropic dispersion contrast (i.e., ratio of longitudinal-to-transverse dispersion), and molecular diffusion. Outputs comprise concentration profiles of the reactants and products. The inputs and outputs of these simulations are concatenated into feature and label matrices, respectively, to train 20 different machine learning (ML) emulators to approximate system behavior. The 20 ML emulators based on linear methods, Bayesian methods, ensemble learning methods, and multilayer perceptron (MLP), are compared to assess these models. The ML emulators are specifically trained to classify the state of mixing and predict three quantities of interest (QoIs) characterizing species production, decay (i.e., average concentration, square of average concentration), and degree of mixing (i.e., variances of species concentration). Linear classifiers and regressors fail to reproduce the QoIs; however, ensemble methods (classifiers and regressors) and the

MLP accurately classify the state of reactive mixing and the QoIs. Among ensemble methods, random forest and decision-tree-based AdaBoost faithfully predict the QoIs. At run time, trained ML emulators are $\approx 10^5$ times faster than the high-fidelity numerical simulations. Speed and accuracy of the ensemble and MLP models facilitate uncertainty quantification, which usually requires 1,000s of model run, to estimate the uncertainty bounds on the QoIs.

Introduction

reactive-transport phenomena dictate the distribution of chemical species in fluids (e.g., coastal waters) and subsurface porous media. Accurate quantification of species concentration is critical to remediation applications such as nuclear remediation, spill distribution, algal-bloom forecasting, etc (Ahmmmed, 2015; Cama, Soler, & Ayora, 2019; L. Chen, Wang, Kang, & Tao, 2018; Lagneau et al., 2019; Lichtner et al., 2015; Lichtner, Steefel, & Oelkers, 2019; Molins & Knabner, 2019; Öztürk, Ashraf, Aksoy, Ahmad, & Hakeem, 2015; Rolle & Le Borgne, 2019; Sin & Corvisier, 2019). Parameters that influence reactive-transport in fluids and subsurface porous media include the structure of the flow field (e.g., chaotic advection), fluid injection/extraction (i.e., location of wells, injection/extraction rates), subsurface heterogeneity, dispersion, and anisotropy (Mudunuru & Karra, 2019; Vesselinov, Mudunuru, Karra, O’Malley, & Alexandrov, 2019). These parameters have variable impacts on important quantities of interest (QoIs) such as species production and decay (e.g., average and squared average species concentrations) and degree of mixing (i.e., variances of species concentrations). For QoIs, nonlinear partial differential equations are solved using high-fidelity numerical methods (e.g., finite-difference, -element, or -volume methods) that can take hours to days (for $\approx \mathcal{O}(10^6) - \mathcal{O}(10^9)$ degrees-of-freedom) on state-of-the-art, high-performance computing (HPC) machines. Such computation times preclude real-time predictions, which can be critical to decision making for remedia-

tion activities. Hence, alternative faster approaches are needed and machine learning (ML)-based emulators show promise (Hulbert et al., 2019; Srinivasan et al., 2018; Valls et al., 2018; Viswanathan et al., 2018; Wu et al., 2018). Here, I built and compared various ML emulators to predict reactive-transport QoIs. The ML emulators were trained and tested using data from high-fidelity, finite-element numerical simulations, which expressly reflected the underlying reaction-diffusion physics in anisotropic porous media.

Given sufficient data, ML models can successfully detect, quantify, and predict different types of phenomena in the geosciences (Bergen et al., 2019; Reichstein et al., 2019). Applications include remote sensing (Mesa et al., 2018; Valls et al., 2018), ocean wave forecasting (James et al., 2018; O'Donncha et al., 2018, 2019), seismology (Hulbert et al., 2019; Leduc et al., 2017; M.-Zook & Ruppert, 2017; Reynen & Audet, 2017; Wu et al., 2018; Yuan et al., 2019), hydrogeology (Barzegar et al., 2018; Srinivasan et al., 2018; Viswanathan et al., 2018), and geochemistry (Cracknell et al., 2014; Galiano et al., 2015; Kirkwood et al., 2016; Oonk & Spijker, 2015; Zuo, 2017). ML emulators (also known as surrogate models or reduced-order models) can be fast, reliable, and robust when trained on large datasets (Bergen et al., 2019; Reichstein et al., 2019; Salah, 2018). ML emulators are constructed using training data (e.g., features and labels), which include inputs and outputs either from field data, experimental data, high-fidelity numerical simulations, or any combination of these (Brunton & Kutz, 2019; Salah, 2018). In this paper, I compared emulators based on generalized linear methods (Hastie, Tibshirani, & Friedman, 2009; Marquardt, 1970), Bayesian methods (Murphy, 2012; Tipping, 2001), ensemble methods (Breiman, 1996; Freund & Schapire, 1997), and an MLP (Montavon, Samek, & Müller, 2018; Rumelhart, Hinton, & Williams, 1988; Rumelhart et al., 1988) to predict various QoIs.

Previous researchers have used unsupervised and supervised ML methods to reproduce reactive-transport QoIs. Vesselinov et al. (Vesselinov et al., 2019) used

non-negative tensor factorization with custom k -means clustering (unsupervised ML) to identify hidden features in the solutions to reaction-diffusion equations. They determined that anisotropy features (i.e., longitudinal and transverse dispersion) govern reactive mixing at early to middle times while molecular diffusion controls product formation at late times. They also quantified the effects of longitudinal and transverse dispersion and molecular diffusion on species production and decay over time. Mudunuru and Karra (Mudunuru & Karra, 2019) ranked the importance of input parameters/features on reactive-transport QoIs. Also, they developed support vector machine (SVM) and support vector regressor (SVR) emulators to classify the degree of mixing and to predict QoIs. However, SVM/SVR training times significantly increase with the size of the training data set (Mudunuru & Karra, 2019). To obviate this problem, this study built ML emulators whose training times are $\approx 10^5$ times faster than SVM and SVR without compromising accuracy.

Specifically, I compared one linear classifier, two Bayesian classifiers, an ensemble classifier, an MLP classifier, seven linear regressors, six ensemble regressors, and an MLP regressor. Emulator performance was assessed according to training and testing scores, training time, and R^2 score on the QoIs from a blind data set. The blind data set included six realizations that were not seen during training and testing phases. This study addressed the following questions: (1) Can ML emulators accurately classify the mixing state of the anisotropic reaction-diffusion system? (2) How accurately do they predict QoIs of reactive mixing? (3) How fast can they be trained? (4) How does each emulator rank overall?

Governing Equations for Reactive Mixing

Let $\Omega \subset \mathbb{R}^d$ be an open bounded domain, where d indicates the number of spatial dimensions. The boundary was denoted by $\partial\Omega$, which was assumed to be piece-wise smooth. Let $\bar{\Omega}$ be the set closure of Ω and let spatial point $\mathbf{x} \in \bar{\Omega}$. The di-

vergence and gradient operators with respect to \mathbf{x} were denoted by $\text{div}[\bullet]$ and $\text{grad}[\bullet]$, respectively. Let $\mathbf{n}(\mathbf{x})$ be the unit outward normal to $\partial\Omega$. Let $t \in]0, \mathcal{I}[$ denote time, where \mathcal{I} is the length of time of interest. The governing equations were posed on $\Omega \times]0, \mathcal{I}[$ and the initial condition was specified on $\overline{\Omega}$. Consider the fast bimolecular reaction where species A and B reacted irreversibly to yield product C :



The governing equations for this fast bimolecular reaction without volumetric sources/sinks are:

$$\frac{\partial c_A}{\partial t} - \text{div}[\mathbf{D}(\mathbf{x}, t) \text{grad}[c_A]] = -n_A k_{AB} c_A c_B \quad \text{in } \Omega \times]0, \mathcal{I}[, \quad (4.2a)$$

$$\frac{\partial c_B}{\partial t} - \text{div}[\mathbf{D}(\mathbf{x}, t) \text{grad}[c_B]] = -n_B k_{AB} c_A c_B \quad \text{in } \Omega \times]0, \mathcal{I}[, \quad (4.2b)$$

$$\frac{\partial c_C}{\partial t} - \text{div}[\mathbf{D}(\mathbf{x}, t) \text{grad}[c_C]] = +n_C k_{AB} c_A c_B \quad \text{in } \Omega \times]0, \mathcal{I}[, \quad (4.2c)$$

$$c_i(\mathbf{x}, t) = c_i^P(\mathbf{x}, t) \quad \text{on } \Gamma_i^D \times]0, \mathcal{I}[\quad (i = A, B, C), \quad (4.2d)$$

$$(-\mathbf{D}(\mathbf{x}, t) \text{grad}[c_i]) \cdot \mathbf{n}(\mathbf{x}) = h_i^P(\mathbf{x}, t) \quad \text{on } \Gamma_i^N \times]0, \mathcal{I}[\quad (i = A, B, C), \quad (4.2e)$$

$$c_i(\mathbf{x}, t = 0) = c_i^0(\mathbf{x}) \quad \text{in } \overline{\Omega} \quad (i = A, B, C). \quad (4.2f)$$

Traditional numerical formulations for Eqs. (4.2a)–(4.2f) can yield nonphysical solutions for chemical species concentration (Nakshatrala, Mudunuru, & Valocchi, 2013). Also, when anisotropy dominates, the standard Galerkin formulation produces erroneous concentrations (Mudunuru & Nakshatrala, 2012, 2017; Mudunuru, Shabouei, & Nakshatrala, 2015; Nakshatrala et al., 2013). To overcome these problems, a non-negative, finite-element method was used to compute species concentrations (Nakshatrala et al., 2013). This method ensures that concentrations are non-negative and satisfy the discrete maximum principle.

Reaction Tank Problem and Associated QoIs

Figure 4.1 depicts the initial boundary-value problem. The model domain was a square with $L = 1$. Zero-flux boundary conditions $h_i^P(\mathbf{x}, t) = 0$ were enforced on

all sides of the domain. For all chemical species, the non-reactive volumetric source $f_i(\mathbf{x}, t)$ was equal to zero. Initially, species A and B were segregated (see Fig. 4.1) and stoichiometric coefficients were $n_A = 1$, $n_B = 1$, and $n_C = 1$. The total time of interest was $\mathcal{I} = 1$. The dispersion tensor was taken from the subsurface literature (Nakshatrala et al., 2013; Pinder & Celia, 2006):

$$\mathbf{D}_{\text{subsurface}}(\mathbf{x}) = D_m \mathbf{I} + \alpha_T \|\mathbf{v}\| \mathbf{I} + \frac{\alpha_L - \alpha_T}{\|\mathbf{v}\|} \mathbf{v} \otimes \mathbf{v}. \quad (4.3)$$

The model velocity field was used to define the dispersion tensor according to stream function (Adrover, Cerbelli, & Giona, 2002; Mudunuru & Nakshatrala, 2016; Tsang, 2009):

$$\psi(\mathbf{x}, t) = \begin{cases} \frac{1}{2\pi\kappa_f} [\sin(2\pi\kappa_f x) - \sin(2\pi\kappa_f y) + v_0 \cos(2\pi\kappa_f y)] & \text{if } \nu T \leq t < (\nu + \frac{1}{2}) T \\ \frac{1}{2\pi\kappa_f} [\sin(2\pi\kappa_f x) - \sin(2\pi\kappa_f y) - v_0 \cos(2\pi\kappa_f x)] & \text{if } (\nu + \frac{1}{2}) T \leq t < (\nu + 1) T \end{cases} \quad (4.4)$$

Using Eq. (4.4), the divergence-free velocity field components are:

$$v_x(\mathbf{x}, t) = -\frac{\partial \psi}{\partial y} = \begin{cases} \cos(2\pi\kappa_f y) + v_0 \sin(2\pi\kappa_f y) & \text{if } \nu T \leq t < (\nu + \frac{1}{2}) T \\ \cos(2\pi\kappa_f y) & \text{if } (\nu + \frac{1}{2}) T \leq t < (\nu + 1) T \end{cases}, \quad (4.5)$$

$$v_y(\mathbf{x}, t) = +\frac{\partial \psi}{\partial x} = \begin{cases} \cos(2\pi\kappa_f x) & \text{if } \nu T \leq t < (\nu + \frac{1}{2}) T \\ \cos(2\pi\kappa_f x) + v_0 \sin(2\pi\kappa_f x) & \text{if } (\nu + \frac{1}{2}) T \leq t < (\nu + 1) T \end{cases}. \quad (4.6)$$

In Eqs. (4.5)–(4.6), T controls the oscillation of the velocity field from clockwise to anti-clockwise. v_0 is the perturbation parameter of the underlying vortex-based flow field. Larger values of v_0 skew the vortices into ellipses while smaller values of v_0 yield circular vortex structures in the velocity field. $\frac{\alpha_L}{\alpha_T}$ controls the magnitude of

the anisotropic dispersion contrast. Smaller values of $\frac{\alpha_L}{\alpha_T}$ indicate less anisotropy and vice versa. The magnitude of $\kappa_f L$ governs the size of the vortex structures in the flow field (Mudunuru & Karra, 2019; Vesselinov et al., 2019). Note that varying v_0 does not significantly alter vortex locations.

The QoIs were defined as:

- (1) Species production/decay, which can be analyzed by calculating normalized average concentrations, \bar{c}_i , and normalized average of squared concentrations, $\overline{c^2}_i$. Normalized average of squared concentration, $\overline{c^2}_i$, provides information on the species production/decay as a function of the eigenvalues of anisotropic dispersion. For example, see Theorem 2.3 in Reference (Mudunuru & Karra, 2019), which shows that $\overline{c^2}_i$ is bounded above and below by an exponential function of minimum and maximum eigenvalues of anisotropic dispersion.

These quantities are:

$$\bar{c}_i := \frac{\langle c_i(t) \rangle}{\max [\langle c_i(t) \rangle]} \quad \text{where } \langle c_i(t) \rangle = \int_{\Omega} c_i(\mathbf{x}, t) d\Omega, \quad (4.7)$$

$$\overline{c^2}_i := \frac{\langle c_i^2 \rangle}{\max [\langle c_i^2 \rangle]} \quad \text{where } \langle c_i^2(t) \rangle = \int_{\Omega} c_i^2(\mathbf{x}, t) d\Omega. \quad (4.8)$$

- (2) Degree of mixing was defined as the variance of concentration:

$$\sigma_{c_i}^2 := \frac{\langle c_i^2 \rangle - \langle c_i \rangle^2}{\max [\langle c_i^2 \rangle - \langle c_i \rangle^2]}. \quad (4.9)$$

Note that the values for \bar{c}_i , $\overline{c^2}_i$, and $\sigma_{c_i}^2$ were non-negative and ranged from 0 to 1
 $\forall i = A, B, C$.

Feature Generation

First, a 2D numerical model was built using first-order finite-element structured triangular mesh, which had 81 nodes on each side. A total of 2,500 high-fidelity numerical simulations were run for different sets of reaction-diffusion model input

parameters, of which 2,315 ran to completion because certain parameter combinations did not yield to stable solution. Each simulation used 1,000 time steps ($\mathcal{I} = 0.0$ to 1.0 with a uniform time step of 0.001). Features included: longitudinal-to-transverse anisotropic dispersion ratio $\frac{\alpha_L}{\alpha_T}$, molecular diffusion D_m , the perturbation parameter of the underlying vortex-based velocity field v_0 , and velocity field characteristics scales $\kappa_f L$ and T . Specifically, input parameters were: $v_0 = [1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}]$, $\frac{\alpha_L}{\alpha_T} = [1, 10^1, 10^2, 10^3, 10^4]$, $D_m = [10^{-8}, 10^{-3}, 10^{-2}, 10^{-1}]$, $\kappa_f L = [1, 2, 3, 4, 5]$, and $T = [1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 4 \times 10^{-4}, 5 \times 10^{-4}]$. α_T was varied with α_L held at 1.0. Five features for each of the 2,315 models with 1,000 time steps formed the feature matrix with dimensions $2,315,000 \times 5$.

Machine Learning Emulators

Labels (QoIs) and Preprocessing

Labels were the QoIs of the 2,315 simulations at each time step yielding label vectors. Features and labels were concatenated into training and testing data forming a $2,315,000 \times 6$ matrix. For ML classification, the degree of mixing in the system was characterized by four classes representing: Class-1 (well mixed), Class-2 (moderately mixed), Class-3 (weakly mixed), and Class-4 (ultra-weak mixing). The corresponding σ_i^2 for these classes are 0.0–0.25, 0.25–0.5, 0.5–0.75, and 0.75–1.0, respectively. Of course, additional classes could be defined although this would necessitate re-training of ML emulators. These data were partitioned into training and testing data during construction of the ML emulators and Table 4.1 lists the different partitions. Each emulator was trained using the three different data partitions and the performance of each assessed. First, 0.9% of data were used as training data to identify optimized hyperparameters and other tunable parameters. Subsequently, emulators using the optimized hyperparameters were validated against 63% and 81% of data partitions.

Preprocessing is typically required for ML emulator development. ML emulators that use the Euclidean norm (e.g., kernel-based methods) must have all features/input parameters of the same scale to make accurate predictions (Buitinck et al., 2013; Müller & Guido, 2016; Pedregosa et al., 2011a). Common preprocessors are standardization (recasting all feature data into the standard normal distribution $N(0, 1)$), normalization (independently scaling each feature between 0 and 1), and max-abs scaling (scale and translate individual features such that the maximal absolute value of a feature is 1). In this study, except for Random Forests (RF), which is agnostic to feature scaling, because features are neither sparse nor skewed and do not have outliers, all data were standardized. For polynomial regression, I used the quadratic transformation of the data.

Optimization of Hyperparameter and Other Tunable Parameters

Every ML emulator learns a function or a set of functions by comparing features and corresponding labels. During this process, different hyperparameters for each ML emulator control the learning process. Some common hyperparameters are regularization, learning rate, and the cost function. In addition, there are additional tunable parameters for each ML emulator that also speed the learning process and make a more robust emulator, including the number of training iterations, kernel, truncation value, etc. Because hyperparameter optimization is an exhaustive, time-consuming process, 0.9% of the data (23 simulations) were used with the `Gridsearch` algorithm in `Scikit-learn` (Pedregosa et al., 2011b), a Python ML package. Tables 4.2 and 4.3 list the hyperparameters for each ML emulator. Later, 7% and 9% of the data were used for validation with 30% or 10% reserved as blind data for testing.

Because, ML emulators can introduce bias during training, overfitting is a common phenomenon. To ameliorate this, k -fold cross-validation algorithm was used to avoid bias, to determine optimal computational times, and to calculate reliable vari-

Table 4.1: Summary of training and testing data partitions used in ML emulator development and testing.

Training data	Validation data	Testing data	Size of samples for QoIs	
			Training	Testing
0.9% (20)	0.1 (3) %	99% (2,292)	20,150	2,291,850
63% (1458)	7% (162)	30% (690)	1,458,500	694,500
81% (1875)	9% (208)	10% (230)	1,875,500	231,500

ances (Chou, Tsai, Pham, & Lu, 2014; Kohavi, 1995). In this work, 10-fold cross-validation was used (Chou et al., 2014; Kohavi, 1995). First, it subdivided training data into equal ten subsets. Then, it used nine sets for training while one set was left for validation, and this process was repeated leaving out each subset once. The average performance on the 10 withheld data sets were reported along with their variance.

ML Emulators

This research applied 20 ML emulators to classify the state of reactive mixing and to predict the reactive-transport QoIs. Among the 20 ML emulators, eight were linear, five were Bayesian, six were ensemble, and one was an MLP. The eight linear ML emulators were ordinary least square regressor (LSQR), ridge regressor (RR), lasso regressor (LR), elastic-net regressor (ER), Huber regressor (HR), polynomial, logistic regression (LogR), and kernel ridge (KR). Among the linear emulators, only LogR is a classifier. The five Bayesian techniques were – Bayesian ridge (BR), Gaussian process (GP), naïve Bayes (NB), linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA). Among these Bayesian emulators, LDA and QDA are classifiers and remaining are regressors. The six ensemble ML emulators were bagging, decision tree (DT), random forest (RF), AdaBoost (AdaB), DT-based AdaB, and gradient boosting method (GBM). Among the six ensemble emulators, RF was used as both classifier and regressor. MLP was also used as both classifier and regressor.

Linear ML Emulators

Linear ML emulators tend to fit a straight line to the labels. Each linear emulators' equation along with its corresponding cost function is described in Appendix B. A brief mathematical description of each linear ML emulator is explained at Appendix A. The equation for polynomial regression is not listed here because it applies the LSQR formula to quadratic-scaled data. For LSQR and polynomial regressor, I optimized intercept. For RR, α_2 and ϵ (tolerance/threshold) were optimized. For LR, α_1 , ϵ , and maximum iteration number were optimized. For ER, α_1 , α_2 , ϵ , l_1 ratio, and maximum iteration number were optimized. For HR, α_1 , ϵ , and maximum iteration number were optimized. Optimized hyperparameters and other tunable parameters (bolded) for linear ML emulators were listed in Table 4.2. For Logistic regression, multi-class (binary or multi-class), solver, ϵ , and maximum number of iterations were optimized and corresponding settings are presented in Table 4.2. Tested solvers included Newton's method, limited memory large-scale bound constrained (LBFGS) solver, and the stochastic average gradient (SAG) solver. For KR, α_1 , λ , and kernels were optimized (see, Table 4.2).

Bayesian ML Emulators

Bayesian ML emulators apply Bayes' rule to learn function from labels to predict equivalent label. Equations for Bayesian ML emulators are described in Appendix B. Also, a brief mathematical description of each Bayesian ML emulator is explained in Appendix A. For BR, β , ω , maximum iterations, and ϵ were the hyperparameters and their optimized values are shown in bold in Table 4.3. For GP, kernel was optimized and its best is listed in Table 4.3. In NB, only priors and variance smoothing were hyperparameters. For LDA, solver was optimized; solvers included singular value decomposition (SVD), LSQR, eigen value decomposition. Among these three, SVD was fastest. For QDA, only tolerance was optimized and best value was 10^{-4} .

Ensemble Emulators

If the relationship between features and label is nonlinear, linear ML emulators are not expected to perform well. Instead, nonlinear ML emulators such as an MLP and ensemble methods should work better. Ensemble methods bootstrap (random sampling with replacement) data to develop different tree models/predictors. Each label is used with replacement as input for developing individual models; therefore, tree models have different labels based on the bootstrap process. Because bootstrapping captures many uncorrelated base learners to develop a final model, it reduces variance; resulting in a reduced prediction error. Also, in ensemble models, many different trees predict the same target variable; therefore, they predict better than any single tree alone.

Ensemble techniques are further classified into Bagging (bootstrapping aggregating) and Boosting (form many weak trees/learners into a strong tree). While bagging emulators work best with strong and complex trees (e.g., fully developed decision trees), boosting emulators work best with weak models (e.g., shallow decision trees). In this study, several averaging/bagging and boosting ensemble emulators were explored to classify and predict reactive mixing. The averaging emulators included bagging and RF while boosting emulators include AdaBoost (AdaB), DT-based AdaB, and gradient boosting method (GBM).

For DT, maximum tree depth, maximum number of features, and minimum sample splitting were optimized and best settings are listed in bold in Table 4.4. In Bagging, tree number, bootstrapping, and maximum number of features were optimized and their best settings are prescribed in Table 4.4. In RF, maximum depth of tree, tree number in forest, minimum sample splitting number, bootstrapping, and maximum feature number were optimized and their best settings are listed in bold in Table 4.4. For AdaB and DT-based AdaB, number of trees, loss function, and γ were optimized and their best settings are in bold in Table 4.4. In GBM, number of

trees, sub-sampling, and γ were optimized and their best settings are prescribed in bold in Table 4.4. For MLP, number of hidden layers, activation function, α , γ , solver, and maximum number of iteration were optimized and their best values are bold in Table 4.5. Solvers in MLP are adaptive momentum (Adam), LBFGS, and SGD.

Performance Metrics

Training time and R^2 score are performance metrics for each emulator. Training time should be fast while R^2 measures the correlation between y and \hat{y} . For n pairs of data points, the R^2 score is:

$$R^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2 - \sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (4.10)$$

which ranged from 0 to 1 for the worst and best predictions, respectively. For classification, the performance metrics is defined as:

$$\text{Accuracy} = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} \mathbb{1}(y) (\hat{y}_i = y_i), \quad (4.11)$$

where $\mathbb{1}(y)$ is the indicator function (Hastie et al., 2009).

Results

After time $t = 0$, reactants A and B were allowed to mix and formed product C . The extent of mixing depended upon the reaction-diffusion inputs (features). Increased degree of mixing increased the yield of product C . Product C yield at normalized simulation times $t = 0, 0.5$, and 1.0 are shown in Figs. 4.2-4.4 revealing the significance of $k_f L$ on product formation at different times. The importance of $\frac{\alpha_L}{\alpha_T}$ on product formation at various times was also evident. For $k_f L = 2$ and $\frac{\alpha_L}{\alpha_T} = 10^3$ (see Fig. 4.2 (a-c)) at $t = 0.1$, there is little reaction at the center of the vortices. However, regions with zero concentration decreased as $k_f L$ increases. For example, at $k_f L = 3$ and $t = 1.0$, more product was formed and negligible zero concentration of

Table 4.2: Hyperparameters and tunable parameters for generalized linear ML emulators, logistic regression, and KR with the best parameters in bold numbers and text in type-writer font.

Emulator	Hyperparameter and tunable parameter	Sought range
LSQR	Fit intercept	<code>True</code> , <code>False</code>
RR	α_2	$1.0, 100, 1,000$
	Max. no. of iterations	$50, 300, 1,000$
	α_1	$10^{-1}, 10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
LR	ϵ	$\mathbf{10^{-3}}, 10^{-4}$
	Max. no. of iterations	$50, 100, 300, \mathbf{1,000}$
	α_1 and α_2	$10^{-1}, 10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
ER	ϵ	$10^{-2}, \mathbf{10^{-3}}, 10^{-4}$
	l_1 ratio	$0.1, 0.5, 1.0$
	Max. no. of iterations	$10^2, 10^3, \mathbf{10^4}$
	Tolerance	$10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
HR	α_1	$10^{-1}, 10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
	ϵ	$10^{-3}, \mathbf{10^{-4}}, 10^{-5}$
	Max. no. of iterations	$10, 50, 100$
LogR	Multi-class	OVR, Multinomial
	Solver	Newton-cg, lbfgs, SAG
	ϵ	$10^{-3}, \mathbf{10^{-4}}, 10^{-5}$
KR	Max. no. of iterations	$10, 50, 100, 200, 300$
	α	$10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
KR	λ	$1, 2, 3$
	Kernel	linear, polynomial, RBF

Table 4.3: Hyperparameters and tunable parameters for Bayesian emulators where bold numbers and text in type-writer font parameters were best suited parameters.

Exponential sine squared ($\mathcal{K}(x, x') = \sigma^2 \exp(-2\sin^2(\pi|x - x'|/p)/l^2)$) is parameterized by a length-scale parameter (l) > 0 and a periodicity (p) > 0 .

Emulator	Hyperparameter and tunable parameter	Sought range
BR	No. of iterations	$100, 200, 300$
	ϵ	$10^{-2}, 10^{-3}, \mathbf{10^{-4}}$
GP	Kernel	Exponential sine squared, RBF
	Priors	<code>True</code> , <code>None</code>
NB	Variance smoothing	$10^{-7}, 10^{-8}, \mathbf{10^{-9}}$
LDA	Solver	SVD, LSQR, Eigen
QDA	Tolerance	$10^{-3}, \mathbf{10^{-4}}, 10^{-5}$

Table 4.4: Hyperparameters and tunable parameters for ensemble ML emulators with the best parameters in bold numbers and text in type-writer font.

Emulator	Hyperparameter and tunable parameter	Sought range
DT	Maximum depth	2, 3, None
	Max. no. of features	3, 4, 5
	Min. sample splits	5 3, 4
Bagging	No. of trees	100 , 200, 500
	Bootstrap	True, False
	Max. no. of features	3, 4, 5
RF	Maximum depth	2, 3, None
	No. of trees in the forest	250, 500 , 1,000
	Bootstrap	True, False
AdaB	Max. no. of features in a tree	3, 4, 5
	Min. sample splits	2, 3, 4
	No. of trees	100, 200, 300
DT-based AdaB	Loss function type	linear, square, exponential
	γ	0.1, 0.5, 0.75, 1.0
	No. of trees	100, 200, 500
GBM	Loss function type	linear, square, exponential
	γ	0.1, 0.5, 1.0
	No. of trees	100, 200, 500
GBM	Sub-sample	0.5, 0.7, 0.8
	γ	0.1, 0.25, 0.5

Table 4.5: Hyperparameters and tunable parameters for MLP emulator with the best parameters in bold numbers and text in type-writer font.

Emulator	Hyperparameter and tunable parameter	Sought range
MLP	No. of hidden layers	5, 25, 50, 100, 200
	Activation function	ReLU, tanh, logistic
	α	$10^{-1}, 10^{-2}, \mathbf{10^{-4}}$
	γ	$10^{-1}, 10^{-1}, \mathbf{10^{-3}}$
	Solver	Adam, lbfgs, sgd
	Max. no. of iterations	1–5,000, 200

C was present in the model domain. At $k_f L = 5$ and $t = 1.0$, the system was nearly well-mixed even at high anisotropy. Because high $k_f L$ created a higher number of vortices that enhanced reactant mixing, it increased product yield. Figure 4.3 shows the product C yield under medium anisotropy ($\frac{\alpha_L}{\alpha_T}$) from 1,000 to

100 improved product yield even under low $\kappa_f L$ (see Fig. 4.3(c)). Among $\frac{\alpha_L}{\alpha_T}$, $k_f L$, and D_m , $\frac{\alpha_L}{\alpha_T}$ controlled the reaction at early times while $k_f L$ and D_m controlled reaction at late times. Higher values of $\frac{\alpha_L}{\alpha_T}$ decreased product yield but higher values of $k_f L$ and D_m increased the product yield.

ML emulators were also used to classify the mixing state of the system. Out of 20 ML emulators, only LogR, LDA, QDA, RF, and MLP were used for classification. Table 4.6 shows the training score, testing score, sample sizes, and training time for each linear ML emulator. Because the progress of reactive-transport is nonlinear, linear ML emulators (e.g., LogR, LDA, QDA) failed to learn an accurate function for the state of mixing. Mixing state classification by linear classifiers on training and testing data had accuracies $<80\%$. Nonlinear classifiers such as RF and MLP learnt better functions whose accuracies were $>95\%$. Results from RF and MLP were used to plot the confusion matrix of Figure 4.5 to show true and false predictions. Confusion matrix for RF and MLP were constructed using approximately 1% of data (23 simulations as training data) while the remaining 99% (2,292 simulations) data were used as testing data. In the confusion matrix, diagonal and off-diagonal elements show true and false predictions, respectively. The RF and MLP emulators false prediction scores were less than 2% and 10%, respectively. Similar trends were observed for species A and B , hence the confusion matrices for them are not shown here.

Table 4.7 shows the training and testing scores for the six linear ML emulators. Although training times were short (always <20 minutes), training and testing R^2 scores never exceed 73%. Also, three Bayesian ML emulators (e.g., BR, GP, NB) were applied to predict QoIs that show similar performance as linear emulators. Among them, training and testing scores of BR and NB were $<75\%$. GPs failed to converge for large datasets because of lack of sparsity and due to large training sample size ($\approx \mathcal{O}(10^4) - \mathcal{O}(10^6)$); however, GP trained on a smaller sample size scored $>99\%$. This increased prediction capability of GP compared to other Bayesian ML emulators can

be attributed to the RBF kernels. As species A and B decayed or product C increased in an exponential fashion, RBF kernels used by GP emulators were better suited to model such a reactive-transport system. Hence, GP emulators trained on small (0.25% of data) data performed best and showed promise to predict QoIs.

Table 4.8 compares the training and testing scores for ensemble and MLP emulators. The R^2 scores for training and testing datasets were greater than 90% (e.g., Bagging, DT, RF, MLP). For six unseen (blind) realizations, Bagging, DT, RF, AdaBoost, DT-based AdaB, and GBM showed astounding match between true QoIs and their corresponding predictions by RF and GBM. Here, only figures for RF and GBM emulators (see, Figures 4.6–4.7) are shown here because the remaining ensemble emulators showed the similar trend. These results indicate that tree-based methods outperformed linear ML methods in capturing the QoIs of the reactive-transport system. Also, Figure 4.8 shows the QoIs predictions by the MLP emulator for the six blind realizations. The test R^2 score ($>99\%$) on different data sizes and generalized cross-validation during emulator development indicate that overfitting was not a problem. As the size of the training dataset increased, the ensemble and MLP emulator development time increased.

Finally, the computational costs to run the high-fidelity model and the ML emulators were investigated. Tables 4.6–4.8 compare the computational cost of development of various ML emulators. These tables provide details on training time for various training dataset sizes on a 32-core processor (Intel(R) Xeon(R) CPU E5-2695 v4 2.10GHz). A single, high-fidelity numerical simulation required approximately 1,500 s on a single core. Testing an ML emulator (e.g., RF, MLP) took 0.01–0.1 s about 1/100,000th of the time of the high-fidelity numerical simulation.

Discussion

A suite of linear, Bayesian, and nonlinear ML emulators are trained to classify and replicate QoIs from high-fidelity anisotropic bi-linear diffusion numerical simulations. For this highly nonlinear system, linear and Bayesian ML emulators never exceeded 70% classification accuracy while LogR and QDA achieved only 75% and 77% classification accuracies, respectively. On the other hand, nonlinear emulators performed well (95% classification accuracies for RF and MLP). For the regression problem (predicting the three QoIs for each chemical species), as expected, linear regressors predicted QoIs at only $R^2 = 69\%$, but decision-tree-based ensembles and the MLP neural network performed remarkably well with accuracy $>85\%$. DTs (with and without AdaBoost), RFs, and the MLP all had $R^2 = 99\%$ with GBM (98%), bagging (95%), and AdaBoost (85%) performed somewhat worse.

These results indicate that ensemble emulators outperformed other ML emulators in predicting the progress of reactive mixing on unseen data. However, not all of them performed equally. For example, RF outperformed other averaging ensembles (e.g., Bagging, DT) while DT-based AdaB outperformed other boosting methods (e.g., AdaB, GBM). Each bagging/averaging ensemble methods introduced randomness and voting-based evaluation metrics in unique ways; therefore, their performance was not the same. For example, DTs often used the first feature to split; resultantly, the order of variables in the training data was critical for DT-based model construction. Also, in DTs, trees were pruned and not fully grown. Contrarily, RF had unpruned and fully grown trees and were not sensitive to the feature order as in DTs. Also, each tree in an RF learns using random sampling, and at each node, a random set of features were considered for splitting. This random sampling and splitting introduced diversity among trees in a forest. After randomly selecting features, RF built a number of regression trees and averaged (aka bagging) them. With enough trees, combinations of randomly selected features and averaging (aka voting), RF emulators reduced the

variance of predictions and deter the overfitting. Resultantly, their performances were best among all averaging ensemble emulators.

Among boosting methods, DT-based AdaB outperformed AdaB and GBM because it combined DT and boosting estimators to predict QoIs. In this study, the DT-based AdaB used 100 trees as a base estimator to build DT-based AdaB emulator. Two base estimators enhanced the confidence on QoI predictions; resultantly, the DT-based AdaB emulator scored better than other two boosting approaches. Based on the ML analyses presented in Sec. 3, linear and Bayesian ML emulators (e.g., NB, BR, GP) are a poor choice to classify and predict reactive-transport QoIs. Overall, RF, DT-based AdaB, GBM, and MLP emulators accurately predicted unseen realizations with average accuracies $>90\%$. From the computational-cost perspective, generalized linear and Bayesian ML emulators were faster to train than ensemble and MLP emulators. Among ensemble and boosting methods, RF and GBM emulators took longest to train. Also, MLP emulators were more expensive to develop than other ML emulators. However, ensemble and MLP emulators took $1/100,000^{\text{th}}$ of the time required for a high-fidelity simulation to predict equivalent QoIs.

Conclusions

The primary purpose was to accurately understand reactive-transport state and expedite predictions of species concentration (QoIs) due to reactive mixing. A suite of linear, Bayesian, ensemble, and MLP ML emulators were compared to classify the state of reactive mixing and to predict species concentrations. All ML emulators were developed based on high-fidelity numerical simulation datasets. A total of 2,315 simulations were carried out to generate data to train and test the emulators. Data were generated by solving the anisotropic reaction-diffusion equations using the non-negative finite element method. Because of the highly nonlinear reactive-transport system, linear and Bayesian (except GP) ML emulators performed poorly

in classifying and predicting the state of reactive mixing (e.g., $R^2 \approx 70\%$). Among Bayesian ML emulators, GP showed promise for accurate prediction of QoIs for small datasets. On the other hand, ensemble and MLP emulators accurately classified the state of reactive-transport and predicted associated QoIs. For example, RF and MLP emulators classified the state of reactive-transport with an accuracy of $>90\%$. Moreover, they predicted the progress of reactive-transport with an accuracy of $>95\%$ on training, testing, and unseen data. Among bagging ensemble methods, RF emulators provided comparatively better predictions than bagging and DT emulators. Similarly, among boosting ensemble methods, DT-based AdaBoost emulators provided better predictions than AdaBoost and GBM emulators. Computationally, for QoI predictions, ML emulators were approximately 10^5 faster than a high-fidelity numerical simulation. Finally, ensemble ML and MLP emulators proved good classifiers and predictors for interrogating the progress of reactive mixing. Looking to the future, ensemble ML and MLP emulators will be validated on both reservoir-scale field and simulation data.

Table 4.6: Performance metrics of ML emulators on training and test datasets for classifying the mixing state (i.e., degree of mixing) of the reaction-diffusion system.

Emulator	Training size (%)	Testing size (%)	Training score (%)	Testing score (%)	Training time (s)
LogR	0.9	99	75	75	31
	63	30	75	75	138
	81	10	75	75	174
LDA	0.9	99	72	72	28
	63	30	72	72	93
	81	10	72	72	102
QDA	0.9	99	77	77	66
	63	30	77	77	128
	81	10	77	77	133
RF	0.9	99	100	98	6,527
	63	30	100	99	22,161
	81	10	100	99	24,015
MLP	0.9	99	97	96	3,384
	63	30	99	99	50,397
	81	10	99	99	66,381

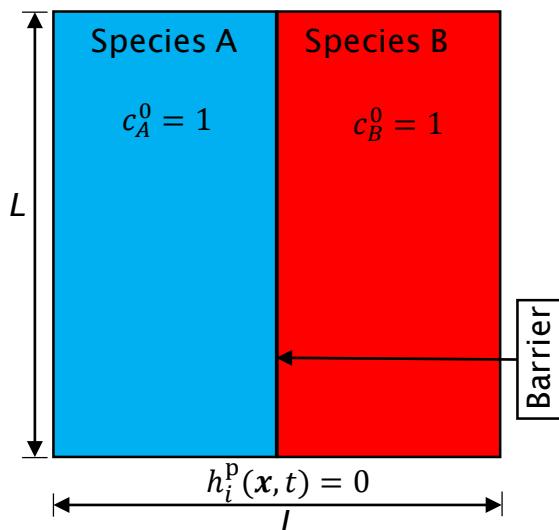


Figure 4.1: Schematic of the initial boundary value problem. L , $h_i^p(\mathbf{x}, t)$, c_A^0 , and c_B^0 are the length of the domain, diffusive flux on the boundary for i^{th} chemical species, initial concentration of species A , and initial concentration of species B , respectively. Species A and B were initially on the left and right sides of the domain, respectively. Initial concentrations of A and B were 1.0 and mixing commenced for $t > 0$.

Table 4.7: Performance metrics of linear and Bayesian ML emulators (regressors). Note, GP and KR failed to converge even on 1% of training data because of a memory leak due to storage of a dense matrix.

Emulator	Training size (%)	Testing size (%)	Training score (%)	Testing score (%)	Training time (s)
LSQR	0.9	99	69	69	12
	63	30	69	69	52
	81	10	69	69	57
RR	0.9	99	69	69	10
	63	30	69	69	42
	81	10	69	69	50
LR	0.9	99	69	69	95
	63	30	69	69	330
	81	10	69	69	368
ER	0.9	99	69	69	121
	63	30	69	69	1,077
	81	10	69	69	1,227
HR	0.9	99	69	69	14
	63	30	69	69	185
	81	10	69	69	195
Polynomial	0.9	99	89	89	79
	63	30	89	89	143
	81	10	89	89	164
BR	0.9	99	69	69	12
	63	30	69	69	62
	81	10	69	69	69
NB	0.9	99	73	73	69
	63	30	73	73	73
	81	10	73	73	91

Table 4.8: Performance metrics of ensemble and MLP emulators.

Emulator	Training size (%)	Testing size (%)	Training score (%)	Testing score (%)	Training time (s)
DT	0.9	99	100	99	42
	63	30	99	99	100
	81	10	99	99	110
Bagging	0.9	99	98	95	42
	63	30	98	95	110
	81	10	98	95	100
RF	0.9	99	100	99	1,435
	63	30	100	99	5,468
	81	10	100	99	6,044
AdaB	0.9	99	90	90	72
	63	30	89	89	1,378
	81	10	89	89	1,585
DT-based AdaB	0.9	99	99	99	103
	63	30	99	99	1,648
	81	10	99	99	1,778
GBM	0.9	99	98	98	133
	63	30	98	98	1,533
	81	10	98	98	2,048
MLP	0.9	99	99	99	688
	63	30	99	99	4,678
	81	10	99	99	9,691

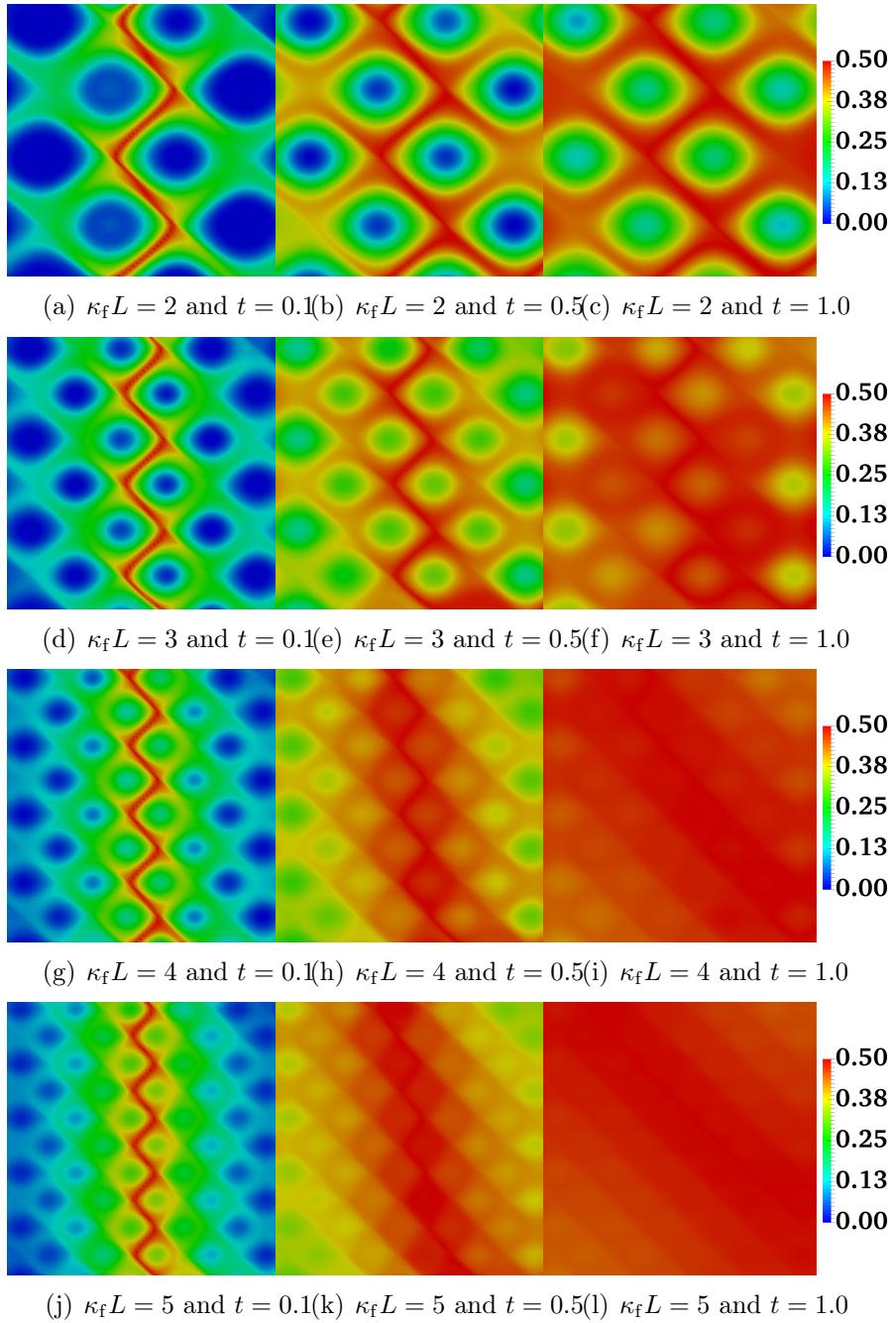


Figure 4.2: Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 10^3$ (high anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. Increased $\kappa_f L$ increases C production, especially at later times.

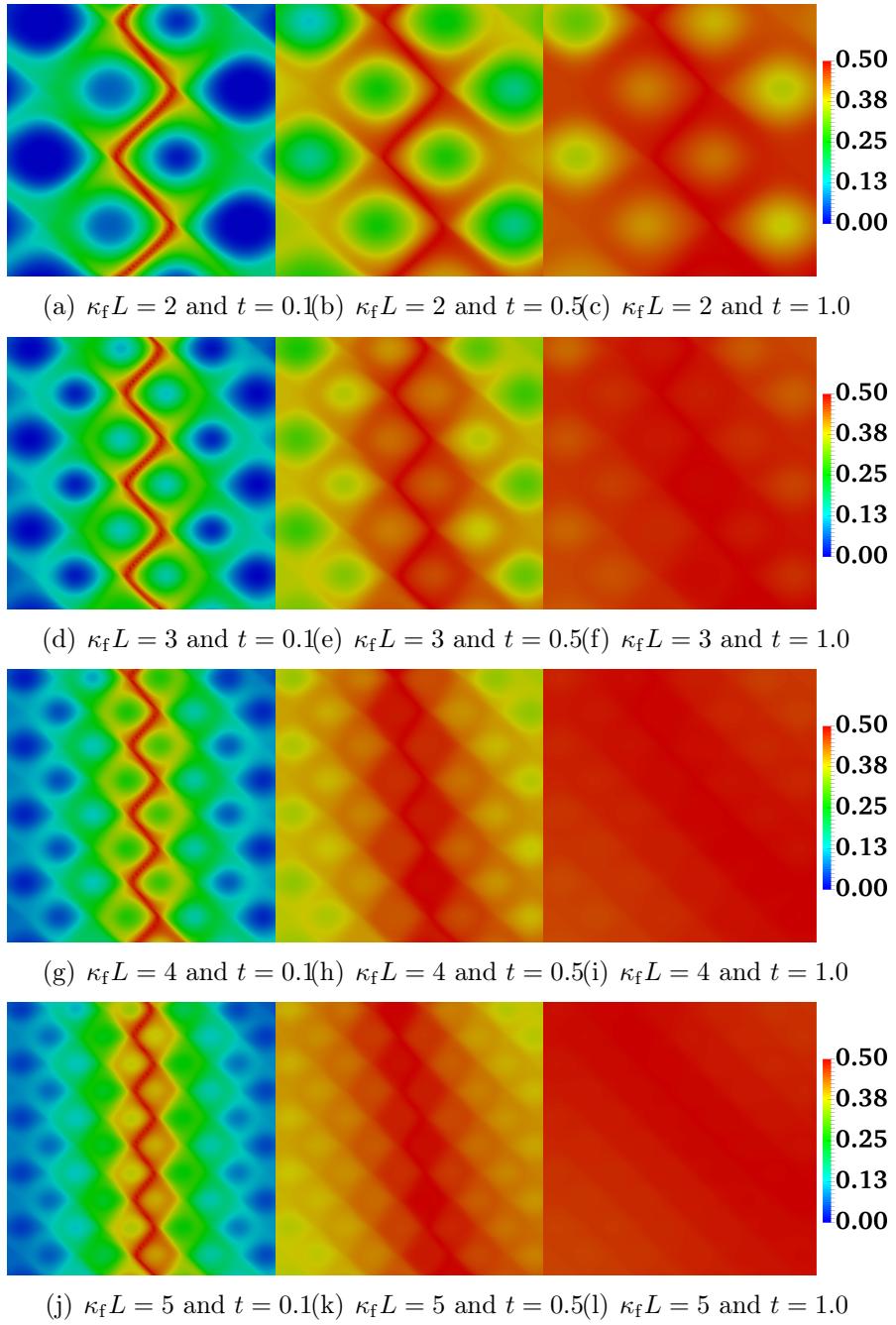


Figure 4.3: Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 100$ (medium anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. Lower anisotropy increased C production than higher anisotropy in Fig. 4.2.

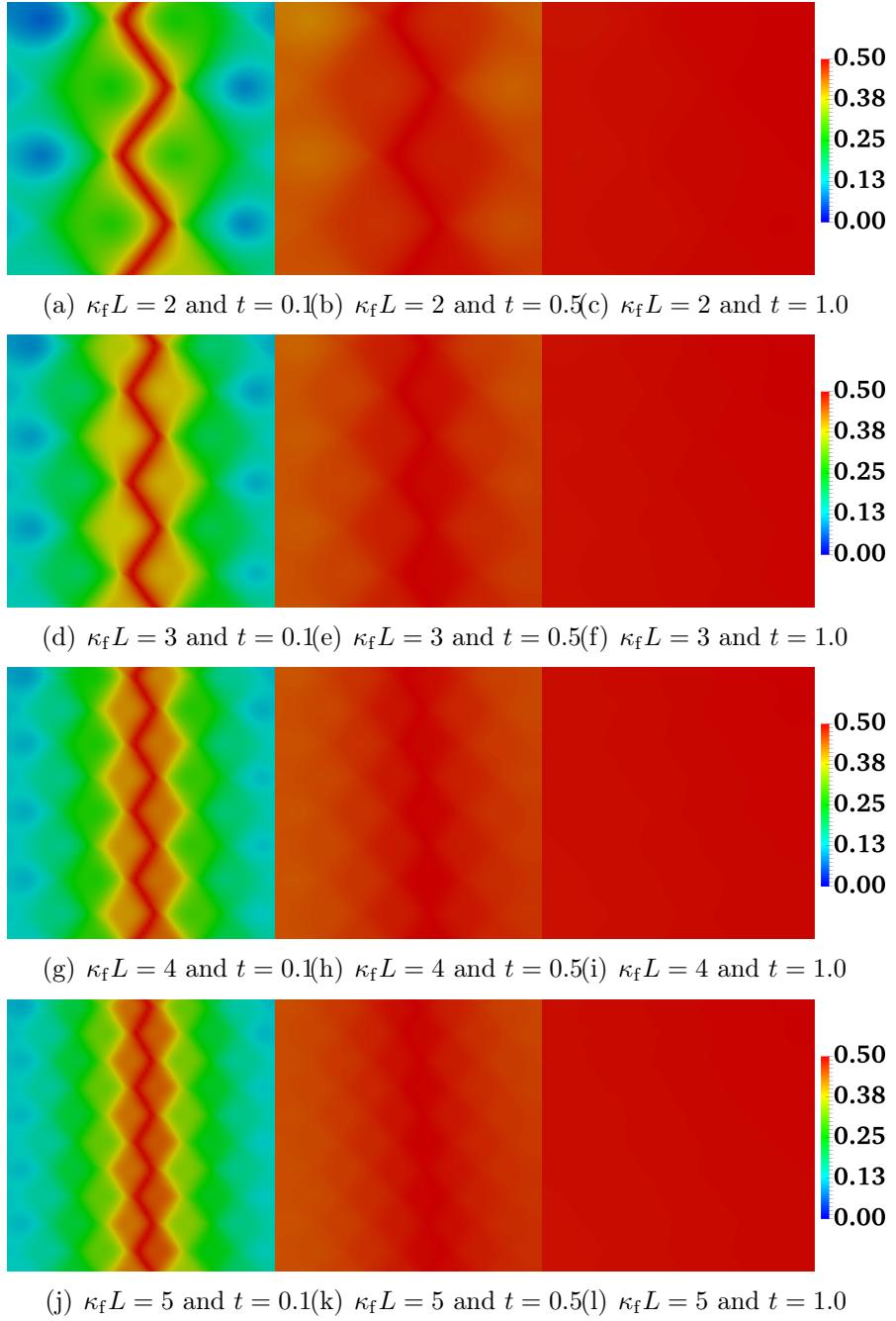


Figure 4.4: Concentration of product C at times $t = 0.1, 0.5$, and 1.0 . Other input parameters were $\frac{\alpha_L}{\alpha_T} = 10$ (low anisotropy), $v_0 = 1$, $T = 0.1$, and $D_m = 10^{-3}$. At low anisotropy, production of C increased. During late times (e.g., $t = 0.5$ and 1.0), diffusion dominates C production while $\kappa_f L$ and $\frac{\alpha_L}{\alpha_T}$ minimally affect C production.

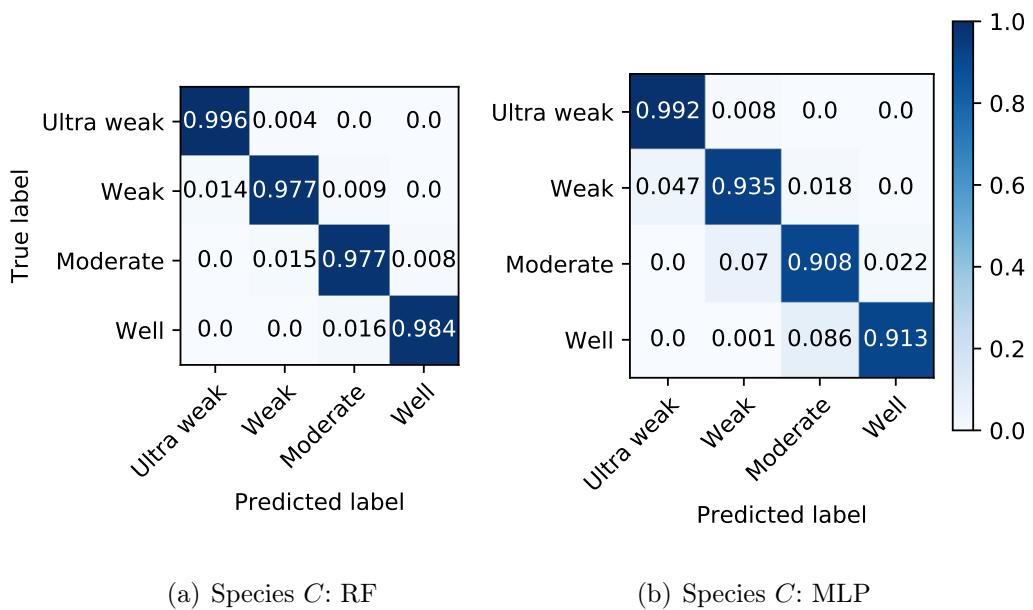


Figure 4.5: Confusion matrices classifying the degree of mixing for the RF (left) and MLP (right) emulators.

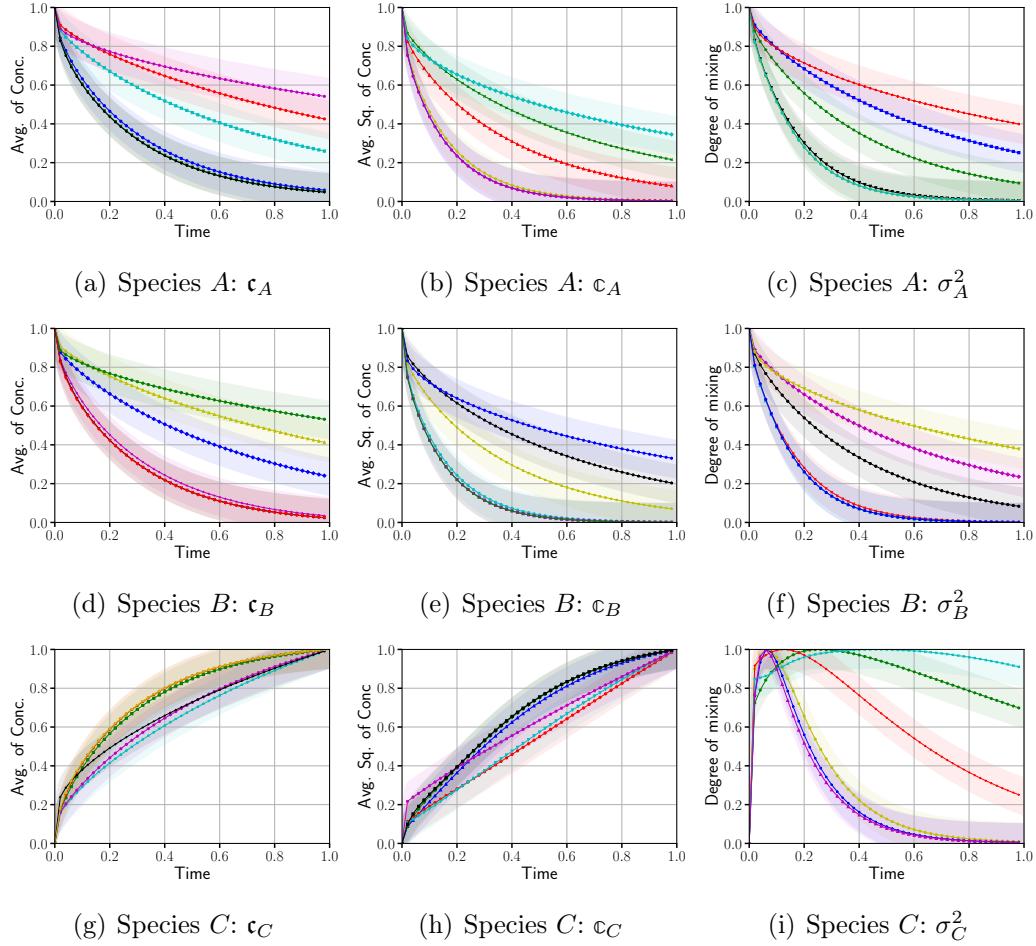


Figure 4.6: This figure shows the true (markers) and RF emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A; (d)–(f) of species B, and (g)–(i) of species C.

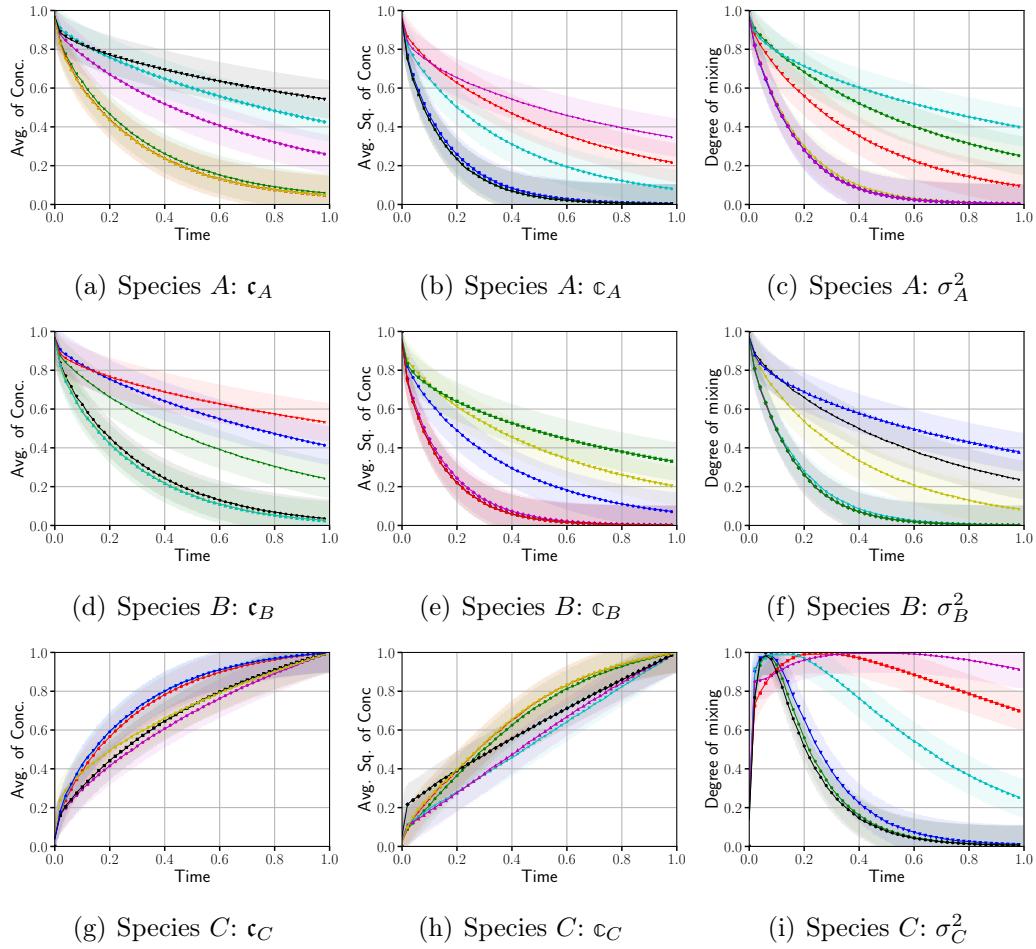


Figure 4.7: This figure shows the true (markers) and GBM emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A; (d)–(f) of species B, and (g)–(i) of species C.

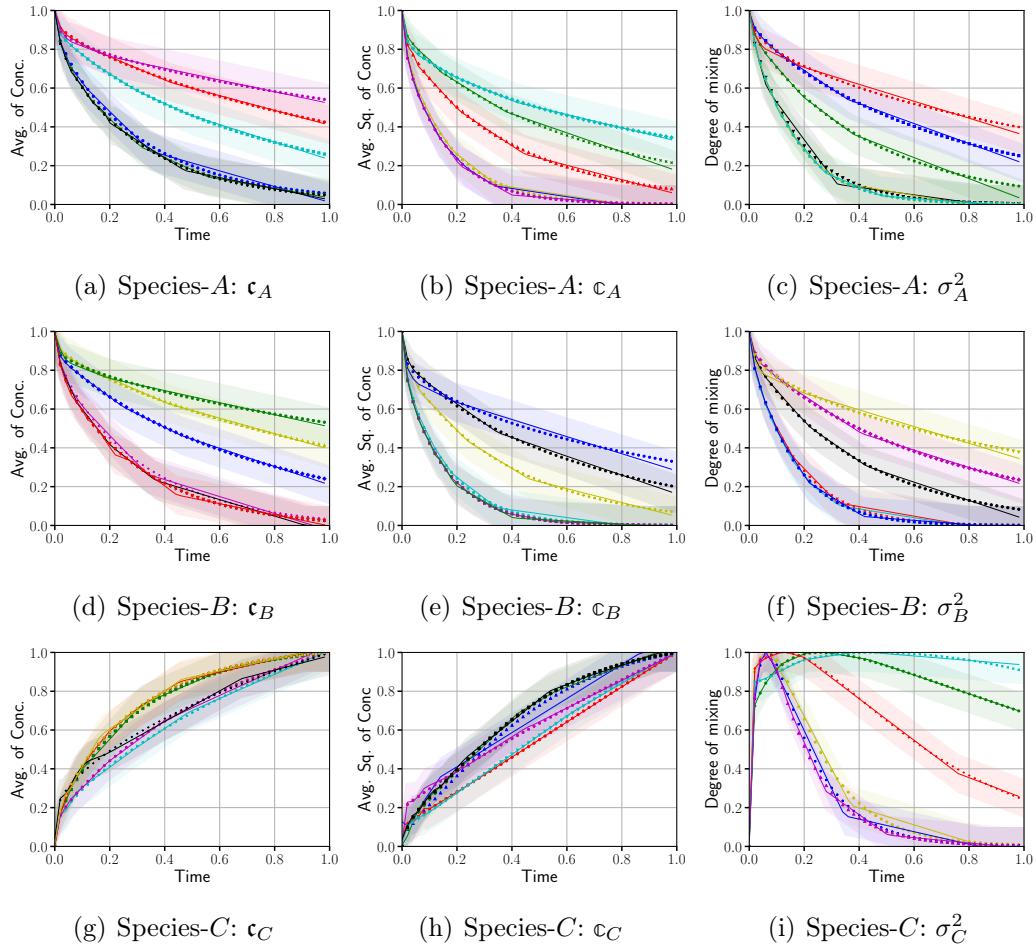


Figure 4.8: This figure shows the true (markers) and ANN emulator predictions (solid curves) of average concentrations, squared of average concentrations, and degree of mixing (a)–(c) of species A ; (d)–(f) of species B , and (g)–(i) of species C .

CHAPTER FIVE

Conclusions

The primary purpose of this dissertation was to demonstrate the importance of numerical modeling, uncertainty analyses, and machine learning for decision support in the Geosciences. To this end, I began by developing a numerical model of a chemical waste site followed by a thorough analysis of uncertainties in model parameters and predictions. This work suggested how NWIRP site characterization could have been improved. Next, I coupled a flow and reactive-transport code, PFLOTRAN and a electrical resistivity code, E4D to simulated flow, reactive-transport and SIP process at different frequencies. This code can be used to simulate remediation activities for chemical, nuclear, green, and bio waste sites. Finally, I labeled reactive-transport model outputs and used them in 20 ML models to predict simulation outcomes. Machine learning study showed that not all ML models were accurate but a few of them were extremely accurate and fast when predicting reactive-transport outcomes.

In the first paper, I developed a groundwater model of the NWIRP site using MODFLOW and particle tracking model using MODPATH. Based on MODFLOW and MODPATH calibration and model interrogation, I quantified parameter and predictive uncertainties, parameter identifiabilities, and observation worth for both existing and hypothetical monitoring wells. Parameter uncertainties were reduced up to 92% and 36 of 156 parameters exceeded a 10% reduction when constrained by the calibration data set. Seven parameters were highly identifiable (>0.5) while 12 parameters had identifiabilities between 0.1 and 0.5. Travel-time uncertainties were reduced up to 92%. Further, pre-calibration travel-time uncertainties were reduced by $>50\%$ for two particles released at site M and between 5 and 40% for the remaining sites. An observation-worth analysis showed that the existing monitoring well network does not

strongly constrain travel times. However, targeted data collection could reduce travel-time uncertainties for all particles by factors from 1.04 to 4.3. Using this information, decision maker could more effectively characterize the site with reduced budget and resources. Also, this study predicted that conservative tracers exited the flow system through the more conductive upper layer within a year, which agreed with site measurements. Looking to the future, a transient numerical-modeling study would reveal a better understanding of the system. The second paper described a code, which coupled PFLOTTRAN and E4D to model electrical-impedance and phase-shift data for SIP due to changes in subsurface characteristics. PFLOTTRAN and E4D are massively parallel codes that simulate fluid flow, reactive transport, and SIP. A mathematical relationship based on Archie's Law and the Cole-Cole model linked flow and solute-transport state variables at various frequencies. A reservoir-scale tracer transport model demonstrated the capability of the newly developed PFLOTTRAN-SIP framework, which clearly showed evolution of flow, reactive transport, and SIP processes over one year. The process started by solving flow and reactive-transport model for tracers. Next, electrical potentials for various electrode configurations at different frequencies were simulated. These simulations revealed that contrast in real potential was minimal even as the frequency varied. However, complex potential showed variable responses across frequencies. Additionally, phase shift (combination of real and complex potentials) helped identify regions where tracer concentrations were high. Combining frequency-dependent real potential, complex potential, and phase responses from an SIP survey/simulation paints a more detailed picture of the subsurface with an enhanced ability to detect contaminants/tracers. Moreover, coupling fluid flow, reactive transport, and SIP models can supports improved detection of contaminants compared to either the ERT or SIP method alone because flow and reactive-transport simulations capture tracer evolution. For instance, through a numerical example, solute transport simulations provided insight into the tracer distribution. This information was used

to customize SIP inversion to estimate frequency-dependent electrical conductivities, which yielded an improved image of tracer concentrations. This work highlighted the benefit of PFLOTTRAN-SIP to detect tracers and it can be easily extended to detect hydrocarbon flow, changes in the subsurface due to geochemical reactions, sulfide minerals, metallic objects, municipal wastes, and salinity intrusion.

The third paper studied which ML models are best for replicating reactive-transport outcomes in subsurface porous media. This study developed 20 ML models to emulate reactive-transport state and to forecast species concentrations (QoIs) due to reactive transport. These ML models included a suite of linear, Bayesian, ensemble, and MLP ML models that were developed based on high-fidelity numerical simulation datasets. A total of 2,315 simulations were carried out to generate data to train and test the models by solving the anisotropic reaction-diffusion equations using the non-negative finite element method. Because of the highly nonlinear reactive-transport system, linear and Bayesian (except GP) ML models performed poorly when classifying and predicting the state of reactive transport (e.g., $R^2 \approx 70\%$). Among Bayesian ML models, GP showed promise for accurate prediction of QoIs for small datasets. On the other hand, ensemble and MLP models accurately classified the state of reactive-transport and predicted associated QoIs. Specifically, RF and MLP models classified the state of reactive-transport with an accuracy of $>90\%$. Moreover, they predicted the progress of reactive-transport with an accuracy of $>95\%$ on training, testing, and unseen data. Among bagging ensemble methods, RF models issued comparatively better predictions than bagging and DT models. Similarly, among boosting ensemble methods, DT-based AdaBoost models provided better predictions than AdaBoost and GBM models. Computationally, for QoI predictions, ML models were approximately 10^5 faster than a high-fidelity numerical simulation. Finally, ensemble ML and MLP models proved good classifiers and predictors for interrogating the progress of reactive

transport. Looking to the future, ensemble ML and MLP emulators will be validated on both reservoir-scale field and simulation data.

APPENDICES

APPENDIX A
NWIRP

Table A.1: Locations and measurement of hydraulic conductivity (hk).

Well	x (m)	y (m)	hk (m/s)	Well	x (m)	y (m)	hk (m/s)
K_1	9461	3499	1×10^{-5}	K_{51}	8192	7845	2×10^{-7}
K_2	2522	3864	3×10^{-6}	K_{52}	10613	4765	2×10^{-5}
K_3	5572	6719	3×10^{-6}	K_{53}	5622	6770	1×10^{-8}
K_4	7602	6134	7×10^{-6}	K_{54}	6807	6080	6×10^{-6}
K_5	9997	4928	5×10^{-6}	K_{55}	8366	7603	4×10^{-8}
K_6	6325	6121	4×10^{-6}	K_{56}	3430	6566	1×10^{-5}
K_7	2264	4547	4×10^{-6}	K_{57}	2158	4522	2×10^{-6}
K_8	6426	6810	4×10^{-7}	K_{58}	7971	6755	2×10^{-6}
K_9	4682	6375	1×10^{-9}	K_{59}	2000	4080	4×10^{-6}
K_{10}	6387	7868	6×10^{-8}	K_{60}	2260	4063	1×10^{-5}
K_{11}	6767	6108	5×10^{-7}	K_{61}	2430	3445	1×10^{-7}
K_{12}	9788	6078	2×10^{-6}	K_{62}	3547	3221	1×10^{-8}
K_{13}	1996	5308	2×10^{-6}	K_{63}	7072	8002	4×10^{-7}
K_{14}	10322	4829	2×10^{-7}	K_{64}	5067	7610	2×10^{-6}
K_{15}	7789	5298	8×10^{-8}	K_{65}	2328	4154	1×10^{-8}
K_{16}	10371	4824	3×10^{-6}	K_{66}	3550	4842	1×10^{-8}
K_{17}	2857	3932	1×10^{-6}	K_{67}	4460	6441	1×10^{-8}
K_{18}	10423	4895	7×10^{-8}	K_{68}	11344	4146	1×10^{-8}
K_{19}	10429	4933	1×10^{-5}	K_{69}	1804	3916	3×10^{-8}
K_{20}	2432	4206	5×10^{-7}	K_{70}	9477	7319	5×10^{-6}
K_{21}	4901	6201	9×10^{-9}	K_{71}	6457	5976	8×10^{-5}

K_{22}	10943	4806	4×10^{-7}	K_{72}	10544	4756	2×10^{-8}
K_{23}	5633	7727	2×10^{-5}	K_{73}	6832	6307	1×10^{-9}
K_{24}	9115	5669	1×10^{-8}	K_{74}	1659	3818	4×10^{-6}
K_{25}	10347	5043	5×10^{-7}	K_{75}	2398	4053	1×10^{-9}
K_{26}	10396	4947	2×10^{-7}	K_{76}	8770	4813	2×10^{-8}
K_{27}	6406	6231	1×10^{-8}	K_{77}	2328	4389	1×10^{-4}
K_{28}	8054	8068	3×10^{-8}	K_{78}	7806	8198	3×10^{-9}
K_{29}	1509	3145	4×10^{-6}	K_{79}	2175	3788	1×10^{-7}
K_{30}	10554	4908	1×10^{-7}	K_{80}	9897	4600	4×10^{-9}
K_{31}	10201	4997	1×10^{-5}	K_{81}	5625	6712	2×10^{-6}
K_{32}	2106	4628	2×10^{-6}	K_{82}	2430	4132	2×10^{-6}
K_{33}	1967	4532	2×10^{-6}	K_{83}	5460	4628	6×10^{-7}
K_{34}	6808	6097	3×10^{-7}	K_{84}	7239	6867	5×10^{-7}
K_{35}	6754	6498	1×10^{-6}	K_{85}	4037	6562	3×10^{-6}
K_{36}	4668	5910	8×10^{-6}	K_{86}	4365	5671	3×10^{-8}
K_{37}	4082	6038	2×10^{-6}	K_{87}	4662	3360	2×10^{-7}
K_{38}	4863	5076	4×10^{-7}	K_{88}	4659	4076	1×10^{-6}
K_{39}	4787	5128	9×10^{-7}	K_{89}	2560	4571	9×10^{-6}
K_{40}	6357	6207	2×10^{-7}	K_{90}	11249	5943	4×10^{-6}
K_{41}	2119	4505	5×10^{-7}	K_{91}	6525	5924	2×10^{-7}
K_{42}	6970	7814	4×10^{-7}	K_{92}	3556	3791	1×10^{-8}
K_{43}	7693	5727	1×10^{-7}	K_{93}	11602	5230	4×10^{-6}
K_{44}	5722	3315	1×10^{-8}	K_{94}	10419	3715	8×10^{-6}
K_{45}	10649	6319	2×10^{-9}	K_{95}	7194	6002	1×10^{-10}
K_{46}	2206	3029	3×10^{-7}	K_{96}	10606	4771	8×10^{-6}
K_{47}	10472	4744	8×10^{-8}	K_{97}	2455	3391	4×10^{-8}
K_{48}	6290	6239	6×10^{-9}	K_{98}	11304	4914	3×10^{-6}

K_{49}	2239	4120	8×10^{-7}	K_{99}	10440	5531	6×10^{-7}
----------	------	------	--------------------	----------	-------	------	--------------------

Table A.2: Locations and average heads of observations.

Observation	x (m)	y (m)	Head (m)	Observation	x (m)	y (m)	Head (m)
MW3-1	6422	6072	233.65	GAF-2	6832	6306	232.26
MW3-2	6407	6230	232.66	GAF-3	6940	6124	233.09
MW3-3	6357	6206	232.30	GAF-4	6749	5949	235.23
MW3-4	6346	6173	232.44	GAF-5	6577	6027	234.64
MW3-5	6456	6142	233.47	GAF-6	6512	6045	234.52
MW3-6	6458	5975	234.12	GAF-7	6426	6809	230.17
MW3-7	6326	6120	232.83	GAF-8	6755	6498	230.02
MW3-8	6302	6177	232.31	GAF-9	6835	6673	229.55
MW3-9	6291	6238	232.29	AF0MW01	6668	6135	235.35
MW3-10	6526	5924	234.80	AF0MW02	6539	6054	234.65
MW3-11	6315	6064	233.03	AF0MW03	6543	6012	234.63
MW3-12	6513	6283	232.93	AF0MW04	6439	5940	234.54
MW3-16	6409	6307	231.74	MW1-3	10396	4946	210.89
MW8-1	6768	6108	234.22	MW2-1	2001	4079	245.60
MW8-2	6801	6102	234.03	MW2-4	1804	3916	241.10
MW8-3	6808	6097	234.03	MW4-3	5587	6757	230.89
MW8-4	6807	6079	233.70	MW5-1	2397	4190	243.69
MW8-5	6802	6151	233.70	MW5-2	2239	4119	247.10
MW8-6	6771	6151	233.84	MW7-1	2264	4547	247.19
GAF-1	6685	6433	230.49	MW7-3	2107	4627	246.50
MW3-13	6231	6164	232.65	MW3-14	6228	6263	232.10
MW3-15	6474	6417	231.23				

Table A.3: hk of pilot point (PP).

PP	x	y	hk (m/s)	PP	x	y	hk (m/s)
P_1	9461	-6201	1.00×10^{-5}	P_{51}	8192	-1855.	2.00×10^{-7}
P_2	2522	-5836	3.00×10^{-6}	P_{52}	10613	-4935.	2.00×10^{-5}
P_3	5572	-2981	3.00×10^{-6}	P_{53}	5622	-2930	1.00×10^{-8}
P_4	7602	-3566	7.00×10^{-6}	P_{54}	6807	-3620	6.00×10^{-6}
P_5	9997	-4772	5.00×10^{-6}	P_{55}	8366	-2097	4.00×10^{-8}
P_6	6325	-3579	4.00×10^{-6}	P_{56}	3430	-3134	1.00×10^{-5}
P_7	2264	-5153	4.00×10^{-6}	P_{57}	2158	-5178	2.00×10^{-6}
P_8	6426	-2890	4.00×10^{-7}	P_{58}	7971	-2945	2.00×10^{-6}
P_9	4682	-3325	1.00×10^{-9}	P_{59}	2000	-5620.	4.00×10^{-6}
P_{10}	6387	-1832	6.00×10^{-8}	P_{60}	2260	-5637.	1.00×10^{-5}
P_{11}	6767	-3592	5.00×10^{-7}	P_{61}	2430	-6255.	1.00×10^{-7}
P_{12}	9788	-3622	2.00×10^{-6}	P_{62}	3547	-6479.	1.00×10^{-8}
P_{13}	1996	-4392	2.00×10^{-6}	P_{63}	7072	-1698.	4.00×10^{-7}
P_{14}	10322	-4871	2.00×10^{-7}	P_{64}	5067	-2090.	2.00×10^{-6}
P_{15}	7789.	-4402	8.00×10^{-8}	P_{65}	2328	-5546.	1.00×10^{-8}
P_{16}	10371	-4876	3.00×10^{-6}	P_{66}	3550	-4858.	1.00×10^{-8}
P_{17}	2857	-5768	1.00×10^{-6}	P_{67}	4460	-3259.	1.00×10^{-8}
P_{18}	10423	-4805	7.00×10^{-8}	P_{68}	11344	-5554	1.00×10^{-8}
P_{19}	10429	-4767	1.00×10^{-5}	P_{69}	1804	-5784.0	3.00×10^{-8}
P_{20}	2432	-5494	5.00×10^{-7}	P_{70}	9477	-2381.	5.00×10^{-6}
P_{21}	4901	-3499	9.00×10^{-9}	P_{71}	6457	-3724.	8.00×10^{-5}
P_{22}	10943	-4894	4.00×10^{-7}	P_{72}	10544	-4944	2.00×10^{-8}
P_{23}	5633	-1973	2.00×10^{-5}	P_{73}	6832	-3393	1.00×10^{-9}
P_{24}	9115	-4031	1.00×10^{-8}	P_{74}	1659	-5882	4.00×10^{-6}

P_{25}	10347	-4657	5.00×10^{-7}	P_{75}	2398	-5647	1.00×10^{-9}
P_{26}	10396	-4753	2.00×10^{-7}	P_{76}	8770	-4887	2.00×10^{-8}
P_{27}	6406	-3469	1.00×10^{-8}	P_{77}	2328	-5311	1.00×10^{-4}
P_{28}	8054	-1632	3.00×10^{-8}	P_{78}	7806	-1502	3.00×10^{-9}
P_{29}	1509	-6555	4.00×10^{-6}	P_{79}	2175	-5912	1.00×10^{-7}
P_{30}	10554	-4792	1.00×10^{-7}	P_{80}	9897	-5100	4.00×10^{-9}
P_{31}	10201	-4703	1.00×10^{-5}	P_{81}	5625	-2988	2.00×10^{-6}
P_{32}	2106	-5072	2.00×10^{-6}	P_{82}	2430	-5568	2.00×10^{-6}
P_{33}	1967	-5168	2.00×10^{-6}	P_{83}	5460	-5072	6.00×10^{-7}
P_{34}	6808	-3603	3.00×10^{-7}	P_{84}	7239	-2833	5.00×10^{-7}
P_{35}	6754	-3202	1.00×10^{-6}	P_{85}	4037	-3138	3.00×10^{-6}
P_{36}	4668	-3790	8.00×10^{-6}	P_{86}	4365	-4029	3.00×10^{-8}
P_{37}	4082	-3662	2.00×10^{-6}	P_{87}	4662	-6340	2.00×10^{-7}
P_{38}	4863	-4624	4.00×10^{-7}	P_{88}	4659	-5624	1.00×10^{-6}
P_{39}	4787	-4572	9.00×10^{-7}	P_{89}	2560	-5129	9.00×10^{-6}
P_{40}	6357	-3493	2.00×10^{-7}	P_{90}	11249	-3757	4.00×10^{-6}
P_{41}	2119	-5195	5.00×10^{-7}	P_{91}	6525	-3776	2.00×10^{-7}
P_{42}	6970	-1886	4.00×10^{-7}	P_{92}	3556	-5909	1.00×10^{-8}
P_{43}	7693	-3973	1.00×10^{-7}	P_{93}	11602	-4470	4.00×10^{-6}
P_{44}	5722	-6385	1.00×10^{-8}	P_{94}	10419	-5985	8.00×10^{-6}
P_{45}	10649	-3381	2.00×10^{-9}	P_{95}	7194	-3698	1.00×10^{-10}
P_{46}	2206	-6671	3.00×10^{-7}	P_{96}	10606	-4929	8.00×10^{-6}
P_{47}	10472	-4956	8.00×10^{-8}	P_{97}	2455	-6309	4.00×10^{-8}
P_{48}	6290	-3461.	6.00×10^{-9}	P_{98}	11304	-4786	3.00×10^{-6}
P_{49}	2239	-5580.	8.00×10^{-7}	P_{99}	10440	-4169	6.00×10^{-7}
P_{50}	4541	-2193.	3.00×10^{-8}				

APPENDIX B
Manuscript: ML to reactive-transport data

Variables

A = Diagonal matrix

a_n^l = Activation function of neuron n at layer l

b = Bias

c_i [mol m⁻³] = The molar concentration of chemical species i

$c_i^0(\mathbf{x})$ = The initial concentration of chemical species i

$c_i^p(\mathbf{x}, t)$ [mol m⁻³] = Prescribed molar concentration

$\mathbf{D}(\mathbf{x}, t)$ [s² m⁻¹] = The anisotropic dispersion tensor

D_m [m³ s⁻²] = The molecular diffusivity

det = Determinant

\mathbb{E} = Expectation

F = Activation function

f = Function

f_m = m^{th} classifier

G = Impurity

H = Gini impurity function

H_e = Truncation value for Huber loss

h_m = Base learner/tree

$h_i^p(\mathbf{x}, t)$ [m s⁻¹] = Flux

i = Index

I = The identity tensor

inf = The greatest upper bound

$J = \sup |\hat{y}_i - y_i|$

j = Feature index

k = Class variable index

k_{AB} [m^{-1}] = The bi-linear reaction rate coefficient

L = Loss function

L_1 = L_1 norm

L_2 = L_2 norm

l = Layer index

l = The length of the ‘wiggles’ in sine function

M = Number of training data

m = Tree node index

N = Gaussian or normal distribution

n = Node number

n_A = Stoichiometric coefficient for species A

n_B = Stoichiometric coefficient for species B

n_C = Stoichiometric coefficient for species C

o = Random coefficient

p [%] = Probability

p = The distance between repetitions of the sine function

p_{mk} [%] = Probability at mk^{th} leaf

Q = Data at tree node m

q = Batch or subsample

r = Random coefficient

s = Split

sup = The least upper bound

t_m = Threshold at which trees split

u = Random coefficient

\mathbf{v} [m s^{-1}] = The velocity vector field

v_0 = The perturbation parameter

w = Coefficient

\mathbf{w} = Coefficient vector

w_0 = Intercept

\mathbb{X} = Feature matrix

\mathbf{x} = Feature vector

y = Label

\mathbf{y} = Label vector

\hat{y} = Approximation to y

z = The dummy variable

Greek Symbols

α = Penalty/regularization parameter

α_1 = Regularization parameter for w

α_2 = Regularization parameter for w

α_L [$m^2 s^{-1}$] = The longitudinal diffusivity

α_T [$m^2 s^{-1}$] = The transverse diffusivity

β = Regularization parameter

Γ_i^D = Dirichlet boundary condition

Γ_i^N = Neumann boundary condition

θ_m = Confidence in the prediction for m^{th} data

$\kappa_f L$ and T [-] = The characteristic spatial and temporal scales of the flow field

γ = Learning rate

ϵ = Truncation value under which no penalty is associated with the training loss

η = Noise

θ = Confidence function for prediction

\mathcal{K} = Node number

\mathcal{K} = Kernel

λ = Spread of kernel

μ = Mean

ν = Integer

$\pi = \cos^{-1}(-1)$

Σ = The covariance matrix

Ψ = Vector of $\mathcal{N} \times 1$ size

ω = Regularization parameter

\mathbb{I} = Identity matrix

\mathfrak{T} = Number of regression tree

Φ = Design matrix of $\mathcal{N} \times (\mathcal{N} + 1)$ size

ω = Regularization parameter

\otimes = The tensor product

$\mathbb{1}$ = Indicator function (either 1 or 0)

Linear Emulators

Generalized Linear Emulators

Suppose there are n features x_1 through x_n that correspond to a label y . LSQR calculates the closest \hat{y} by finding the best linear combination of features as:

$$\hat{y}(\mathbf{w}, \mathbf{x}) = w_0 + w_1x_1 + \cdots + w_nx_n = \mathbf{x} \cdot \mathbf{w}. \quad (\text{B.1})$$

Linear regressors minimize a loss (or cost) function. The cost/loss function in this case is the residual sum of squares between a set of training feature vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ and predicted targets y_1, y_2, \dots, y_m of the form:

$$L_{\text{lin}} = \min_{\mathbf{w}} \|\mathbf{x}\mathbf{w} - \mathbf{y}\|_2^2 + \alpha_1 \|\mathbf{w}\|_1 + \alpha_2 \|\mathbf{w}\|_2^2 + \Sigma \sum_{i=1}^m \left[1 + H_\epsilon \left(\frac{\mathbf{x}_i \cdot \mathbf{w} - y_i}{\Sigma} \right) \right]. \quad (\text{B.2})$$

For LSQR, $\alpha_1 = \alpha_2 = \Sigma = 0$, for RR, $\alpha_1 = \Sigma = 0$, for LR, $\alpha_2 = \Sigma = 0$, for ER, $\Sigma = 0$, and for HR, $\alpha_2 = 0$.

$$H_\epsilon(e) = \begin{cases} e^2 & \text{if } e < \epsilon \\ 2\epsilon|e| - \epsilon^2 & \text{otherwise} \end{cases}. \quad (\text{B.3})$$

The LSQR method minimizes L_{lin} without regularization. RR uses the L_2 norm, which does not use sparsity constraints. However, it includes a penalty α_2 to weights, which is known as the ridge coefficient. This prevents weights from getting too large as well as overfitting. LR is another linear regressor that penalizes the L_1 norm. This penalty α_1 on the absolute value of weights results in sparse models tending toward small weights. The α_1 controls the strength of the regularization penalty, and more parameters are eliminated with increasing α_1 . With increasing α_1 , bias increases, but variances decrease and vice versa. ER is another linear regressor that combines the L_1 and L_2 penalties of RR and LR. It is useful for data with multiple features that are correlated with each other. LR likely picks one of these correlated features at random, but ER picks all the correlated features. HR is a generalized linear regression method that put a sample as an inlier, if the absolute error of that sample is less than the specified threshold. HR puts a sample as an outlier, if absolute errors go beyond the specified threshold. Polynomial regression applies the LSQR formula on quadratic scaled data.

Logistic Regression

Despite its name, logistic regression is a classifier; it uses the linear regression scheme to correlate a probability for each class. Logistic regression predicts the outcome in terms of probability and provides a meaningful threshold at which distinguishing between classes is possible Molnar (2019). Multi-class classification is achieved through either One-vs-One or One-vs-Rest strategy Pedregosa et al. (2011a). A simple linear ML emulator fails to provide multi-class output as probabilities. But

the logistic regression provides the probabilities through the logistic function. Consider an ML model with two features x_1 and x_2 with one label y , which is classified with a probability p . If we assume a linear relationship between predictor variables and the log-odds of the event:

$$\ln \frac{p}{1-p} = w_0 + w_1 x_1 + w_2 x_2. \quad (\text{B.4})$$

With simple algebraic manipulation, the probability p of classifying the predictor variable can be recast as:

$$p = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}. \quad (\text{B.5})$$

Here, the loss function is defined by cross-entropy loss as:

$$L_{\text{cross-entropy}} = -\frac{1}{n} \sum_{n=1}^n [y_n \log(p_n) + (1 - y_n) \log(1 - p_n)]. \quad (\text{B.6})$$

Kernel Ridge (KR) Regression

KR regression combines RR with kernel tricks Murphy (2012) to learn a linear function induced by both the kernel and data. The kernel trick enables a linear ML emulator to learn nonlinear functions without explicitly mapping a linear learning algorithm. The kernel function is applied on each label to map the original nonlinear observations into a higher-dimensional space. In this work, the stationary radial basis function (RBF) kernel is the optimized kernel. The RBF kernel on two different feature vectors, \mathbf{x}_1 and \mathbf{x}_2 , is:

$$\mathcal{K}_{\text{RBF}}(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\lambda \|\mathbf{x}_1 - \mathbf{x}_2\|^2). \quad (\text{B.7})$$

If the kernel is Gaussian then high λ shrinks the spread of Gaussian distribution and vice versa. The squared-loss function is used to learn the linear mapping function:

$$L_{\text{squared}} = (y - \hat{y})^2. \quad (\text{B.8})$$

Bayesian Ridge (BR) Regression

Using Bayes' Rule, BR formulates a probabilistic model of the regression problem. BR assumes labels y as normally distributed around $\mathbb{X}w$ and obtains a probabilistic model by:

$$p(y|\mathbb{X}, w, \beta) = N(y|\mathbb{X}w, \beta). \quad (\text{B.9})$$

The prior for the coefficient vector w is given by a spherical Gaussian distribution:

$$p(w|\omega) = N(w|0, \omega^{-1}\mathbb{I}). \quad (\text{B.10})$$

The β and ω are selected to be conjugate priors and gamma distributions. The parameters β and ω are estimated by maximizing the log-marginal likelihood MacKay (1992); Tipping (2001) as:

$$L_{\text{lml}} = -\frac{1}{2} \left[\log_{10} \|\omega^{-1}\mathbb{I} + \Phi A^{-1} \Phi^\top\| + \Psi^\top (\omega^{-1}\mathbb{I} + \Phi A^{-1} \Phi^\top)^{-1} \Psi \right] + \text{extension}, \quad (\text{B.11})$$

where extension = $\sum_{i=0}^n (o \log \beta_i - r \beta_i) + u \log \omega - w \omega$.

Gaussian Process (GP)

GPs are generic supervised learning methods for prediction and probabilistic classification that use properties inherited from the normal distribution. GP has the capability of using kernel tricks, which differentiate GP from BR. GP emulators are not sparse; as a result they are computational inefficient when developing models in high-dimensional spaces. That is, computing GP emulators are difficult to implement if features exceed a few dozens Rasmussen (2003); Santner, Williams, and Notz (2018) in **Scikit-learn** and our training sample size is $\mathcal{O}(10^5)$. In this study, the RBF kernel (Eq. (B.7)) is used to obtain GP emulators by maximizing the first term on the right of Eq. (B.11) to predict \hat{y} .

Naïve Bayes (NB)

NB emulators are supervised ML methods that also apply Bayes' Theorem with the naïve assumption of conditional independence between every pair of features given the label value Manning, Raghavan, and Schütze (2010); McCallum and Nigam (1998); Metsis, Androutsopoulos, and Palioras (2006); Rennie, Shih, Teevan, and Karger (2003). NB maximizes $p(x_i | y)$ and $p(y)$ by maximizing the a posteriori function Pedregosa et al. (2011a). Various naïve Bayes regressions differ by the assumptions they make regarding the distribution of $p(x_i | y)$ Zhang (2004). Herein, we use Gaussian-naïve Bayes emulator:

$$p(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left[-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right]. \quad (\text{B.12})$$

NB updates model parameters such as feature means and variance using different batch sizes, which makes NB computationally efficient C., G, and L. (1983).

Linear and Quadratic Discriminant Analyses (LDA/QDA)

LDA and QDA are classifiers that use Bayes rule. They compute the class conditional distribution of data $p(\mathbf{x}|y = k)$ for each class k . Based on $p(\mathbf{x}|y = k)$, for partition $y = q$ of sample space, predictions are made using Bayes' rule:

$$p(y = k|\mathbf{x}) = \frac{p(\mathbf{x}|y = k)p(y = k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y = k)p(y = k)}{\sum_q p(\mathbf{x}|y = q)p(y = q)}. \quad (\text{B.13})$$

Later, class k is selected to maximize the conditional probability. Specifically, $p(\mathbf{x}|y)$ is modeled using a multivariate Gaussian distribution with density:

$$p(y = k|\mathbf{x}) = \frac{1}{(2\pi)^{j/2} |\det[\Sigma_k]|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \mu_k \mathbf{1}) \cdot (\Sigma_k)^{-1} (\mathbf{x} - \mu_k \mathbf{1})\right]. \quad (\text{B.14})$$

Using training data, it estimates the class priors $p(y = k)$, class means μ_k , and covariance matrices Σ_k either by the empirical sample class covariance matrices or by a regularized estimator. In LDA, each class shares the same covariance matrix

(i.e., $\Sigma_k = \Sigma$), which leads to linear decision surface Hastie et al. (2009):

$$\begin{aligned} \log \left(\frac{p(y=k|\mathbf{x})}{p(y=q|\mathbf{x})} \right) &= \log \left(\frac{p(\mathbf{x}|y=k)p(y=k)}{p(\mathbf{x}|y=q)p(y=q)} \right) = 0 \iff \\ (\mu_k - \mu_q)\mathbf{1} \cdot \Sigma^{-1}\mathbf{x} &= \frac{1}{2} (\mu_k\mathbf{1} \cdot \Sigma^{-1}\mu_k\mathbf{1} - \mu_q\mathbf{1} \cdot \Sigma^{-1}\mu_q\mathbf{1}) - \log \frac{p(y=k)}{p(y=q)}. \end{aligned} \quad (\text{B.15})$$

However, QDA does not assume covariance matrices of the Gaussian's, which leads to a quadratic decision surface Hastie et al. (2009). Both LDA and QDA use the cross-entropy loss function Eq. (B.6).

Ensemble ML Emulators

Decision Tree (DT)

DT is interpretable as a weak ML classifier and regressor. DTs split leaves in a tree and find the best or optimal split s^* that increases the purity/accuracy of the resulting tree Breiman (2001); Breiman, Friedman, Olshen, and Stone (2017); Galiano et al. (2015); Geurts, Ernst, and Wehenkel (2006). A single tree reduces error in a locally optimal way during feature space splitting while a regression tree minimizes the residual squared error. For n pairs of training samples, the DT recursively partitions the space to bring the same labels under the same group. Let data at node m be represented by Q . For each candidate split $s = (j, \epsilon_m)$ consisting of feature j and threshold ϵ_m , DT splits data into $Q_{\text{left}}(s)$ and $Q_{\text{right}}(s)$ subsets. For regression, the impurity at m is computed using the Gini impurity function $H(\mathbb{X}_m) = \frac{1}{\mathfrak{T}_m} \sum_{i \in \mathfrak{T}_m} (y_i - \hat{y}_i)^2$ using:

$$G(Q, s) = \frac{n_{\text{left}}}{\mathfrak{T}_m} H(Q_{\text{left}}(s)) + \frac{n_{\text{right}}}{\mathfrak{T}_m} H(Q_{\text{right}}(s)), \quad (\text{B.16})$$

where $\mathfrak{T}_m \leq \text{min}_{\text{samples}}$ or $\mathfrak{T}_m = 1$. Then, the DT selects the parameters that minimize the impurity:

$$s^* = \operatorname{argmin}_s G(Q, s). \quad (\text{B.17})$$

DT recursively find $Q_{\text{left}}(s^*)$ and $Q_{\text{right}}(s^*)$ till to reach maximum allowable depth, $\mathfrak{T}_m < \text{min}_{\text{samples}}$ or $\mathfrak{T}_m = 1$. For regression, the loss function, L_{MSE} , is defined as mean

squared error (MSE) between the high-fidelity simulations and the ML emulators:

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (\text{B.18})$$

Bagging Emulator

Bagging is a simple ensemble technique that builds on many independent tree/predictors and combines them using various model averaging techniques such as a weighted average, majority vote, or arithmetic average. For n pairs of training samples, bagging (bootstrap aggregating) selects M set of samples from n with replacement. Based on each sample, it trains functions $f_1(\mathbf{z}_1), \dots, f_M(\mathbf{z}_M)$. Then, these individual functions or trees are aggregated for regression as:

$$\hat{f} = \sum_{i=1}^M f_i(\mathbf{z}_i). \quad (\text{B.19})$$

The optimized regression criteria, or loss function, to select locations for splits is L_{MSE} (see Eq. (B.18)).

Random Forest (RF)

RF is a model-free ensemble emulator, which provides good accuracy by combining the performance of numerous DTs to classify or predict the value of a variable Breiman (2001); Breiman et al. (2017). For given input data (e.g., feature vector \mathbf{z}), RF builds a number of regression trees (M) and averages the results. For each tree $\mathfrak{T}_m(\mathbf{z})$ for all $m = 1, 2, \dots, M$, the RF prediction is:

$$\hat{f}_{rf}^M = \frac{1}{M} \sum_{m=1}^M \mathfrak{T}_m(\mathbf{z}). \quad (\text{B.20})$$

For regression, L_{MSE} (Eq. (B.18)) is used for the loss function. For classification, the Gini impurity function is used for the loss function and for k class variables, the Gini impurity is:

$$H(\mathbf{z}_m) = \sum_k p_{mk} (1 - p_{mk}). \quad (\text{B.21})$$

AdaBoost (AdaB)

AdaB (aka Adaptive Boosting) converts weak learners into a strong learners Freund (1995); Freund and S. (1996); Freund and Schapire (1996, 1997); Schapire and Freund (1995). Weak learners are DTs with a single split that are also known as decision stumps. AdaB is a greedy and forward stage-wise additive model (adding up multiple models to create a composite model) with an exponential loss function that iteratively fits a weak classifier to improve the current estimator. AdaB puts more weight on difficult-to-learn labels and less on others. AdaB constructs a tree regressor, f_m , from training data so that $f_m : \mathbf{x} \rightarrow y$. Every pair of training data is passed through f_m . Then, f_m calculates a loss for each training datum using the square-loss function:

$$L_i = \frac{|\hat{y}_i - y_i|^2}{J^2}. \quad (\text{B.22})$$

Then, the L_i is averaged by $\hat{L} = \sum_{i=1}^n L_i p_i$ to measure confidence in the prediction as:

$$\theta = \frac{\hat{L}}{1 - \hat{L}}. \quad (\text{B.23})$$

The resulting θ is used to update weights: $w_i \rightarrow w_i \theta \exp(1 - L_i)$. For \mathbf{x}_i , each of M trees/regressors makes a prediction h_m , $m = 1, \dots, T$, to form a cumulative function:

$$f = \inf \left[y \in \mathbb{Y} : \sum_{m:h_m \leq y} \log \left(\frac{1}{\theta_m} \right) \geq \frac{1}{2} \sum_m \log \left(\frac{1}{\theta_m} \right) \right]. \quad (\text{B.24})$$

DT-based AdaB is a heterogeneous emulator that applies both DT and boosting base estimators to learn a prediction function.

Gradient Boosting Method (GBM)

GBM learns functions like AdaB, but it generalizes the model by allowing optimization of an arbitrary differentiable loss function. GBM builds learning function f

for M trees as:

$$f = \sum_{m=1}^M \gamma_m h_m(\mathbf{x}_m). \quad (\text{B.25})$$

After learning each weak model, the additive model (f_m) is built in a greedy fashion:

$$f_m = f_{m-1} + \gamma_m h_m, \quad (\text{B.26})$$

where the newly added tree minimizes the least-squared function, L_{lsqr} , for previous model f_{m-1} by:

$$L_{\text{lsqr}} = \min_w \sum_{i=1}^n (\mathbf{x}_i w - y_i)^2, \quad (\text{B.27})$$

where $i = 1, \dots, n$. The new learner is:

$$h_m = \arg \min_h \sum_{i=1}^n L_{\text{lsqr}} [y_i, f_{m-1}(\mathbf{x}_i) + h(\mathbf{x}_i)]. \quad (\text{B.28})$$

GBM minimizes the L_{lsqr} (optimal loss function for this work) by using steepest descent where the steepest descent direction is the negative gradient of the L_{lsqr} determined at the f_{m-1} . The steepest gradient direction and rate is calculated by:

$$\gamma_m = \arg \min_\gamma \sum_{i=1}^n L_{\text{lsqr}} \left(y_i, f_{m-1}(\mathbf{x}_i) - \gamma \frac{\partial L_{\text{lsqr}}(y_i, f_{m-1}(\mathbf{x}_i))}{\partial f_{m-1}(\mathbf{x}_i)} \right). \quad (\text{B.29})$$

Artificial Neural Networks

Multi-layer Perceptron (MLP)

An MLP is a supervised ML method for classification and prediction. MLPs are feed-forward neural networks or neural networks that are generalizations of linear models for prediction after multi-processing stages Müller and Guido (2016). MLPs consist of numerous simple computation elements called neurons arranged in layers. Neuron output is calculated as the result from a nonlinear activation function whose input is the sum of weighted inputs from all neurons in the preceding layer. The output from neuron n in layer l is:

$$a_n^{(l)} = F \left(\sum_{\mathcal{K}=1}^{\mathcal{N}_{l-1}} w_{\mathcal{K},n}^{(l)} a_{\mathcal{K}}^{(l-1)} + b_n^{(l)} \right). \quad (\text{B.30})$$

The rectified linear unit (ReLU) Nair and Hinton (2010) is the optimal activation function for this work:

$$f(z) = \max(0, z). \quad (\text{B.31})$$

The MSE function (see Equations (B.18)) and cross-entropy function (see Equation (B.6)) are optimal loss functions for regression and classification, respectively.

REFERENCES

- Adrover, A., Cerbelli, S., & Giona, M. (2002). A spectral approach to reaction/diffusion kinetics in chaotic flows. *Computers & Chemical Engineering*, 26, 125-139.
- Ahmmmed, B. (2015). *Numerical modeling of CO₂-water-rock interactions in the Farnsworth, Texas hydrocarbon unit, USA* (Unpublished master's thesis). University of Missouri.
- Alvarez, R. (1973). Complex dielectric permittivity in rocks: A method for its measurement and analysis. *Geophysics*, 38, 920–940.
- Archie, G. E. (1942). The electrical resistivity log as an aid in determining some reservoir characteristics. *Transactions of the AIME*, 146, 54–62.
- Aristodemou, E., & Thomas-Betts, A. (2000). DC resistivity and induced polarisation investigations at a waste disposal site and its environments. *Journal of Applied Geophysics*, 44, 275–302.
- Ashworth, J. B., & Hopkins, J. (1995). *Aquifers of Texas* (Tech. Rep.). Texas Water Development Board.
- Atekwana, E. A., & Slater, L. D. (2009). Biogeophysics: A new frontier in earth science research. *Reviews of Geophysics*, 47(4).
- Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., ... Kaushik, D. (2017). *PETSc users manual revision 3.8* (Tech. Rep.). Argonne National Lab.(ANL), Argonne, IL (United States).
- Barquest, B. A. (1989). *A hydrogeologic assessment of the Austin Chalk outcrop belt, central Texas: Waco, TX* (MS Thesis). Baylor University.
- Barzegar, R., Moghaddam, A. A., Deo, R., Fijani, E., & Tziritis, E. (2018). Mapping groundwater contamination risk of multiple aquifers using multi-model ensemble of machine learning algorithms. *Science of The Total Environment*, 621, 697 - 712.
- Bergen, K. J., Johnson, P. A., Maarten, V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid earth geoscience. *Science*, 363, eaau0323.

- Bingham, N. L. (1993). *Runoff, soil water, and groundwater along a first order stream in the Austin Chalk: Waco, Texas* (PhD Thesis). Baylor University.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24, 123–140.
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5–32.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and Regression Trees*. Taylor & Francis Group, New York.
- Brunton, S. L., & Kutz, J. N. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. In *Ecmi pkdd workshop: Languages for data mining and machine learning* (pp. 108–122).
- Butler, D. K. (2005). *Near-surface geophysics*. Society of Exploration Geophysicists.
- Byrdina, S., Rucker, C., Zimmer, M., Friedel, S., & Serfing, U. (2012). Self-potential signals preceding variations of fumarole activity at Merapi volcano, Central Java. *Volcanology and Geothermal Research*, 215-216, 40-47.
- C., T. F., G, G. H., & L., R. J. (1983). Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3), 242-247.
- Cama, J., Soler, J. M., & Ayora, C. (2019, 09). Acid water-rock-cement interaction and multicomponent reactive transport modeling. *Reviews in Mineralogy and Geochemistry*, 85, 459-498.
- Cannata, S. (1988). *Hydrogeology of a portion of the Washita Prairie Edwards Aquifer, Central Texas: Waco, Texas* (MS Thesis). Baylor University.
- Carpenter, C. (2017). Stress dependence of sandstone electrical properties and deviations from Archie's law. *Journal of Petroleum Technology*, 69, 51–58.
- Chelidze, T. L., & Gueguen, Y. (1999). Electrical spectroscopy of porous rocks: A review-I. Theoretical models. *Geophysical Journal International*, 137, 1–15.
- Chen, L., Wang, M., Kang, Q., & Tao, W. (2018). Pore-scale study of multiphase multicomponent reactive transport during CO₂ dissolution trapping. *Advances in Water Resources*, 116, 208–218.

- Chen, Y., & Or, D. (2006). Geometrical factors and interfacial processes affecting complex dielectric permittivity of partially saturated porous media. *Water Resources Research*, 42.
- Chou, J. S., Tsai, C. F., Pham, A. D., & Lu, Y. H. (2014). Machine learning in concrete strength simulations: Multi-nation data analytics. *Construction and Building Materials*, 73, 771–780.
- Chowdhury, A. H., Osting, T., Furdan, J., & Mathews, R. (2010). *Groundwater-surface Water Interaction in the Brazos River Basin: Evidence from Lake Connection History and Chemical and Isotopic Compositions* (Tech. Rep. No. 375). Texas Water Development Board.
- Clark, B. (2000). *Modeling groundwater velocities in fractured limestone* (MS Thesis). Baylor University.
- Cole, K. S., & Cole, R. H. (1941). Dispersion and absorption in dielectrics I. Alternating current characteristics. *The Journal of Chemical Physics*, 9, 341–351.
- Cole, K. S., & Cole, R. H. (1942). Dispersion and absorption in dielectrics II. Direct current characteristics. *The Journal of Chemical Physics*, 10, 98–105.
- Council, N. R. (2000). *Research needs in subsurface science*. National Academies Press.
- Cracknell, M. J., Reading, A. M., & McNeill, A. W. (2014). Mapping geology and volcanic-hosted massive sulfide alteration in the hellyer–mt charter region, tasmania, using random forests and self-organising maps. *Australian Journal of Earth Sciences*, 61, 287–304.
- Craig, M., & Burdick, K. (2007). Navy divestiture deep in the heart of Texas: Collaboration leads to successful remediation and redevelopment of McGregor property [Magazine]. *The Navy's Environmental Magazine Currents*.
- de Lima, O. A. L., & Sharma, M. M. (1992). A generalized maxwell-wagner theory for membrane polarization in shaly sands. *Geophysics*, 57, 431–440.
- Deutsch, C., & Journel, A. (1992). *Gslib: Geostatistical software library and user's guide*. Oxford University Press. Retrieved from <https://books.google.com/books?id=BiZOAQAAIAAJ>
- Dias, C. A. (2000). Developments in a model to describe low-frequency electrical polarization of rocks. *Geophysics*, 65, 437–451.

- Doherty, J. (2003). Ground water model calibration using pilot points and regularization [Journal Article]. *Groundwater*, 41(2), 170–177.
- Doherty, J. (2016). *Model-independent Parameter Estimation User Manual Part I: PEST, SENSA and Global Optimisers* (6th ed.; Tech. Rep.). Watermark Numerical Computing. Retrieved from <http://pesthomepage.org>.
- Doherty, J., & Hunt, R. J. (2009). Two statistics for evaluating parameter identifiability and error reduction. *Journal of Hydrology*, 366(1-4), 119–127.
- Dukhin, S. S., Shilov, V. N., & Bikerman, J. J. (1974). Dielectric phenomena and double layer in disperse systems and polyelectrolytes. *Journal of the Electrochemical Society*, 121, 154C–154C.
- EarthImager 3D. (2008). Resistivity inversion software manual (Vol. 1.5.3) [Computer software manual].
- Ensafe Inc. (1999). *Draft Groundwater Investigation Report, NWIRP, Mcgregor, Texas* (Tech. Rep.). EnSafe, Inc.
- ESRI. (1996). ArcView GIS, ARC/Info: User's Manual [Computer software manual]. (version 3.2)
- Fiandaca, G., Ramm, J., Binley, A., Christiansen, A. V., & Auken, E. (2013). Spectral information through 2d inversion of time domain induced polarization. In *Symposium on the application of geophysics to engineering and environmental problems 2013* (pp. 808–808).
- Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121, 256–285.
- Freund, Y., & S., R. E. (1996). Experiments with a new boosting algorithm. In *icml* (Vol. 96, pp. 148–156).
- Freund, Y., & Schapire, R. E. (1996). Game theory, on-line prediction and boosting. In *Colt* (Vol. 96, pp. 325–332).
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55, 119–139.
- Galiano, V. R., Castillo, M. S., Olmo, M. C., & Rivas, M. C. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71, 804 - 818.

- Gburek, W. J., Folmar, G. J., & Urban, J. B. (1999). Field data and ground water modeling in a layered fractured aquifer [Journal Article]. *Groundwater*, 37(2), 175-184. Retrieved from <http://dx.doi.org/10.1111/j.1745-6584.1999.tb00972.x> doi: 10.1111/j.1745-6584.1999.tb00972.x
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63, 3–42.
- Glover, P. W. J. (2010). A generalized Archie's law for n -phases. *Geophysics*, 75, E247–E265.
- Golden Software. (1997). *Surfer, Surface Mapping System* [Report]. Retrieved from <http://www.tnris.state.tx.us/DigitalData/DEMs/dems.htm> (version 6.04)
- Gresse, M., Vandemeulebrouck, J., Byrdina, S., Chiodini, G., Revil, A., Johnson, T. C., ... Metral, L. (2017). Three-dimensional electrical resistivity tomography of the Solfatara crater (Italy): Implication for the multiphase flow structure of the shallow hydrothermal system. *Journal of Geophysical Research: Solid Earth*, 122, 8749-8768.
- Günther, T., Rücker, C., & Spitzer, K. (2006). Three-dimensional modelling and inversion of dc resistivity data incorporating topography-II. Inversion. *Geophysical Journal International*, 166, 506–517.
- Hammond, G. E., Lichtner, P. C., Lu, C., & Mills, R. T. (2012). Pflotran: Reactive flow & transport code for use on laptops to leadership-class supercomputers. *Groundwater Reactive Transport Models*, 141–159.
- Hammond, G. E., Lichtner, P. C., & Mills, R. T. (2014). Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN. *Water Resources Research*, 50, 208-228.
- Hare, J. P. (2000). Remediation of perchlorate contamination at NWIRP McGregor, Texas [Journal Article]. *Federal Facilities Environmental Journal*, 36(8), 33–38.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer, New York.
- He, Z., Jiang, W. O., Liu, P., & Cui, X. (2005). Hydrocarbon detection with high-power spectral induced polarization, two cases. In *67th eage conference & exhibition*.

- Hughes, J., Langevin, C., Chartier, K., & White, J. (2012). *Documentation of the Surface-Water Routing (SWR1) Process for Modeling Surface-water Flow with the U.S. Geological Survey Modular Ground-Water Model (MODFLOW - 2005)* (Vol. 6; Tech. Rep.). U.S. Geological Survey.
- Hughes, T. J. R. (2012). *The finite element method: Linear static and dynamic finite element analysis*. Prentice-Hall, New Jersey, USA.
- Hulbert, C., Rouet-Leduc, B., Johnson, P. A., Ren, C. X., Rivière, J., Bolton, D. C., & Marone, C. (2019). Similarity of fast and slow earthquakes illuminated by machine learning. *Nature Geoscience*, 12, 69.
- Hunt, R. J., Doherty, J., & Tonkin, M. J. (2007). Are models too simple? Arguments for increased parameterization [Journal Article]. *Groundwater*, 45(3), 254–262. Retrieved from <http://dx.doi.org/10.1111/j.1745-6584.2007.00316.x> doi: 10.1111/j.1745-6584.2007.00316.x
- Hunt, R. J., Fienen, M. N., & White, J. T. (2019). Revisiting "An exercise in groundwater model calibration and prediction" after 30 years: Insights and new directions. *Groundwater*, 1–15. (<https://doi.org/10.1111/gwat.12907>)
- James, S. C., Doherty, J. E., & Eddebbarh, A.-A. (2009). Practical postcalibration uncertainty analysis: Yucca Mountain, Nevada. *Groundwater*, 47(6), 851–869.
- James, S. C., Zhang, Y., & O'Donncha, F. (2018). A machine learning framework to forecast wave conditions. *Coastal Engineering*, 137, 1–10.
- Johnson, T. C., Hammond, G. E., & Chen, X. (2017). PFLOTRAN-E4D: A parallel open-source PFLOTRAN module for simulating time-lapse electrical resistivity data. *Computers & Geosciences*, 99, 72–80.
- Johnson, T. C., & Thomle, J. (2017). 3D decoupled inversion of complex conductivity data in the real number domain. *Geophysical Journal International*, 212, 284–296.
- Johnson, T. C., Versteeg, R. J., Ward, A., Day-Lewis, F. D., & Revil, A. (2010). Improved hydrogeophysical characterization and monitoring through parallel modeling and inversion of time-domain resistivity and induced-polarization data. *Geophysics*, 75, WA27–WA41.
- Kaselow, A., & Shapiro, S. A. (2004). Stress sensitivity of elastic moduli and electrical resistivity in porous rocks. *Journal of Geophysics and Engineering*, 1, 1.

- Kearey, P., Brooks, M., & Hill, I. (2013). *An introduction to geophysical exploration*. John Wiley & Sons. Massachusetts, USA.
- Kenma, A., Vandeborcht, J., Kulessa, B., & Vereecken, H. (2002). Imaging and characterisation of subsurface solute transport using electrical resistivity tomography (ERT) and equivalent transport models. *Journal of Hydrology*, 267, 125 - 146.
- Kirkwood, C., Cave, M., Beamish, D., Grebby, S., & Ferreira, A. (2016). A machine learning approach to geochemical mapping. *Journal of Geochemical Exploration*, 167, 49 - 61.
- Knox, S. W. (2018). *Machine learning: A concise introduction* (Vol. 285). John Wiley & Sons.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).
- Lagneau, V., Regnault, O., & Descostes, M. (2019, 09). Industrial deployment of reactive transport simulation: An application to uranium in situ recovery. *Reviews in Mineralogy and Geochemistry*, 85, 499-528.
- Leduc, R., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. *Geophysical Research Letters*, 44, 9276–9282.
- Leroy, P., & Revil, A. (2009). A mechanistic model for the spectral induced polarization of clay materials. *Journal of Geophysical Research: Solid Earth*, 114.
- Lesmes, D. P., & Morgan, F. D. (2001). Dielectric spectroscopy of sedimentary rocks. *Journal of Geophysical Research: Solid Earth*, 106, 13329–13346.
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., ... Kumar, J. (2015, 1). Pflotran user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes.
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., ... Frederick, J. M. (2019a). *PFLOTTRAN user manual* (Tech. Rep.). (<http://documentation.pflotran.org>)
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., ... Frederick, J. M. (2019b). *PFLOTTRAN Web page*. (<http://www.pflotran.org>)
- Lichtner, P. C., Steefel, C. I., & Oelkers, E. H. (2019). *Reactive Transport in Porous Media* (Vol. 34). Walter de Gruyter GmbH & Co KG.

- Loke, M. H. (2019). RES3DINV Ver. 3.15 Rapid 3D resistivity & I. P. inversion using the least-square method [Computer software manual]. Saint-Petersburg.
- Loke, M. H., Acworth, I., & Dahlin, T. (2003). A comparison of smooth and blocky inversion methods in 2D electrical imaging surveys. *Exploration Geophysics*, 34, 182–187.
- Loke, M. H., & Barker, R. D. (1996). Rapid least-squares inversion of apparent resistivity pseudosections by a quasi-Newton method. *Geophysical Prospecting*, 44, 131–152.
- Lotti, F., & Doherty, J. (2016). The role of numerical models in environmental decision-making. *Acque Sotterranei-Italian Journal of Groundwater*, 5(3), 73–75.
- Luo, Y., & Zhang, G. (1998). *Theory and application of spectral induced polarization* (No. 8). Tulsa, Oklahoma, USA: Society of Exploration Geophysicists.
- Mace, R. E. (1998). *Groundwater flow and solute transport in a fractured chalk outcrop, north-central Texas: Austin, Texas* (PhD Thesis). University of Texas, Austin.
- MacKay, D. J. C. (1992). Bayesian interpolation. *Neural computation*, 4, 415–447.
- Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16, 100–103.
- Marquardt, D. W. (1970). Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics*, 12, 591–612.
- Marshall, D. J., & Madden, T. R. (1959). Induced polarization, a study of its causes. *Geophysics*, 24, 790–816.
- McCallum, A., & Nigam, K. (1998). A comparison of event models for naïve Bayes text classification. In *Aaa-98 workshop on learning for text categorization* (Vol. 752, pp. 41–48).
- Mellage, A., Smeaton, C. M., Furman, A., Atekwana, E. A., Rezanezhad, F., & Van Cappellen, P. (2018). Linking spectral induced polarization (SIP) and subsurface microbial processes: Results from sand column incubation experiments. *Environmental Science & Technology*, 52, 2081–2090.
- Merriam, J. B. (2007). Induced polarization and surface electrochemistry. *Geophysics*, 72, F157–F166.

- Mesa, R., Reichstein, M., Mahecha, M., Kraft, B., & Denzler, J. (2018). Predicting landscapes as seen from space from environmental conditions. In *Igarss 2018-2018 ieee international geoscience and remote sensing symposium* (pp. 1768–1771).
- Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naïve Bayes—which naïve Bayes? In *Ceas* (Vol. 17, pp. 28–69).
- Moeck, C., Molson, J., & Schirmer, M. (2019). Pathline density distributions in a Null-Space Monte Carlo approach to assess groundwater pathways. *Groundwater*, 1–19. (<https://doi.org/10.1111/gwat.12900>)
- Molins, S., & Knabner, P. (2019, 09). Multiscale approaches in reactive transport modeling. *Reviews in Mineralogy and Geochemistry*, 85, 27-48.
- Molnar, C. (2019). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar.
- Montavon, G., Samek, W., & Müller, K. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73, 1–15.
- Moore, C. (2006). *The use of regularized inversion in groundwater model calibration and prediction uncertainty analysis* (Unpublished doctoral dissertation). University of Brisbane.
- Moore, C., & Doherty, J. (2005). Role of the calibration process in reducing model predictive error. *Water Resources Research*, 41(5), W05020.
- Moore, C., & Doherty, J. (2006). The cost of uniqueness in groundwater model calibration [Journal Article]. *Advances in Water Resources*, 29(4), 605-623. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0309170805001752> doi: <http://dx.doi.org/10.1016/j.advwatres.2005.07.003>
- Moore, T. L., & McSpadden, H. J. (2009). From bombs to rockets at McGregor, Texas. In *The new horizons forum and aerospace exposition*. Orlando, FL: American Institute of Aeronautics and Astronautics.
- Morgan, F. D., Scira-Scappuzzo, F., Shi, W., Rodi, W., Sogade, J., Vichabian, Y., & Lesmes, D. P. (1999). Induced polarization imaging of a jet fuel plume. In *Symposium on the application of geophysics to engineering and environmental problems* (pp. 541–548).
- Mudunuru, M. K., & Karra, S. (2019). Physics-informed machine learning models for predicting the progress of reactive mixing. *arXiv preprint arXiv:1908.10929v1*.

- Mudunuru, M. K., & Nakshatrala, K. B. (2012). A framework for coupled deformation-diffusion analysis with application to degradation/healing. *International Journal for Numerical Methods in Engineering*, 89, 1144–1170.
- Mudunuru, M. K., & Nakshatrala, K. B. (2016). On enforcing maximum principles and achieving element-wise species balance for advection–diffusion–reaction equations under the finite element method. *Journal of Computational Physics*, 305, 448–493.
- Mudunuru, M. K., & Nakshatrala, K. B. (2017). On mesh restrictions to satisfy comparison principles, maximum principles, and the non-negative constraint: Recent developments and new results. *Mechanics of Advanced Materials and Structures*, 24, 556–590.
- Mudunuru, M. K., Shabouei, M., & Nakshatrala, K. B. (2015). On local and global species conservation errors for nonlinear ecological models and chemical reacting flows. In *Proceedings of asme 2015 international mechanical engineering congress and exposition* (p. V009T12A018-V009T12A018).
- Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT press.
- Myrick, M. (1989). *Aquifer-stream interactions and their hydrologic implications in nonkarstic limestones, Washita Prairie, Central Texas: Waco, Texas* (PhD Thesis). Baylor University.
- M.-Zook, S. A., & Ruppert, S. D. (2017). Explosion monitoring with machine learning: A LSTM approach to seismic event discrimination. In *Agu fall meeting abstracts*.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (icml-10)* (pp. 807–814).
- Nakshatrala, K. B., Mudunuru, M. K., & Valocchi, A. J. (2013). A numerical framework for diffusion-controlled bimolecular-reactive systems to enforce maximum principles and the non-negative constraint. *Journal of Computational Physics*, 253, 278–307.
- Niswonger, R. G., Panday, S., & Motomu, I. (2011). *MODFLOW-NWT, a Newton Formulation for MODFLOW-2005* (Vol. 6; Tech. Rep. No. 6-A37). U.S. Geological Survey. Retrieved from <http://pubs.er.usgs.gov/publication/tm6A37> doi: 10.3133/tm6A37

- O'Donncha, F., Zhang, Y., Chen, B., & James, S. C. (2018). An integrated framework that combines machine learning and numerical models to improve wave-condition forecasts. *Journal of Marine Systems*, 186, 29 - 36.
- O'Donncha, F., Zhang, Y., Chen, B., & James, S. C. (2019). Ensemble model aggregation using a computationally lightweight machine-learning model to forecast ocean waves. *Journal of Marine Systems*, 199, 103206.
- Oldenburg, D. W., & Li, Y. (1994). Inversion of induced polarization data. *Geophysics*, 59, 1327–1341.
- Olhoeft, G. R. (1986). Direct detection of hydrocarbon and organic chemicals with ground penetrating radar and complex resistivity. In *Proc. nwwa/api conf. petroleum hydrocarbons and organic chemicals in ground water-prevention, detection and restoration*.
- Oonk, S., & Spijker, J. (2015). A supervised machine-learning approach towards geochemical predictive modelling in archaeology. *Journal of Archaeological Science*, 59, 80 - 88.
- Öztürk, M. A., Ashraf, M., Aksoy, A., Ahmad, M. S. A., & Hakeem, K. R. (2015). *Plants, pollutants and remediation*. Springer.
- Pacific Northwest National Laboratories. (2014). *E4d*.
<https://e4d.pnnl.gov/Pages/Home.aspx>.
- Parker, R. L. (1977). Understanding inverse theory [Journal Article]. *Annual Review of Earth and Planetary Sciences*, 5(1), 35-64. doi: 10.1146/annualrev.ea.05.050177.000343
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011b). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pinder, G. F., & Celia, M. A. (2006). *Subsurface Hydrology*. New Jersey, USA: John Wiley & Sons, Inc.
- Pollock, D. (2016). *User Guide for MODPATH Version 7-A Particle-tracking Model for MODFLOW* (Vol. 35; Report No. OFR 2016-1086). U.S. Geological Survey. Retrieved from <http://dx.doi.org/10.3133/ofr20161086>

- Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning* (pp. 63–71).
- Reichstein, M., C.-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566, 195.
- Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naïve Bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 616–623).
- Revil, A. (2012). Spectral induced polarization of shaly sands: Influence of the electrical double layer. *Water Resources Research*, 48.
- Revil, A., Finizola, A., Ricci, T., Delcher, E., Peltier, A., B.-Cabusson, S., ... Sun, E. T. H. (2011). Hydrogeology of Stromboli volcano, Aeolian islands (Italy) from the interpretation of resistivity tomograms, self-potential, soil temperature and soil CO₂ concentration measurements. *Geophysical Journal International*, 186, 1078-1094.
- Revil, A., Finizola, A., Sortino, F., & Ripepe, M. (2004). Geophysical investigations at Stromboli volcano, Italy: Implications for ground water flow and paroxysmal activity. *Geophysical Journal International*, 157, 426-440.
- Revil, A., & Florsch, N. (2010). Determination of permeability from spectral induced polarization in granular media. *Geophysical Journal International*, 181, 1480-1498.
- Revil, A., Florsch, N., & Camerlynck, C. (2014). Spectral induced polarization porosimetry. *Geophysical Journal International*, 198, 1016–1033.
- Reynen, A., & Audet, P. (2017). Supervised machine learning on a network scale: Application to seismic event classification and detection. *Geophysical Journal International*, 210, 1394–1409.
- Robinson, J., Johnson, T. C., & Slater, L. (2015). Challenges and opportunities for fractured rock imaging using 3D cross-borehole electrical resistivity. *Geophysics*, 80, E49-E61.
- Rolle, M., & Le Borgne, T. (2019, 09). Mixing and reactive fronts in the subsurface. *Reviews in Mineralogy and Geochemistry*, 85, 111-142.
- Rücker, C., Günther, T., & Spitzer, K. (2006). Three-dimensional modelling and inversion of DC resistivity data incorporating topography-I. Modelling. *Geophysical Journal International*, 166, 495–505.

- Rücker, C., Günther, T., & Wagner, F. M. (2017). pyGIMLi: An open-source library for modelling and inversion in geophysics. *Computers and Geosciences*, 109, 106–123.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1.
- Salah, M. K. (2018). *Machine Learning for Model Order Reduction*. Springer.
- Santner, T. J., Williams, B. J., & Notz, W. (2018). *The Design and Analysis of Computer Experiments* (Vol. 1). Springer.
- Scales, J. A., & Gerszenkorn, A. (1988). Robust methods in inverse theory. *Inverse Problems*, 4, 1071.
- Schapire, R. E., & Freund, Y. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Second european conference on computational learning theory* (pp. 23–37).
- Seigel, H., Nabighian, M., Parasnus, D. S., & Vozoff, K. (2007). The early history of the induced polarization method. *The Leading Edge*, 26(3), 312–321.
- Seigel, H. O. (1959). Mathematical formulation and type curves for induced polarization. *Geophysics*, 24, 547–565.
- Shah, P. H., & Singh, D. N. (2005). Generalized Archie's law for estimation of soil electrical conductivity. *Journal of ASTM International*, 2, 1–20.
- Si, H. (2015). TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41, 11.
- Sin, I., & Corvisier, J. (2019, 09). Multiphase multicomponent reactive transport and flow modeling. *Reviews in Mineralogy and Geochemistry*, 85, 143–195.
- Slater, L., Binley, A., Versteeg, R., Cassiani, G., Birken, R., & Sandberg, S. (2002). A 3D ERT study of solute transport in a large experimental tank. *Journal of Applied Geophysics*, 49, 211 - 229.
- Snieder, R., Hubbard, S., Haney, M., Bawden, G., Hatchell, P., Revil, A., & Group, D. G. M. W. (2007). Advanced noninvasive geophysical monitoring techniques. *Annual Review of Earth and Planetary Sciences*, 35.
- Srinivasan, S., Hyman, J., Karra, S., O'Malley, D., Viswanathan, H. S., & Srinivasan, G. (2018). Robust system size reduction of discrete fracture networks: A multi-fidelity method that preserves transport characteristics. *Computational Geosciences*, 22, 1515–1526.

- Sumner, J. S. (1976). *Principles of induced polarization for geophysical exploration* (Vol. 5). Elsevier.
- T. C. Johnson. (2014). E4D: A distributed memory parallel electrical geophysical modeling and inversion code (Vol. 1.0) (Computer software manual No. PNNL-SA-23783). Pacific Northwest National Laboratories.
- Tarasov, A., & Titov, K. (2013). On the use of the Cole–Cole equations in spectral induced polarization. *Geophysical Journal International*, 195, 352–356.
- Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1, 211–244.
- Titov, K., Komarov, V., Tarasov, V., & Levitski, A. (2002). Theoretical and experimental study of time domain-induced polarization in water-saturated sands. *Journal of Applied Geophysics*, 50, 417–433.
- TNRIS. (1999). *Digital data* [Website].
<http://www.tnris.state.tx.us/DigitalData/DEMs/dems.htm>, Accessed in Aug. 1999
- Tonkin, M., & Doherty, J. (2009). Calibration-constrained monte carlo analysis of highly parameterized models using subspace techniques [Journal Article]. *Water Resources Research*, 45(12). Retrieved from <http://dx.doi.org/10.1029/2007WR006678> doi: 10.1029/2007WR006678
- Tonkin, M., Doherty, J., & Moore, C. (2007). Efficient nonlinear predictive error variance for highly parameterized models [Journal Article]. *Water Resources Research*, 43(7). Retrieved from <http://dx.doi.org/10.1029/2006WR005348> doi: 10.1029/2006WR005348
- Tonkin, M. J., & Doherty, J. (2005). A hybrid regularized inversion methodology for highly parameterized environmental models [Journal Article]. *Water Resources Research*, 41(10).
- Tsang, Y. K. (2009). Predicting the evolution of fast chemical reactions in chaotic flows. *Physical Review E*, 80, 026305(8).
- Valls, C., Martino, L., Svendsen, D. H., C.-Taberner, M., M.-Marí, J., Laparra, V., ... G.-Haro, F. J. (2018). Physics-aware gaussian processes in remote sensing. *Applied Soft Computing*, 68, 69–82.
- Vanhala, H. (1997). Mapping oil-contaminated sand and till with the spectral induced polarization (SIP) method. *Geophysical prospecting*, 45, 303–326.

- Vaudelet, P., Revil, A., Schmutz, M., Franceschi, M., & Bègassat, P. (2011). Induced polarization signatures of cations exhibiting differential sorption behaviors in saturated sands. *Water Resources Research*, 47.
- Vaudelet, P., Revil, A., Schmutz, M., Franceschi, M., & Bégassat, P. (2011). Induced polarization signatures of cations exhibiting differential sorption behaviors in saturated sands. *Water Resources Research*, 47.
- Verma, R. K., Rao, M. K., & Rao, C. V. (1980). Resistivity investigations for ground water in metamorphic areas near Dhanbad, India [Journal Article]. *Groundwater*, 18(1), 46–55. Retrieved from <http://dx.doi.org/10.1111/j.1745-6584.1980.tb03370.x> doi: 10.1111/j.1745-6584.1980.tb03370.x
- Vesselinov, V., Mudunuru, M., Karra, S., O’Malley, D., & Alexandrov, B. (2019). Unsupervised machine learning based on non-negative tensor factorization for analyzing reactive mixing. *Journal of Computational Physics*, 395, 85 - 104.
- Vinegar, H. J., & Waxman, M. H. (1984). Induced polarization of shaly sands. *Geophysics*, 49, 1267–1287.
- Viswanathan, H. S., Hyman, J. D., Karra, S., O’Malley, D., Srinivasan, S., Hagberg, A., & Srinivasan, G. (2018). Advancing graph-based algorithms for predicting flow and transport in fractured rock. *Water Resources Research*, 54, 6085-6099.
- Waco Regional Airport. (2016). *National Weather Service, Zip code: 76708*. (http://weather-warehouse.com/WeatherHistory/PastWeatherData_WacoRgnArpt_Waco_TX.html, Accessed on 2016-01-30)
- Welter, D. E., White, J. T., Hunt, R. J., & Doherty, J. E. (2015). *Approaches in Highly Parameterized Inversion – PEST++ Version 3, A Parameter ESTimation and Uncertainty Analysis Software Suite Optimized for Large Environmental Models* (Vol. 7; Tech. Rep.). US Geological Survey.
- Wong, J. (1979). An electrochemical model of the induced-polarization phenomenon in disseminated sulfide ores. *Geophysics*, 44, 1245–1265.
- Wu, Y., Lin, Y., Zhou, Z., & Delorey, A. (2018). Seismic-net: A deep densely connected neural network to detect seismic events. *arXiv preprint arXiv:1802.02241*.
- Yan, L., Xiang, K., Li, P., Liu, X., & Wang, Z. (2014). Study on the induced polarization model in the exploration for shale gas in southern China. In *Seg technical program expanded abstracts* (pp. 912–916).

- Yuan, B., Tan, Y. J., Mudunuru, M. K., Marcillo, O. E., Delorey, A. A., Roberts, P. M., ... Johnson, P. A. (2019). Using machine learning to discern eruption in noisy environments: A case study using CO₂-driven cold-water geyser in Chimayó, New Mexico. *Seismological Research Letters*, 90, 591–603.
- Zhang, H. (2004). The optimality of naïve Bayes. *AA*, 1, 3.
- ZONDRES3D. (2017). Software for three-dimensional interpretation of data obtained by resistivity and induced polarization methods (land, borehole, and marine variants) [Computer software manual]. Saint Petersburg.
- Zuo, R. (2017, 10 01). Machine learning of mineralization-related geochemical anomalies: A review of potential methods. *Natural Resources Research*, 26, 457–464.