UNIVERSITY OF CALIFORNIA, MERCED

Inverse modeling of soil moisture dynamics: Estimation of soil hydraulic properties and surface water flux

A dissertation submitted in partial satisfaction of the requirements

for the degree Doctor of Philosophy

in

Environmental Systems

by

Toshiyuki Bandai

Committee in charge:

Professor Teamrat A. Ghezzehei, Chair
Professor John T. Abatzoglou
Professor Thomas C. Harmon
Professor Noémi Petra

2022

The dissertation of Toshiyuki Bandai is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____ Date _____

Professor John T. Abatzoglou

_____ Date _____

Professor Thomas C. Harmon

_____ Date _____

Professor Noémi Petra

_____ Date _____

Professor Teamrat A. Ghezzehei, Chair

University of California, Merced
2022

Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to express my gratitude to Dr. Teamrat A. Ghezzehei for his sustained support, which significantly contributed to my education. I enjoyed many discussions on many topics in science and beyond. He gave me intellectual freedom, which was initially challenging but enjoyable later. It has been my great privilege to have supportive committee members. Dr. John T. Abatzoglou and Dr. Thomas C. Harmon reminded me of broader perspectives from other fields through the qualifying exam and occasional guidance. I have had invaluable guidance on the mathematical aspect of Chapter 4 from Dr. Noémi Petra, which opened paths to subjects and careers I would not imagine without her.

It is a pleasure to acknowledge the help of collaborators. I acknowledge Dr. Morteza Sadeghi for providing his experimental data and valuable suggestions for Chapter 3. My sincere thanks go to Dr. Ebrahim Babaeian, Dr. Scott B. Jones, and Dr. Markus Tuller for their suggestions for Chapter 3. I acknowledge the help from Dr. Markus Berli, who provided me with valuable field data and guided me during my visit to the Research Desert Institute to learn about their lysimeter facilities. I am indebted to Dr. Chung Chang for his guidance on X-ray CT during my stay at the Lawrence Berkeley National Laboratory.

It has been a great pleasure to have wonderful people from the soil physics and soil biogeochemistry laboratories. I acknowledge Dr. Asmeret A. Berhe for her energetic guidance during the lab meeting and soil science class. I would like to give special thanks to Dr. Kyungjin Min and Dr. Jing Yan for discussions on soil science and beyond. Outside the laboratories, I have been fortunate to be surrounded by warm graduate students. I am deeply indebted to undergraduate students at UC Merced, who gave me invaluable opportunities to teach various subjects and endured my English. I acknowledge Dr. Ki-Tae Kim and Tucker Hartland for their implementations of an adjoint method for other problems.

With gratitude, I acknowledge the financial support from Graduate Student Summer Fellowship from the ES graduate group, Graduate Dean's Dissertation Fellowship from the Graduate Group, Graduate Student Research support from USDA-NIFA Grant 2016-67019-25283, and the Next Generation Workforce - Ingenuity Internship program from the Lawrence Berkeley National Laboratory. Also, I express my gratitude to the Soil Science Society of America and the HydroML conference for financially supporting me in presenting my studies at conferences. I acknowledge computing time on the Multi-Environment Computer for Exploration and Discovery (MERCED) cluster at UC Merced, which was supported by National Science Foundation Award No. ACI-1429783.

I would like to express my gratitude to my old friends in Japan. Occasional exchanges with them reminded me of the memory of enjoyable past times and enabled me to survive the COVID-19 pandemic. I would like to acknowledge my parents, brother, uncles, and grandmothers for their sustained support. Without their support, I could not pursue my career in the US. Finally, my greatest thanks go to my grandfathers. Memories with them will stay with me forever.

Curriculum Vitae

Education

University of California, Merced, USA *August 2018 - December 2022*
Ph.D. in Environmental Systems
Dissertation: "Inverse modeling of soil moisture dynamics: Estimation of soil hydraulic properties and surface water flux"
Supervisor: Dr. Teamrat A. Ghezzehei

The University of Tokyo, Japan *April 2015 - March 2017*
Master in Biological and Environmental Engineering,
Graduate School of Agricultural and Life Sciences
Thesis: "Forced convective heat transport in saturated porous media"
Supervisor: Dr. Taku Nishimura, Dr. Shoichiro Hamamoto

The University of Tokyo, Japan *April 2011 - March 2015*
Bachelors in Agronomy
Department of Agriculture

Publication[1]

Bandai, T., Ghezzehei, T. A. 2021. Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*. DOI: 10.1029/2020WR027642
Bandai, T., and Ghezzehei, T. A. 2022. Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition, *Hydrol. Earth Syst. Sci.*, 26, 4469–4495, https://doi.org/10.5194/hess-26-4469-2022

[1]The list is limited to the ones during the PhD program.

Oral Presentations[2]

Bandai, T., Ghezzehei, T. A., Petra, N. Estimation of surface flux from soil moisture measurements through PDE-constrained optimization. Kirkham Conference, Skukuza, South Africa, (August 2022).
Bandai, T., Ghezzehei, T. A., Petra, N. Estimation of rainfall rates from soil moisture measurements through PDE-constrained optimization. HydroML Symposium, Pennsylvania, USA, (May 2022).
Bandai, T., Sadeghi, M., Babaeian, E., Tuller, M., Jones, S. B., Ghezzehei, T. A. Characterization of unsaturated water flow in soils using short-wave infrared imaging through inverse modeling. ASA, CSSA, SSSA International Annual Meeting 2021, Salt Lake City, USA (November 2021).
Bandai, T., Ghezzehei, T. A. Numerical simulation of soil moisture dynamics using physics-informed neural networks Jppan Geoscience Union Meeting 2021, Online, (June 2021, **Invited**).
Bandai, T., Ghezzehei, T. A. Application of physics informed neural networks for modeling soil water dynamics. InterPore, Online, (May 2021, **Invited**).
Bandai, T., Ghezzehei, T. A. Estimation of constitutive relationships and soil water flux density from volumetric water content measurements using physics-informed neural networks. ACS International Annual Meeting 2020, Online, (November 2020).
Bandai, T., Ghezzehei, T. A. Inverse solution of soil water dynamics using physics-constrained machine learning. American Geophysical Union Fall Meeting 2019, San Francisco, USA, (December 2019).

Poster Presentations[3]

Ghezzehei, T. A., **Bandai, T.** Simulation of near-surface soil moisture measurement without boundary conditions using physics-informed neural networks. American Geophysical Union Fall Meeting 2021, New Orleans (Online), USA, (December 2020).
Bandai, T., Ghezzehei, T. A. Application of physics-informed neural networks to near-surface soil moisture dynamics. American Geophysical Union Fall Meeting 2020, Online, (December 2020).

[2]See footnote 1
[3]See footnote 1

Abstract

Inverse modeling of soil moisture dynamics:
Estimation of soil hydraulic properties and surface water flux

by

Toshiyuki Bandai

in Environmental Systems

University of California, Merced

Professor Teamrat A. Ghezzehei, Chair

Soil moisture is essential for many applications, such as agricultural irrigation, water resources management, and natural disasters, such as landslides and droughts. With the advancement of measurement technology, a vast amount of soil moisture data is available from ground-based sensors and remote sensing. How can we use such abundant data in a meaningful way? For better interpretability and extrapolation capability, soil moisture data should be analyzed based on a known physical model through an inverse modeling framework. In the dissertation, I explored the inverse modeling of soil moisture dynamics based on the Richardson-Richards equation (RRE) via techniques recently developed in applied mathematics. In Chapter 1, a general introduction is presented. Chapter 2 investigated the application of a neural network-based inverse method called physics-informed neural networks (PINNs). I demonstrated that PINNs with domain decomposition could approximate the solution to the RRE for layered soils by comparing PINNs with an analytical solution of the RRE. Chapter 3 estimated soil hydraulic properties from upward infiltration experiments using the Peters-Durner-Iden (PDI) model. I demonstrated that the PDI model better captured soil moisture dynamics for dry conditions than the van-Genuchten Mualem model. Chapter 4 discusses the estimation of surface water flux from soil moisture measurements through inverse modeling. I compared an adjoint method with PINNs and demonstrated through numerical examples that both methods gave reasonable estimates of surface water flux from soil moisture measurements. However, the adjoint method was more robust than PINNs regarding the reconstructed soil moisture profile for a data-limited case. In Chapter 5, I summarized the limitations of the current approaches and discussed future perspectives of inverse modeling of soil moisture dynamics.

Chapter 1

Introduction

Soil moisture is vital in many applications (Robinson et al., 2008; Vereecken et al., 2008; Babaeian et al., 2019). Monitoring soil moisture can assist in effective irrigation and water resource management and predict the occurrence of landslides. Accurate simulation of soil moisture dynamics for dry conditions is necessary to assess and predict the impact of droughts. With the advancement of technology, more and more soil moisture data are becoming available from ground-based sensors and remote sensing. The overall research question of the dissertation is how we extract knowledge and insight from the soil moisture data. I approached this research question by inverse modeling, which aims to estimate unknown attributes of interest from measurement data combined with their mathematical modeling (Vogel, 2002). Inverse modeling is more promising than data-driven statistical methods because soil moisture data are usually sparse in time and space. Also, physics-based modeling allows interpretation and extrapolation (Willcox et al., 2021).

Soil moisture dynamics can be described by a non-linear partial differential equation (PDE), commonly referred to as the Richardson-Richards equation (RRE) (Richardson, 1922; Richards, 1931). In the RRE, the hydraulic properties of soils are represented by water retention curves (WRCs) and hydraulic conductivity functions (HCFs). These two soil hydraulic functions (SHFs) embody the soil pore network affected by soil texture and structure (Assouline and Or, 2013). SHFs are intrinsically scale-dependent functions, and thus converting laboratory-based SHFs into filed scales is challenging (Miller et al., 1998). Therefore, SHFs need to be measured or estimated for the scale of interest. In addition to SHFs, initial and boundary conditions are required to solve the RRE. Particularly, the upper boundary condition, or surface water flux (i.e., precipitation and evaporation), is an essential variable for water balance. Measuring surface water flux is only possible by building costly lysimeters (Dijkema et al., 2017). To gain knowledge from soil moisture data via inverse modeling based on the RRE, we need to optimize both SHFs and surface water flux.

The overarching goal of the dissertation is to develop an inverse framework to estimate SHFs and surface water flux from near-surface soil moisture measurements. My focus is to harness advancements in applied mathematics, such as physics-informed neural networks (PINNs) (Raissi et al., 2019) and an adjoint method (Ghattas and

Willcox, 2021), as well as a recent model of SHFs called the Peters-Durner-Iden (PDI) model (Peters, 2013; Iden and Durner, 2014; Peters, 2014).

In Chapter 2, I applied PINNs to soil moisture dynamics for layered soils. PINNs solve the forward and inverse problems governed by PDEs based on the universal approximation capability of neural networks (Cybenko, 1989; Hornik, 1991). PINNs have been used in many fields, such as incompressible flows (Jin et al., 2021), subsurface transport (He et al., 2020; Tartakovsky et al., 2020), and water dynamics in soils (Bandai and Ghezzehei, 2021). However, the application of PINNs to soil moisture dynamics in layered soils, where there exists discontinuity in SHFs at the layer boundary, has not been tested. I evaluated the performance of PINNs to approximate the solution to the RRE by comparing it with analytical solutions and other numerical methods, such as a finite difference method and a finite element method. I demonstrated that the accuracy of PINNs with domain decomposition was comparable to the other numerical methods. Nevertheless, training PINNs requires extensive hyperparameter tuning and computational demand.

In Chapter 3, I investigated the performance of the PDI model to describe soil moisture dynamics for dry conditions. Water molecules are adsorbed on the surface of soil minerals and flow as thin films for dry conditions. Unlike the commonly used van Genuchten-Mualem (VGM) model (Mualem, 1976; van Genuchten, 1980), the PDI model can describe film flow for dry conditions. I used soil moisture data from upward infiltration experiments for seven soils with distinctive textures conducted by Sadeghi et al. (2017), where soil moisture was measured by a shortwave infrared (SWIR) imaging camera. The SWIR framework enabled us to monitor the advancement of sharp wetting fronts, where the effect of film flow is essential. I demonstrated that the PDI model accurately captured soil moisture dynamics for dry conditions than the VGM model. However, the inverse modeling framework was successful only for medium-textured soils. Its application to coarse-textured and clay-rich soils was challenging due to the violation of one-dimensional flow and homogeneous soil assumptions, respectively.

In Chapter 4, I investigated the use of an adjoint method to estimate surface water flux from near-surface soil moisture measurements. The inverse modeling is computationally challenging because the number of parameters is the number of time steps, which can be one hundred to one thousand. An adjoint method enabled us to efficiently compute the gradient and the Hessian information (Hessian-vector product). The objective function was minimized by the inexact Newton method (Steihaug, 1983; Eisenstat and Walker, 1996), where the Hessian-vector product, not the Hessian itself, was used to solve the Newton system via a conjugate gradient method. This adjoint framework was compared with PINNs by three numerical experiments. The comparison demonstrated that both methods gave a reasonable estimate of the surface water flux. However, the adjoint method was more robust than PINNs in terms of the reconstructed soil moisture profile for a data-limited case.

In Chapter 5, I summarized the limitation of the current studies and discussed future perspectives on inverse modeling of soil moisture dynamics.

References for Chapter 1

Assouline, S. and Or, D. (2013). Conceptual and parametric representation of soil hydraulic properties: A review. *Vadose Zone Journal*, 12(4):1–20.

Babaeian, E., Sadeghi, M., Jones, S. B., Montzka, C., Vereecken, H., and Tuller, M. (2019). Ground, proximal, and satellite remote sensing of soil moisture. *Reviews of Geophysics*, 57:530–616.

Bandai, T. and Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*, 57.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals Systems*, 2:303–314.

Dijkema, J., Koonce, J. E., Shillito, R. M., Ghezzehei, T. A., Berli, M., van der Ploeg, M. J., and van Genuchten, M. T. (2017). Water distribution in an arid zone soil: Numerical analysis of data from a large weighing lysimeter. *Vadose Zone Journal*.

Eisenstat, S. C. and Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM Journal of Scientific Computing*, 17(1):16–32.

Ghattas, O. and Willcox, K. (2021). Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554.

He, Q., Brajas-solano, D., Tartakovsky, G., and Tartakovsky, A. M. (2020). Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources*, 141:103610.

Hornik, K. (1991). Approximation capabilities of multilayer neural network. *Neural Networks*, 4(1991):251–257.

Iden, S. C. and Durner, W. (2014). Comment on "Simple consistent models for water retention any hydraulic conductivity in the complete moisture range" by A. Peters. *Water Resources Research*, 50:7530–7534.

Jin, X., Cai, S., Li, H., and Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426.

Miller, C. T., Christakos, G., Imhoff, P., McBride, J. F., and Pedit, J. A. (1998). Multiphase flow and transport modeling in heterogeneous porous media: Challenges and approaches. *Advances in Water Resources*, 21(2):77–120.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522.

Peters, A. (2013). Simple consistent models for water retention and hydraulic conductivity in the complete moisture range. *Water Resources Research*, 49:6765–6780.

Peters, A. (2014). Reply to comment by S. Iden and W. Durner on "Simple consistent for water retention and hydraulic conductivity in the complete moisture range". *Water Resources Research*, 50:7535–7539.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333.

Richardson, L. F. (1922). *Weather prediction by numerical process*. Cambridge University Press, Cambridge, United Kingdom.

Robinson, D. A., Campbell, C. S., Hopmans, J. W., Hornbuckle, B. K., Jones, S. B., Knight, R., Ogden, F., Selker, J., and Wendroth, O. (2008). Soil moisture measurement for ecological and hydrological watershed-scale observatories: A review. *Vadose Zone Journal*, 7:358–389.

Sadeghi, M., Sheng, W., Babaeian, E., Tuller, M., and Jones, S. B. (2017). High-resolution shortwave infrared imaging of water infiltration into dry soil. *Vadose Zone Journal*, 16(13).

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 23(8):26–28.

Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., and Barajas-Solano, D. (2020). Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research*, 56:e2019WR026731.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America*, 44:892–898.

Vereecken, H., Huisman, J. A., Bogena, H., Vanderborght, J., Vrugt, J. A., and Hopmans, J. W. (2008). On the value of soil moisture measurements in vadose zone hydrology: A review. *Water Resources Research*, 44.

Vogel, C. R. (2002). *Computational methods for inverse problems.*

Willcox, K. E., Ghattas, O., and Heimbach, P. (2021). The imperative of physics-based modeling and inverse theory in computational science. *Nature Computational Science*, 1:166–168.

Chapter 2

Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition

Abstract. [1]Modeling water flow in unsaturated soils is vital for describing various hydrological and ecological phenomena. Soil water dynamics is described by well-established physical laws (Richardson-Richards equation (RRE)). Solving the RRE is difficult due to the inherent nonlinearity of the processes, and various numerical methods have been proposed to solve the issue. However, applying the methods to practical situations is very challenging because they require well-defined initial and boundary conditions. Recent advances in machine learning and the growing availability of soil moisture data provide new opportunities for addressing the lingering challenges. Specifically, physics-informed machine learning allows both the known physics and data-driven modeling to be taken advantage of. Here, we present a physics-informed neural networks (PINNs) method that approximates the solution to the RRE using neural networks while concurrently matching available soil moisture data. Although the ability of PINNs to solve partial differential equations, including the RRE, has been demonstrated previously, its potential applications and limitations are not fully known. This study conducted a comprehensive analysis of PINNs and carefully tested the accuracy of the solutions by comparing them with analytical solutions and accepted traditional numerical solutions. We demonstrated that the

[1]The current chapter is published as Bandai, T., & Ghezzehei, T. A. (2022). Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition. Hydrology and Earth System Sciences, 26(16), 4469–4495. https://doi.org/10.5194/hess-26-4469-2022.

solutions by PINNs with adaptive activation functions are comparable with those by traditional methods. Furthermore, while a single neural network (NN) is adequate to represent a homogeneous soil, we showed that soil moisture dynamics in layered soils with discontinuous hydraulic conductivities are correctly simulated by PINNs with domain decomposition (using separate NNs for each unique layer). A key advantage of PINNs is the absence of the strict requirement for precisely prescribed initial and boundary conditions. In addition, unlike traditional numerical methods, PINNs provide an inverse solution without repeatedly solving the forward problem. We demonstrated the application of these advantages by successfully simulating infiltration and redistribution constrained by sparse soil moisture measurements. As a free by-product, we gain knowledge of the water flux over the entire flow domain, including the unspecified upper and bottom boundary conditions. Nevertheless, there remain challenges that require further development. Chiefly, PINNs are sensitive to the initialization of NNs and are significantly slower than traditional numerical methods.

2.1 Introduction

Near-surface soil moisture is a critical variable for understanding land–atmosphere interactions. Its applications range from hydrological modeling and agricultural water management to the prediction of natural disasters (Robinson et al., 2008; Vereecken et al., 2008; Babaeian et al., 2019). Near-surface soil moisture is also a dominant factor that regulates microbial activity and organic matter dynamics (Pries et al., 2017).

With the technological advancement of soil moisture sensors and remote sensing, the availability of soil moisture data is proliferating (e.g., Sheng et al., 2017; Babaeian et al., 2019). To gain knowledge and insight from such abundant soil moisture data, machine learning (ML) is an appealing tool as we have seen its successes in various fields, such as image recognition, machine translation, and natural language processing. However, soil moisture data are sparsely collected with measurement errors in heterogeneous and complex soils. Therefore, it may not be enough to solely rely on a data-driven machine learning approach. An alternative approach combines machine learning with physical modeling, mainly described as differential equations. Such a hybrid approach has attracted attention in computational physics and related fields and is called scientific ML or physics-informed ML (Karniadakis et al., 2021).

Physical modeling of soil moisture is commonly conducted by solving a partial differential equation (PDE) called the Richardson–Richards equation (RRE) (Richardson, 1922; Richards, 1931). Water transport properties in soils are expressed in the RRE as two relations, the water retention curve (WRC) and the hydraulic conductivity function (HCF), which are both highly non-linear functions (Assouline and Or, 2013). This double non-linearity makes it challenging to analyze and solve the RRE, which has been investigated by soil physicists and mathematicians for many decades (Philip, 1969; Radu et al., 2008; Mitra and Vohralík, 2021). Indeed, the RRE is one of

the most challenging PDEs in hydrology, and the large-scale soil moisture modeling based on the RRE has been prohibitive due to the computational demand and its unreliable solution (Paniconi and Putti, 2015; Farthing and Ogden, 2017).

Artificial neural networks (NNs) have attracted attention as an alternative numerical solver of PDEs recently. While there are some similarities between artificial NNs and biological NNs (i.e., our brain), in this context, artificial NNs should be considered as mathematical functions with specific architectures with many adjustable parameters. Thus, artificial NNs are simply referred to as NNs in this study. Cybenko (1989) and Hornik (1991) mathematically proved that NNs can approximate any continuous function under certain conditions. The application of NNs to various fields relies on this so-called universal approximation capability to find functional relationships between available input data and target variables. Regardless of the tremendous implication of the universal approximation capability, it is not easy to find NNs that can approximate such functional relationships. Therefore, significant efforts have been devoted in the machine learning community to investigate how to adjust NN parameters (or train NNs) for specific problems.

The solution to PDEs is a functional relationship between dependent and independent variables. Therefore, it is straightforward to apply NNs to approximate the solution. Lagaris et al. (1998) were among the first to use NNs to approximate the solution to boundary value problems for PDEs. To train NNs, they defined a loss function so that NNs satisfy both PDEs and the corresponding boundary conditions and minimized the loss function to estimate the NN parameters. The minimization of the loss function required the computation of the gradient of the loss function with respect to the NN parameters as well as the partial derivatives of the PDEs. At that time, such gradients and partial derivatives were computed by manual derivation and programming. However, recent progress in the software environment (e.g., Tensorflow (Abadi et al., 2015) and Pytorch (Paszke et al., 2019)) implementing automatic differentiation (Baydin et al., 2018) enabled us to compute such gradients and partial derivatives in a scalable manner with advanced processors such as graphics processing units (GPUs). With the technological progress, the NN-based methods to solve PDEs were reformulated as physics-informed neural networks (PINNs) by Raissi et al. (2019). PINNs have been applied to both forward and inverse modeling of PDEs in various fields, such as incompressible flows (Jin et al., 2021), subsurface transport (He et al., 2020; Tartakovsky et al., 2020), and water dynamics in soils (Bandai and Ghezzehei, 2021). Karniadakis et al. (2021) provide a comprehensive review of PINN approaches.

PINNs are particularly promising for ill-posed forward and inverse modeling. Traditional numerical solvers, such as finite difference methods (FDMs) and finite element methods (FEMs), require well-defined initial and boundary conditions, but they are often incomplete (i.e., ill-posed). This is usually the case for simulating near-surface soil moisture in real-world field condition, where obtaining accurate initial and boundary conditions is not possible without costly instrumentation (e.g., lysimeters or flux towers) (e.g., Dijkema et al., 2017).

A natural question on PINNs is whether PINNs can replace traditional numerical

solvers. According to previous studies and our experiences, it is currently hard for PINNs to achieve the same accuracy as traditional methods in a competitive computational time. However, PINNs might be a competitive numerical solver for highly non-linear PDEs (e.g., the RRE and the Navier-Stokes equation) in high dimensions, where traditional numerical solvers become computationally demanding. For example, when simulating water flow under infiltration based on the RRE using traditional numerical methods, the spatial mesh must be very small near the soil surface to obtain reliable solutions, which is computationally very expensive or intractable for a three-dimensional watershed scale. Although there is progress in numerical methods other than PINNs, including a discontinuous Galerkin method with adaptive spatial and temporal meshes (Clément et al., 2021), it is worthwhile to seek the potential of PINNs to solve the RRE.

In the study, we conducted a comprehensive analysis of PINNs as a forward and inverse numerical solver of the one-dimensional RRE for homogeneous and heterogeneous soils. This paper is a continuation of our previous work (Bandai and Ghezzehei, 2021), where PINNs were used to estimate soil hydraulic properties from soil moisture measurements. Unlike the previous work, we investigated the ability of PINNs to solve the forward problem in addition to the inverse problem, and the framework was extended to heterogeneous soils. Because of the rapid progress in the field, it is impossible to test all the variants of PINNs and their training methods. Accordingly, we implemented some promising PINN methods, including layer-wise locally adaptive activation function (L-LAAF) (Jagtap et al., 2020) and domain decomposition (Jagtap and Karniadakis, 2020), where two NNs interact with each other through interface conditions to account for the discontinuity of hydraulic conductivity across layers (see Fig. 2.1). To validate and evaluate the solutions derived from PINNs, we compared them to the analytical solutions given by Srivastava and Yeh (1991) and numerical solutions by an FDM and an FEM. The effects of the architecture of NNs and various parameters on the performance of PINNs were investigated. In addition to the forward modelings, we conducted inverse modeling using synthetic data, where a surface water flux upper boundary condition was estimated from near-surface soil moisture measurements by PINNs. Finally, we discuss current challenges and future perspectives of PINNs for forward and inverse modeling of soil moisture dynamics based on the RRE.

2.2 Methods

2.2.1 Richardson-Richards equation

We consider water transport in unsaturated isothermal rigid soils. In this study, hysteresis and vapor flow are ignored, and soil hydraulic properties are isotropic. The mass balance of water in a control volume implies the continuity equation:

$$\frac{\partial \theta}{\partial t} = -\nabla \cdot \mathbf{q} + S, \tag{2.1}$$

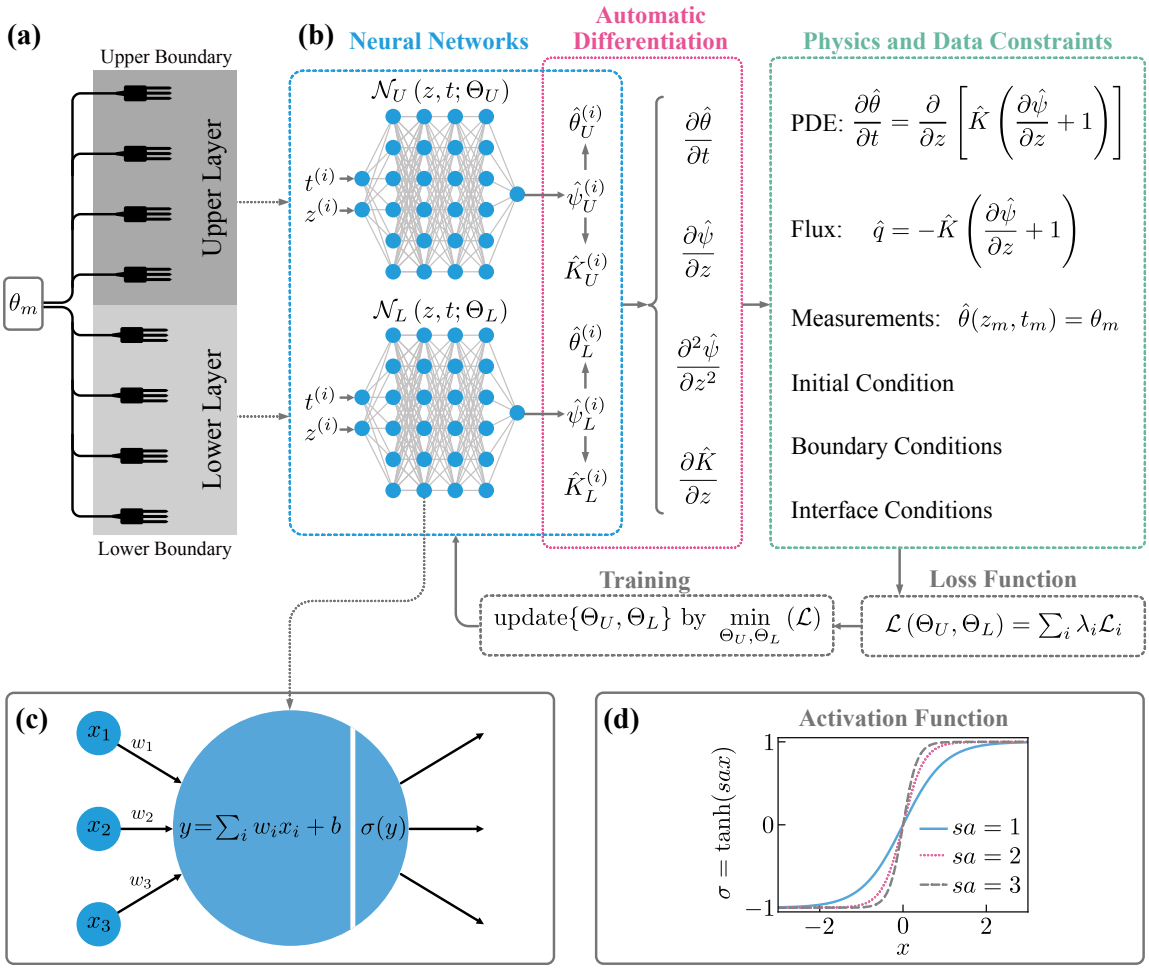

Figure 2.1: (**a**): Schematic description of a soil consisting of two distinct layers with soil moisture sensors. (**b**): Physics-informed neural networks (PINNs) with domain decomposition for a two-layered soil. For each input $t^{(i)}$ and $z^{(i)}$, physics and data constraints are computed through automatic differentiation. The neural network parameters Θ_U and Θ_L are estimated by minimizing the loss function $\mathcal{L}$. (**c**): Computation by a single unit. The input values (x_1, x_2, x_3) are summed with weights (w_1, w_2, w_3) and added by a bias term b. The result is fed into the activation function σ. (**d**): The tanh function as an adaptive activation function with a fixed scaling parameter s and a trainable slope parameter a. The figure was inspired by Jagtap and Karniadakis (2020).

where θ is the volumetric water content [L^3 L^{-3}]; t is the time [T]; $\mathbf{q}$ is the water flux in three dimensions [L T^{-1}]; S is the source term [T^{-1}]. The water flux $\mathbf{q}$ can be derived from the Buckingham-Darcy law (Buckingham, 1907):

$$\mathbf{q} = -K(\theta)\nabla H, \tag{2.2}$$

where K is the hydraulic conductivity [L T^{-1}] and H is the total water head [L], which is the sum of the water potential in soils ψ [L] and the elevation head z (positive upward). Equations (2.1) and (2.2) are combined to derive the Richardson-Richards equation (RRE) (Richardson, 1922; Richards, 1931):

$$\frac{\partial \theta(\psi)}{\partial t} = \nabla \cdot [K(\theta)\nabla(\psi + z)] + S. \tag{2.3}$$

This form of the RRE is called the mixed form RRE, where both the volumetric water content θ and the water potential ψ appear in the equation. To solve the RRE, the two relationships (i.e., $\theta(\psi)$ and $K(\theta)$) need to be defined. The $\theta(\psi)$ relationship is called the water retention curve (WRC), while $K(\theta)$ is referred to as the hydraulic conductivity function (HCF). The WRC and HCF are called constitutive relationships of the RRE that characterize the movement of the water in the pore space. WRCs and HCFs are commonly expressed by parametric models such as the Brooks and Corey model (Brooks and Corey, 1964), the van Genuchten-Mualem model (van Genuchten, 1980), and the Kosugi model (Kosugi, 1996). In this study, the one-dimensional RRE without the source term S is studied, which is written as

$$\frac{\partial \theta(\psi)}{\partial t} = \frac{\partial}{\partial z}\left[K(\theta)\left(\frac{\partial \psi}{\partial z} + 1\right)\right]. \tag{2.4}$$

The zero source term S means the neglect of plant water uptake, which is valid for bare soils or soil moisture dynamics under infiltration. The one-dimensional assumption can be reasonable because water flow in near-surface soils is predominantly vertical.

2.2.2 Analytical solutions

It is difficult to obtain analytical solutions to the RRE because of the non-linearity of the WRC and the HCF. In particular, analytical solutions to the RRE for layered soils are extremely scarce. Srivastava and Yeh (1991) provided one of a few analytical solutions to the transient one-dimensional RRE for both homogeneous and two-layered soils. The analytical solutions are based on the linearization of the RRE using the following relationships for WRCs and HCFs for $\psi < 0$ (Gardner, 1958):

$$\theta = \theta_r + (\theta_s - \theta_r)e^{\alpha_G \psi}, \tag{2.5}$$

$$K = K_s e^{\alpha_G \psi}, \tag{2.6}$$

where θ_r is the residual water content [L^3 L^{-3}]; θ_s is the saturated water content [L^3 L^{-3}]; α_G is the pore-size distribution parameter [L^{-1}]; K_s is the saturated hydraulic

conductivity [L T^{-1}]. Note that the parameter α_G can be interpreted using van-Genuchten parameters α_{VG} and n_{VG} (van Genuchten, 1980), as $\alpha_G \approx 1.3\alpha_{VG}n_{VG}$ (Ghezzehei et al., 2007). Although the parametric expressions for WRCs and HCFs as well as the parameter values used in the study do not necessarily represent hydraulic properties of real soils, the analytical solutions can serve to validate and assess the performance of PINN solutions to the RRE.

2.2.2.1 Homogeneous soil

The analytical solution for a homogeneous soil requires a set of initial and boundary conditions. The lower boundary condition is a Dirichlet boundary condition $\psi = \psi_{lb}$ at $z = -Z$, where Z is the vertical length of the soil [L]. The initial condition is the steady-state solution of the RRE determined by the lower boundary condition and a constant water flux upper boundary condition $q = q_A$ at $z = 0$. The analytical solution to the time-dependent RRE with the initial and lower boundary condition as well as a constant water flux upper boundary condition $q(t) = q_B$ at $z = 0$ is written in terms of $K^* := K/K_s$:

$$\begin{aligned} K^* &= q_B^* - (q_B^* - e^{\alpha_G \psi_{lb}})e^{-(z^*+Z^*)} \\ &- 4(q_B^* - q_A^*)e^{-z^*/2}e^{-t^*/4}\sum_{n=1}^{\infty}\frac{\sin(\kappa_n(z^*+Z^*))\sin(\kappa_n Z^*)e^{-\kappa_n^2 t^*}}{1+Z^*/2+2\kappa_n^2 Z^*}, \end{aligned} \tag{2.7}$$

where $q_A^* := q_A/K_s$; $q_B^* := q_B/K_s$; $z^* := \alpha_G z$; $Z^* := \alpha_G Z$; $t^* = \alpha_G K_s t/(\theta_s - \theta_r)$; κ_n is the positive roots of the equation $\tan(\kappa Z^*) + 2\kappa = 0$. The analytical solution with respect to the volumetric water content θ can be computed from K^* through Eq. 2.5 and Eq. 2.6 . The explicit analytical solution clarifies larger α_G introduces stronger non-linearity of the solution.

2.2.2.2 Heterogeneous soil

Srivastava and Yeh (1991) provided one-dimensional analytical solutions of heterogeneous soils (i.e., two-layered soil). The analytical solution is based on the assumption that α_G is the same for both layers. Therefore, this analytical solution is limited to analyzing layered soils that have a discontinuity in the hydraulic conductivity K across the layers. In fact, the volumetric water content θ is continuous across the layers as the water potential ψ. The initial and boundary conditions are the same as the homogeneous case. The analytical solution is much more complicated than the homogeneous one, so we refer to the original literature for the detail (Srivastava and Yeh, 1991). However, we provide the computed analytical solutions and numerical derivations on Bandai and Ghezzehei (2022b), which can be useful to validate other numerical methods.

2.2.3 Mathematical formulation of PINNs

2.2.3.1 Feedforward neural networks

Feedforward NNs are used to approximate the solution of PDEs in PINNs. In this section, the mathematical formulation of feedforward NNs with L hidden layers with layer-wise locally adaptive activation functions (L-LAAFs) (Jagtap et al., 2020) is introduced. NNs are mathematical functions $\mathcal{N}$ mapping an input vector $\mathbf{x} \in \mathbb{R}^{n^x}$ to an output vector $\hat{\mathbf{y}} \in \mathbb{R}^{n^y}$:

$$\hat{\mathbf{y}} := \mathcal{N}(\mathbf{x}). \tag{2.8}$$

The hat operator represents prediction throughout the paper. NNs are often represented as layers of units (or neurons), as in Fig. 2.1 (b), where two feedforward NNs consisting of four hidden layers with six units are shown.

NNs are compositions of affine transformations (the composition of linear tranformation and translation) and non-linear functions. Herein, $\mathbf{h}^{[k]} \in \mathbb{R}^{n^{[k]}}$ for an integer k such that $1 \leq k \leq L$ represents the vector value corresponding to the kth hidden layer consisting of $n^{[k]}$ units. $\mathbf{h}^{[k]}$ for each k is computed in the following manner:

$$\begin{aligned}
\mathbf{h}^{[1]} &:= \sigma(sa^{[1]}(\mathbf{W}^{[1]}\mathbf{x} + \mathbf{b}^{[1]})),\\
\mathbf{h}^{[2]} &:= \sigma(sa^{[2]}(\mathbf{W}^{[2]}\mathbf{h}^{[1]} + \mathbf{b}^{[2]})),\\
&\vdots\\
\mathbf{h}^{[L-1]} &:= \sigma(sa^{[L-1]}(\mathbf{W}^{[L-1]}\mathbf{h}^{[L-2]} + \mathbf{b}^{[L-1]})),\\
\mathbf{h}^{[L]} &:= \sigma(sa^{[L]}(\mathbf{W}^{[L]}\mathbf{h}^{[L-1]} + \mathbf{b}^{[L]})),
\end{aligned} \tag{2.9}$$

where $\mathbf{W}^{[k]}$ and $\mathbf{b}^{[k]}$ are the weight matrix and bias vector for the kth hidden layer; $s \geq 0$ is a fixed scaling factor; $a^{[k]}$ represents a trainable parameter changing the shape of the element-wise activation function σ. The output of the NN is computed as

$$\hat{\mathbf{y}} := o(\mathbf{W}^{[L+1]}\mathbf{h}^{[L+1]} + \mathbf{b}^{[L+1]}), \tag{2.10}$$

where o is the output function; $\mathbf{W}^{[L+1]}$ and $\mathbf{b}^{[L+1]}$ are the weight matrix and bias vector for the output layer. The collection of the weight matrices $\mathbf{W} := \{\mathbf{W}^{[1]}, ..., \mathbf{W}^{[L+1]}\}$, the bias vectors $\mathbf{b} := \{\mathbf{b}^{[1]}, ..., \mathbf{b}^{[L+1]}\}$, and the slope parameters $\mathbf{a} := \{a^{[1]}, ..., a^{[L]}\}$ are the parameters of the NN, which are denoted by $\Theta := \{\mathbf{W}, \mathbf{b}, \mathbf{a}\}$ in this paper.

To understand the role of the parameters s and $a^{[k]}$ introduced in the L-LAAF (Jagtap et al., 2020), consider a case where $sa^{[k]} = 1$ for all k, and σ is the identity function. In this case, the neural network $\mathcal{N}$ is nothing but an affine transformation and cannot learn a non-linear relationship. In a standard NN, non-linear activation functions, such as the hyperbolic tangent function (tanh), are used with $sa^{[k]} = 1$ for all k to learn non-linear relationships between input and output variables. As shown in Fig. 2.1 (**d**), the tanh function has a "linear" regime near the origin and exhibits the non-linearity outside the region. By increasing the parameter $sa^{[k]}$, we can increase the slope of the activation function and narrow the "linear" regime. Jagtap et al. (2020) reported that larger $sa^{[k]}$ accelerated the training of PINNs and

captured the high-frequency components of the solution of PDEs, while $sa^{[k]}$ that were too large made the training unstable. Throughout the study, $a^{[k]}$ was initialized to 0.05, and s was set to 20 when the L-LAAF was used.

2.2.3.2 Formulation of PINNs for the RRE

In this section, PINNs to solve the forward and inverse problems for the RRE are described. First, we consider the forward modeling of the one-dimensional RRE defined on a domain $\Omega = (-Z, 0)$, the boundary $\partial\Omega$, and the time [0, T):

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial z}\left[K\left(\frac{\partial\psi}{\partial z} + 1\right)\right], \quad z \in \Omega, \quad t \in (0, T), \tag{2.11}$$

$$\theta(z, 0) = g(z), \quad z \in (\Omega \cup \partial\Omega), \tag{2.12}$$

$$\theta(z, t) = h(z, t), \quad z \in \partial\Omega_D, \quad t \in (0, T), \tag{2.13}$$

$$q(z, t) := -K(z, t)\left(\frac{\partial\psi(z, t)}{\partial z} + 1\right) = i(z, t), \quad z \in \partial\Omega_F, \quad t \in (0, T), \tag{2.14}$$

where $g(z)$ is the initial condition; $h(z, t)$ is the Dirichlet boundary condition on the Dirichlet boundary $\partial\Omega_D$; $q(z, t)$ is the water flux in the vertical direction (positive upward); $i(z, t)$ is the water flux boundary condition on the flux boundary $\partial\Omega_F$. Here, we only use the volumetric water content θ for the initial condition and the Dirichlet boundary condition because the measurement of θ is more reliable than the water potential ψ in practical situations, though the modification to the water potential ψ is straightforward (see Sect. A.1.5). Although we limit ourselves to either the Dirichlet or the water flux boundary condition, the framework can be extended to other boundary conditions. In this study, we focus on a particular situation where the soil surface ($z = 0$) is set to $\partial\Omega_F$, and the bottom ($z = -Z$) is set to $\partial\Omega_D$, which corresponds to when soil moisture dynamics is induced by the surface water flux $q(0, t)$ (i.e., evaporation or infiltration) while the volumetric water content at the bottom $\theta(-Z, t)$ is kept to is kept to $h(-Z, t)$

PINNs aim to approximate the solution of the one-dimensional RRE $\psi(z, t)$ by a NN $\mathcal{N}(z, t)$ with the NN parameters $\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{a}\}$. Because the water potential ψ is negative in unsaturated soils, we used the identity function for the output function o of the NN (Eq. 2.10) and transformed the output as

$$\hat{\psi}(z, t) := -\exp(\mathcal{N}(z, t; \Theta)) + \beta, \tag{2.15}$$

where β is a fixed parameter, which can allow $\hat{\psi}(z, t)$ to be zero or positive (saturated). To construct PINNs for the RRE, the residual of the RRE is defined as:

$$\hat{r}(z, t; \Theta) := \frac{\partial\hat{\theta}}{\partial t} - \frac{\partial}{\partial z}\left[\hat{K}\left(\frac{\partial\hat{\psi}}{\partial z} + 1\right)\right]. \tag{2.16}$$

Here, $\hat{\theta}$ and $\hat{K}$ are computed from $\hat{\psi}$ with the predefined WRC and HCF (i.e., $\theta(\psi)$ and $K(\theta)$). The partial derivatives in the residual are computed through the reverse mode

automatic differentiation (Baydin et al., 2018). In this method, all the computations of PINNs are formulated as computational graphs by the software (TensorFlow in this study), and any derivatives related to the computations can be computed exactly, unlike numerical differentiation. This algorithm is highly efficient when the number of parameters is large and thus suitable for training neural networks with tens of thousands or millions of parameters. The collection of the NN parameters Θ are identified by minimizing a loss function, which is defined as

$$\mathcal{L}(\Theta) := \lambda_r \mathcal{L}_r(\Theta) + \sum_i \lambda_i \mathcal{L}_i(\Theta), \tag{2.17}$$

where λ_r is the weight parameter corresponding to the loss term for the residual of the RRE $\mathcal{L}_r(\Theta)$; λ_i is the weight parameter for the loss term $\mathcal{L}_i(\Theta)$ for $i = \{m, ic, D, F\}$, where m, ic, D, and F represent the measurement data, the initial condition, and the Dirichlet and the water flux boundary conditions, respectively. $\mathcal{L}_r(\Theta)$ and $\mathcal{L}_i(\Theta)$ for $i = \{m, ic, D, F\}$ are defined as:

$$\mathcal{L}_r(\Theta) := \frac{1}{N_r} \sum_{i=1}^{N_r} [\hat{r}(z_r^i, t_r^i)]^2, \tag{2.18}$$

$$\mathcal{L}_m(\Theta) := \frac{1}{N_m} \sum_{i=1}^{N_m} [\hat{\theta}(z_m^i, t_m^i) - \theta_m^i]^2, \tag{2.19}$$

$$\mathcal{L}_{ic}(\Theta) := \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} [\hat{\theta}(z_{ic}^i, 0) - g(z_{ic}^i)]^2, \tag{2.20}$$

$$\mathcal{L}_D(\Theta) := \frac{1}{N_D} \sum_{i=1}^{N_D} [\hat{\theta}(z_D^i, t_D^i) - h(z_D^i, t_D^i)]^2, \tag{2.21}$$

$$\mathcal{L}_F(\Theta) := \frac{1}{N_F} \sum_{i=1}^{N_F} [\hat{q}(z_F^i, t_F^i) - i(z_F^i, t_F^i)]^2, \tag{2.22}$$

where $\{z_r^i, t_r^i\}_{i=1}^{N_r}$ denotes the N_r residual points (also called collocation points) at which the residual of the PDE is evaluated; $\{\theta_m^i, z_m^i, t_m^i\}_{i=1}^{N_m}$ denotes the N_m measurement data points; $\{z_{ic}^i\}_{i=1}^{N_{ic}}$ denotes the N_{ic} initial condition points; $\{z_D^i, t_D^i\}_{i=1}^{N_D}$ denotes the N_D Dirichlet boundary condition points; $\{z_F^i, t_F^i\}_{i=1}^{N_F}$ denotes the N_F water flux boundary condition points. Here, $\mathcal{L}_r$ forces PINNs to satisfy the PDE, and $\mathcal{L}_m$ helps PINNs to fit the measurement data while $\mathcal{L}_{ic}$, $\mathcal{L}_D$, and $\mathcal{L}_F$ enforce the initial condition, the Dirichlet and the water flux boundary conditions, respectively. Note that initial and boundary conditions can be encoded in a hard manner so that the approximated solution automatically satisfies these conditions (Lagaris et al., 1998; Sun et al., 2020). However, we encode these conditions in a soft manner and treat them as data points as in Eq. 2.17 to leverage the flexibility of PINNs, which is essential in practical situations where precise initial and boundary conditions are rarely available.

In the framework, the measurement data can be provided, which is not necessary to make the forward modeling well-posed. PINNs can easily incorporate such additional measurement data to improve accuracy and computational efficiency. As for the inverse modeling, the implementation of the PINNs is almost identical to the forward modeling. If we invert physical parameters in PDEs from measurement data (e.g., saturated hydraulic conductivity K_s), the parameters and the NN parameters are simultaneously estimated. Also, one can drop the loss terms for the initial or boundary conditions if they are not available, though this makes the problem ill-posed. In the study, the forward and the inverse modeling was conducted in Sect. 2.3 and Sect. 2.4, respectively.

2.2.3.3 Errors in PINN solutions

Despite the increasing popularity and successes of PINNs in various fields, the theoretical understanding of PINNs is still limited. Shin et al. (2020) were among the first who conducted a rigorous analysis of PINNs, where they formulated the generalization error of PINNs as the sum of the approximation error, the optimization error, and the estimation error. The approximation error is the distance between the best possible approximation by PINNs and the solution of PDEs. The optimization error is due to the difficulty in minimizing the non-linear and non-convex loss function. The estimation error is caused by the insufficiency of data to train PINNs. They demonstrated theoretically and numerically that the sum of the approximation and the estimation error decreased with the increase in the training data for linear second-order elliptic and parabolic PDEs.

Mishra and Molinaro (2022) provided a theoretical framework to estimate the generalization error of PINNs for a variety of PDEs, including non-linear PDEs. They demonstrated that the generalization error would be sufficiently low if 1) PINNs are trained well (i.e., small optimization error); 2) the number of residual points is large; 3) the solution of PDEs is sufficiently regular. In the study, we numerically analyzed the accuracy of PINN solutions to the RRE and investigated whether their theoretical claims could be applied to our case.

2.2.4 Implementation of PINNs

Training of NNs requires trial and error because the theoretical understanding of the mechanism is still limited. However, feedforward NNs used in PINNs have been investigated for many years, so empirical knowledge is available (Bengio, 2012; LeCun et al., 2012). Those techniques are not new, but we would like to reiterate some of them with the explanation of our implementation of PINNs. The PINN algorithm for heterogeneous soils is summarized in Fig. 2.2. We used TensorFlow (Abadi et al., 2015) to implement PINNs, and the source code is available in Bandai and Ghezzehei (2022b).

Algorithm 1 PINNs with domain decomposition

Step 0: Divide the spatial domain into Ω_U and Ω_L and assign two neural networks $\mathcal{N}_U$ and $\mathcal{N}_L$ to each domain. Determine the architecture of two neural networks $\mathcal{N}_U$ and $\mathcal{N}_L$ and weight parameters λ_i in the loss function $\mathcal{L}$ (Eq. 17).
Step 1: Construct the neural networks $\mathcal{N}_U(z,t;\Theta_U)$ and $\mathcal{N}_L(z,t;\Theta_L)$ with neural network parameters Θ_U and Θ_L.
Step 2: Initialize the neural network parameters to Θ_U^0 and Θ_L^0.
Step 3: Given available data (e.g., initial and boundary conditions, measurement data), train the neural networks $\mathcal{N}_U$ and $\mathcal{N}_L$ by minimizing the loss function $\mathcal{L}(\Theta_U, \Theta_L)$.
i $\leftarrow$ 0
while $i <$ max_iteration_Adam **do**
  $\Theta_U^{i+1} \leftarrow \Theta_U^i + \hat{\Theta}_U^i$
  $\Theta_L^{i+1} \leftarrow \Theta_L^i + \hat{\Theta}_L^i$
  i $\leftarrow$ i + 1
end while

while L-BFGS-B stopping criteria are not met **do**
  $\Theta_U^{i+1} \leftarrow \Theta_U^i + \hat{\Theta}_U^i$
  $\Theta_L^{i+1} \leftarrow \Theta_L^i + \hat{\Theta}_L^i$
  i $\leftarrow$ i + 1
end while

Figure 2.2: The algorithm for PINNs with domain decomposition for a two-layered soil. Ω_U and Ω_L refer to the spatial domain for the upper and lower layers, respectively. In Step 3, max_iteration_Adam is the maximum number of the Adam iteration (set to 100000 in the study); $\hat{\Theta}$ represents the update for the neural network parameter Θ; L-BFGS-B stopping criteria are summarized in Sect. 2.2.4.3.

2.2.4.1 Architecture of neural networks

Before training NNs, it is recommended to transform input data so that the components of input variables $\mathbf{x}$ have zero mean, and each variable has a similar variance. In our implementation, we did not transform or normalize input data (i.e., $t \in [0, T]$ and $z \in [-Z, 0]$). However, when both input variables were positive, the training of PINNs was difficult, as mentioned in LeCun et al. (2012). Thus, it is better to transform input data for future studies. As for output variables, it is also important to take into account their range, as in Eq. 2.15.

The architecture of NNs (i.e., the number of hidden layers L and units $n^{[k]}$ for $k = 1, ..., L$) determines the complexity of functions the NN can learn and thus depends on PDEs of interest and the corresponding initial and boundary conditions. It is known that the expressive capability of NNs grows exponentially with the number of hidden layers (Raghu et al., 2017). We used the same number of units for all hidden layers, and the effects of the architecture of NNs on PINN performance were experimentally investigated. As for activation functions, symmetric activation functions with respect

to the origin, such as the tanh function, are preferable to non-symmetric functions such as the sigmoid function. Because PINNs require the second derivative of state variables in the loss function, we used the tanh function for the activation functions σ for all hidden layers.

2.2.4.2 Initialization

At the beginning of the training, the collection of the weight parameters $\mathbf{W}$ and the bias parameters $\mathbf{b}$ have to be initialized. Glorot and Bengio (2010) demonstrated that simple random initialization caused the activation functions to be "saturated," meaning that the slope of the activation function becomes zero. To prevent this, they introduced the Xavier initialization for the weight parameters $\mathbf{W}$, which was used in our implementation. The bias parameters $\mathbf{b}$ were initialized to be 0. Because the initialization of $\mathbf{W}$ significantly affects PINN solutions, different sets of randomization must be tested. Therefore, we used ten different random seeds for the initialization of each setting of PINNs. In addition to $\mathbf{W}$ and $\mathbf{b}$, we can tune the parameter β in Equation 2.15 to give different initializations of $\hat{\psi}$.

2.2.4.3 Training

PINNs were trained to minimize the loss function (Eq. 2.17). The loss term for the residual $\mathcal{L}_r$ was evaluated at randomly sampled residual points in the spatial and temporal domain (Fig. A.1 (**b**)). For each problem, the same residual points were used. We tested the residual-based adaptive refinement algorithm proposed by Lu et al. (2021b), where residual points are chosen where the residual of PDEs is high. For our case study, the effectiveness of the algorithm was minor, and thus the results are only shown in the Appendix (see Sect. A.1.1 and Fig. A.1).

The weight parameters in the loss function λ_i (Eq. 2.17) play a crucial role in minimizing the loss function. In the original PINN framework proposed by Raissi et al. (2019), all λ_i were set to 1. However, Wang et al. (2021) demonstrated that the loss terms corresponding to initial and boundary conditions need to be penalized more, and optimal values of those weight parameters are problem-dependent. To overcome this challenge, they proposed the learning rate annealing algorithm (Wang et al., 2021), where the weight parameters in the loss function λ_i are updated during training to balance the relative importance of each loss term. We tested the algorithm, but this resulted in a modest improvement compared to the L-LAAF, so the results are given in the Appendix (Sect. A.1.2 and Fig. A.2). In the study, the effects of λ_i were investigated in Sect 2.3.1.5.

It is common to use a stochastic gradient descent algorithm to minimize the loss function to train NNs. In this study, we used the Adam algorithm (Kingma and Ba, 2014). Because the Adam optimizer is not enough to achieve solutions with high accuracy, previous studies on PINNs employed a two-step optimization strategy, and we followed it. First, the loss function was minimized using 10^5 iterations of the Adam algorithm in TensorFlow (Abadi et al., 2015) with the exponential decay of

the learning rate. The initial learning rate was set to 0.001 with the decay rate of 0.90, the decay step was set to 1000, and the other parameters were set to their default values. The Adam algorithm used a "mini-batch" of the data, where only 128 of all residual points were considered while all the initial and boundary data points were used for each iteration. After the Adam algorithm, the loss function was further minimized through the L-BFGS-B optimizer (Byrd et al., 1995) from Scipy (Virtanen et al., 2020), which was terminated once the loss function converged with prescribed thresholds. The L-BFGS-B algorithm can utilize the information on the approximated curvature of the loss function and find a better local minimum than a stochastic gradient descent for the case of PINNs. The following L-BFGS-B parameters were used: maxcor = 50, maxls = 50, maxiter = 50000, maxfun = 50000, ftol = 1.0×10^{-10}, gtol = 1.0×10^{-8}, and the default values for the other parameters. Although it is desirable to tune the parameters for each optimizer, we fixed those parameters in the study.

2.2.4.4 Domain decomposition

Natural soils have distinctive layering, and the hydraulic properties of each layer vary between the layers, which results in continuous but not a differentiable water potential distribution in the soil profile. To deal with such spatial heterogeneity, Jagtap and Karniadakis (2020) proposed a domain decomposition method for PINNs, where a computational domain is divided into sub-domains, and a NN is assigned to each sub-domain. Then, the NNs interact with each other during the training through interface conditions such as the continuity of mass and flux. Such interface conditions can be incorporated into the loss function. For simulating water flow in a two-layered soil, two NNs $\mathcal{N}_U$ and $\mathcal{N}_L$ are assigned to the upper and lower layer, and the continuities of water potential, water flux, and the residual of the RRE are imposed at the boundary:

$$\hat{\psi}_U(z_I, t) = \hat{\psi}_L(z_I, t), \tag{2.23}$$

$$\hat{q}_U(z_I, t) = \hat{q}_L(z_I, t), \tag{2.24}$$

$$\hat{r}_U(z_I, t) = \hat{r}_L(z_I, t), \tag{2.25}$$

where the subscripts $_U$ and $_L$ mean a value with the subsucripts (e.g., $\hat{\psi}_U$) was computed by $\mathcal{N}_U$ and $\mathcal{N}_L$, respectively; z_I represents the spatial coordinate of the interface. These interface conditions are incorporated into the loss function as a loss term (Eq. 2.17):

$$\mathcal{L}_{I_\psi}(\Theta_U, \Theta_L) := \frac{1}{N_I} \sum_{i=1}^{N_I} [\hat{\psi}_U(z_I, t^i) - \hat{\psi}_L(z_I, t^i)]^2, \tag{2.26}$$

$$\mathcal{L}_{I_q}(\Theta_U, \Theta_L) := \frac{1}{N_I} \sum_{i=1}^{N_I} [\hat{q}_U(z_I, t^i) - \hat{q}_L(z_I, t^i)]^2, \tag{2.27}$$

$$\mathcal{L}_{I_r}(\Theta_U, \Theta_L) := \frac{1}{N_I} \sum_{i=1}^{N_I} [\hat{r}_U(z_I, t^i) - \hat{r}_L(z_I, t^i)]^2, \tag{2.28}$$

where N_I is the number of points on the interface, where the loss terms are evaluated; Θ_U and Θ_L are the neural network parameters for $\mathcal{N}_U$ and $\mathcal{N}_L$, respectively. In the implementation, we found that the logarithmic transformation of water potential ψ was helpful to balance the loss terms. Therefore, instead of $\mathcal{L}_{I_\psi}$, we imposed the continuity of the output of the neural networks, as in

$$\mathcal{L}_{I_\mathcal{N}}(\Theta_U, \Theta_L) := \frac{1}{N_I} \sum_{i=1}^{N_I} [\mathcal{N}_U(z_I, t^i) - \mathcal{N}_L(z_I, t^i)]^2. \tag{2.29}$$

Note that the original literature Jagtap and Karniadakis (2020) trained each NN separately by constructing multiple loss functions for parallel computation, but we trained the two NNs ($\mathcal{N}_U$ and $\mathcal{N}_L$) simultaneously. The algorithm is summarized in Fig. 2.2. Although our algorithm is for a two-layered soil in one dimension, this method can be extended to more layers with more complex geometries in higher dimensions (Jagtap and Karniadakis, 2020).

2.2.5 Evaluation of numerical error

Numerical errors of PINNs and the other numerical methods (i.e., FDMs and FEMs) in terms of the volumetric water content θ and the water potential ψ were evaluated by comparing the analytical solutions to those numerical solutions computed on a uniform grid with a spatial step of 0.1 cm and temporal step of 0.1 h. Absolute error, relative absolute error, squared error, and relative squared error were computed. Because these different types of errors exhibited strong correlations, we show, in the following sections, relative squared error in terms of the volumetric water content ϵ^θ defined as:

$$\epsilon^\theta := \frac{\sum_t \sum_z (\hat{\theta}(t,z) - \theta(t,z))^2}{\sum_t \sum_z \theta(t,z)^2}, \tag{2.30}$$

where θ and $\hat{\theta}$ represent analytical and numerical solutions, respectively.

2.3 Forward modeling

In this section, we report the main results of the forward modeling of water transport in homogeneous and heterogeneous soils using PINNs. PINN solutions of the RRE with varying NN architectures and parameters were evaluated using the one-dimensional analytical solutions by Srivastava and Yeh (1991) for homogeneous and heterogeneous soils introduced in Sect. 2.2.2. To assess the performance of PINNs, numerical solutions obtained by an FDM and an FEM are also presented. The implementation of the FDM for the homogeneous case is described in Sect. A.1.3 while the FEM solution for the heterogeneous case was obtained by HYDRUS-1D (Šimůnek et al., 2013).

2.3.1 Homogeneous soil

We simulated water infiltration into a homogeneous soil using the RRE. This simple setup was used to understand the characteristics of PINNs with different settings. We specifically investigated 1) the effects of NN architecture (the number of layers and units as well as the use of the layer-wise locally adaptive activation function (L-LAAF)); 2) the effects of the weight parameters λ_i in the loss function (Eq. 2.17); 3) the effects of the number of the residual points and the upper boundary data points.

2.3.1.1 Problem setup

We considered soil moisture dynamics in a homogeneous soil induced by a constant water flux on the surface ($z = 0$ cm), as introduced in Sect. 2.2.2.1, where $Z = 10$ cm, $T = 10$ h, $q_A = -0.1$ cm h^{-1}, $q_B = -0.9$ cm h^{-1}, $\psi_{lb} = 0$ cm, $\theta_r = 0.06$; $\theta_s = 0.40$; $\alpha_G = 1.0$ cm^{-1}; $K_s = 1.0$ cm h^{-1}.

2.3.1.2 Characteristics of PINN solution

The numerical solution by PINNs and the analytical solution are shown in Fig. 2.3 (**a**). The PINN solution was obtained by using a NN of 5 hidden layers with 50 units using the L-LAAF, which was determined after testing various settings described in the following sections. $N_{ic} = 101$ initial data points ($z = -10.0, -9.9, ..., -0.1, 0.0$ cm), $N_{ub} = 1000$ upper boundary data points ($t = 0.01, 0.02, ..., 9.99, 10.0$ h), and $N_{lb} = 100$ lower boundary data points ($t = 0.1, 0.2, ..., 9.9, 10.0$ h) were used to train the NN. The number of residual points N_r was set to 10000. The weight parameters λ_i in the loss function were set to ten for the initial condition (λ_{ic}), the lower boundary condition (λ_{lb}), and the water flux upper boundary condition (λ_{ub}) while λ_r was set to one for the residual loss term. The difference between the PINN and the analytical solution in the volumetric water content θ is shown in the right column of Fig. 2.3 (**a**). Larger errors were observed near the initial and upper boundary conditions, where strong non-linearity exists due to the surface water flux. Except for this, PINNs could approximate the solution with high accuracy. Figure 2.3 (**b**) showed the FDM solution with a spatial mesh $dz = 0.1$ cm and a time step $dt = 0.01$ h, which is comparable to the temporal resolution of the upper boundary data points given to the PINN. In comparison with the FDM solution, the PINN solution was quite reasonable ($\epsilon^\theta = 4.86 \times 10^{-4}$ for the PINN and $\epsilon^\theta = 9.72 \times 10^{-4}$ for the FDM solutions, respectively). Note that the degrees of freedom of the FDM solution were 101000, while the number of parameters of the NN was 10406, which demonstrates the memory efficiency of PINNs. Also, the number of residual points of PINNs was much smaller than the degrees of freedom of the FDM. However, increasing the number of residual points did not improve the PINN solution, as discussed in Sect. 2.3.1.6, while the FDM solution improved by further minimizing dz and dt ($\epsilon^\theta = 1.03 \times 10^{-5}$ was obtained for $dz = 0.01$ cm and $dt = 0.0001$ h, as shown in Fig. A.3). It is important to note here that an FDM with such a very fine mesh size requires solving a large linear system multiple times for each time step because the

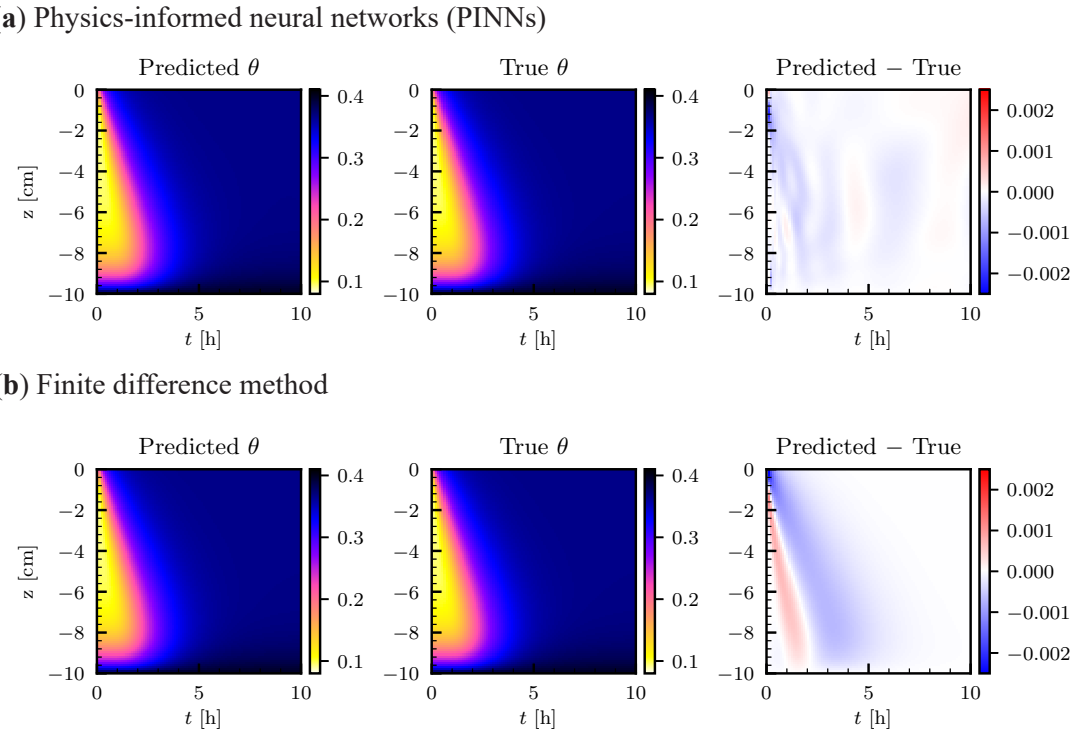

Figure 2.3: Homogeneous soil. (**a**): Physics informed neural network (PINN) solution in terms of volumetric water content θ [-] (left column). The neural network consisted of 5 hidden layers with 50 units with the layer-wise adaptive activation function. True analytical solution (center column) is given by Srivastava and Yeh (1991) (see Sect. 2.2.2.1), and the difference between the PINN and true solutions are shown in the right column. (**b**): Numerical solution by a finite difference method with a spatial mesh of $dz = 0.1$ cm and a time step $dt = 0.01$ h.

RRE is a non-linear PDE, and the number of the iteration increases with decreasing dz, which leads to significant computational demand. This situation becomes worse for higher dimensions. Although we do not test PINNs for higher dimensions, other studies demonstrated the effectiveness of PINNs for higher dimensions (Mishra and Molinaro, 2022). This is the main reason we see the potential of PINNs for a large-scale simulation based on the RRE.

2.3.1.3 Training PINNs

A typical evolution of the loss terms in the loss function during the training (the Adam and L-BFGS-B algorithms) is shown in Fig. 2.4 (**a**). In most cases, the Adam algorithm gave a good minimum of the loss function, and the following L-BFGS-B algorithm met its termination criteria immediately (a spike was observed just before

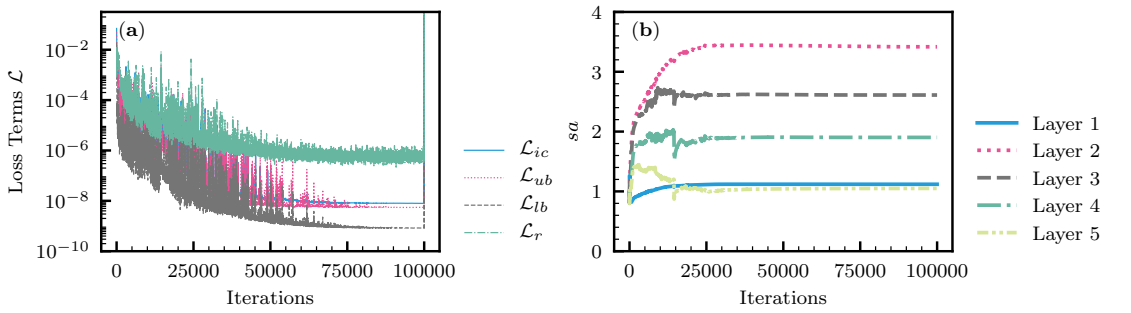

Figure 2.4: Homogeneous soil. (**a**): The evolution of the loss terms for the initial condition $\mathcal{L}_{ic}$, the upper boundary condition $\mathcal{L}_{ub}$, the lower boundary condition $\mathcal{L}_{lb}$, and the residual $\mathcal{L}_r$ during the Adam (100000 iterations) and the following L-BFGS-B training. (**b**): The evolution of the adaptive parameter sa for layer-wise locally adaptive activation functions (Eq. 4.78) for each hidden layer (Layer 1 is next to the input layer).

the termination). Among the loss terms, the residual loss $\mathcal{L}_r$ remained high after the training ($\mathcal{L}_r \approx 10^{-6}$). $\mathcal{L}_r$ indicates whether the RRE was satisfied in the spatial and temporal domain and determines the performance of PINNs. The characteristics of $\mathcal{L}_r$ are further explored in Sect. 2.3.1.7. Figure 2.4 (**b**) shows the evolution of the adaptive parameter sa for the L-LAAF for each hidden layer (Eq. 4.78). The parameter sa changes the slope and the linear regime of activation functions, as shown in Fig. 2.1 (**d**). The parameter sa varied with the iterations of the Adam algorithm and reached its limiting value for each hidden layer. sa for the second hidden layer was the highest, and a smaller sa value was observed for hidden layers closer to the output layer.

Figure 2.5 demonstrates how PINNs learn the solution to the RRE during the training. At the initialization (Fig. 2.5 (**a**)), the PINN solution greatly differed from the true solution. However, PINNs quickly learned the lower boundary condition (see Fig. 2.5 (**b**)). Although the initial condition was given as data points, it took more iterations for PINNs to learn it because of the high non-linearity near the surface induced by the water flux boundary condition. This corresponds to the increase in the adaptive parameter sa for the L-LAAF (see Fig. 2.4 (**b**)). The limiting value of sa for the second hidden layer was 3.4, which makes the tanh function highly non-linear and closer to the step function (see Fig. 2.1 (**d**)). We concluded that the L-LAAF helped PINNs learn the highly non-linear solution of the RRE. Figure 2.5 clearly illustrated that PINNs first learned less complex parts of the solution and then captured the more complex parts. The tendency of feedforward NNs to learn less complex functions is called "spectral bias," and Wang et al. (2022) demonstrated that this spectral bias caused PINNs to fail to learn complex solutions of PDEs. Further research is needed on how PINNs can learn more complex solutions of the RRE, for example, where wetting and drying cycles are studied.

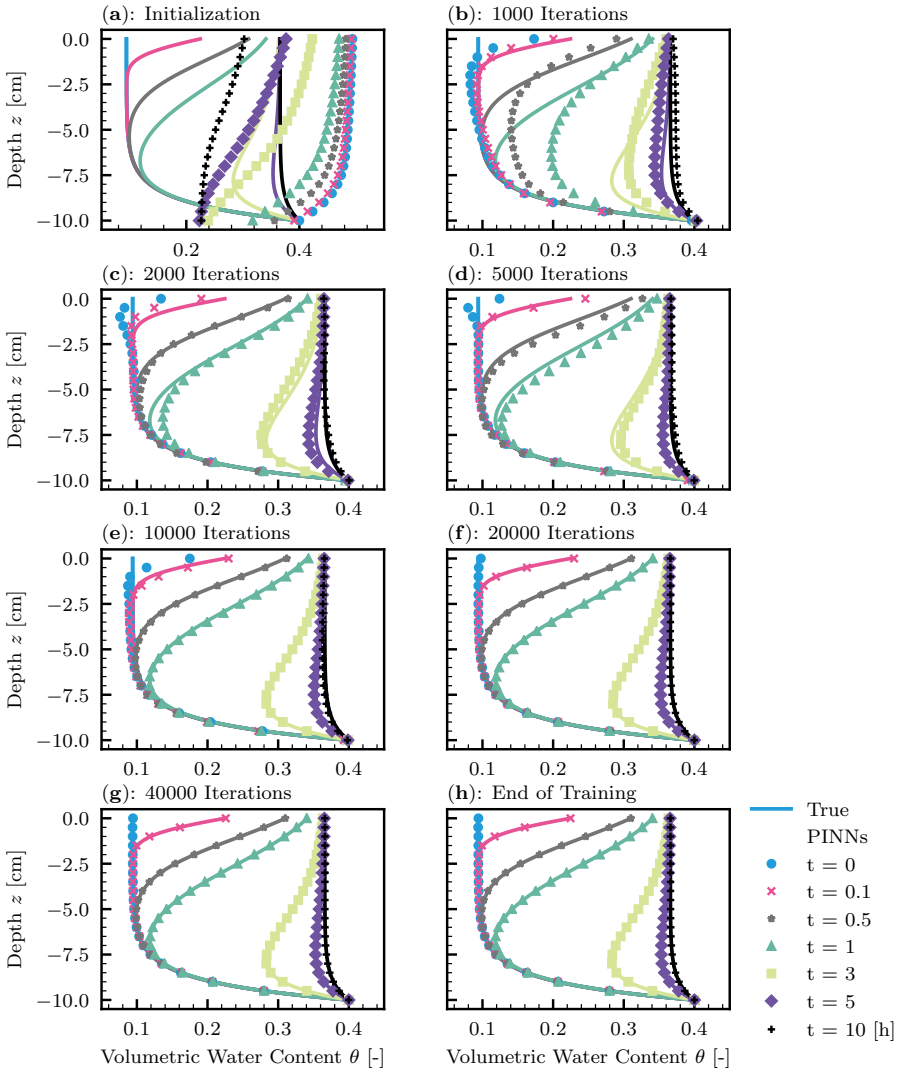

Figure 2.5: Homogeneous soil. The evolution of physics-informed neural network (PINN) solution (points) during the training with the true solution (solid lines). (**a**): Initialization of PINNs. (**b**) to (**g**): 1000, 2000, 5000, 10000, 20000, 40000 iterations of the Adam algorithm. (**h**): The end of 100000 iterations of the Adam algorithm and the following L-BFGS-B algorithm.

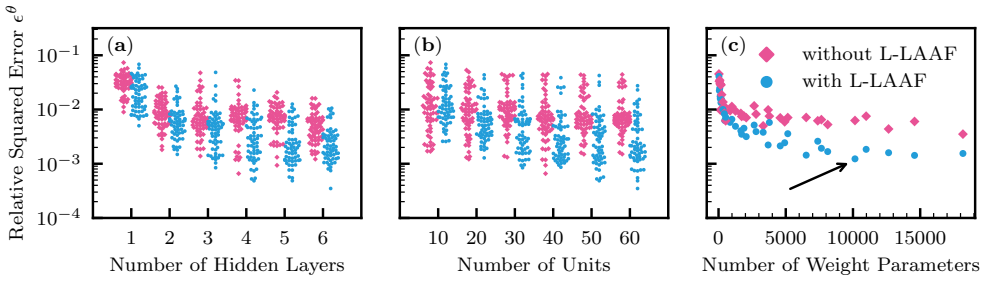

Figure 2.6: Homogeneous soil. Relative squared error in terms of volumetric water content ϵ^θ for different numbers of hidden layers (**a**) and units (**b**) with and without the use of the layer-wise locally adaptive activation function (L-LAAF). In Fig. (**c**), the averaged ϵ^θ were computed for each neural network architecture, and the values were plotted against the number of the weight parameters of each neural network. The arrow indicates the lowest averaged ϵ^θ, which corresponds to neural networks with 5 hidden layers with 50 units used in Fig. 2.3.

2.3.1.4 Effects of neural network architecture

The effects of the number of hidden layers and units of NNs as well as the use of the L-LAAF were investigated. The candidate numbers of hidden layers and units were 1, 2, 3, 4, 5, 6 for hidden layers and 10, 20, 30, 40, 50, 60 for units. The L-LAAF was turned on and off, and ten different random seeds were used for the NN initialization for each NN architecture, which resulted in a total of 720 runs.

The summary of the results is shown in Fig. 2.6. In Fig. 2.6 (**a**) and (**b**), all the 720 data points are plotted while the averaged data for each NN architecture are shown in Fig. 2.6 (**c**) against the number of the weight parameters of each NN architecture. In Fig. 2.6 (**a**), smaller relative squared error ϵ^θ were observed for a larger number of hidden layers. Four hidden layers appeared to be enough for NNs to approximate the solution. As for the number of units, increasing the number gave smaller ϵ^θ though the effect seemed less relevant than that for hidden layers. It is clear from Fig. 2.6 (**c**) that the use of the L-LAAF improved the performance of PINNs. From this analysis, we determined the best NN architecture to be 5 hidden layers with 50 units with the L-LAAF indicated by the arrow in Fig. 2.6 (**c**), whose solution is shown in Fig. 2.3.

2.3.1.5 Effects of weight parameters in loss function

The effects of the weight parameters λ_i in the loss function (Eq. 2.17) were studied by varying λ_i for the initial and boundary conditions while λ_r was fixed to one. We denote λ_{ub} and λ_{lb} as the weight parameters for the upper and lower boundary conditions, respectively. Five different values (1, 3, 10, 30, 100) were tested for each weight parameter with ten different NN initializations, which resulted in a total of

1250 simulations, where the NN architecture was fixed to 5 hidden layers with 50 units with the L-LAAF.

Figure 2.7 shows the effects of the weight parameters λ_i for $i = ic, ub, lb$ on the loss terms $\mathcal{L}_i$ in the loss function and the PINN performance evaluated by the relative squared error ϵ^θ. Each panel in the figure contains all the 1250 simulations, and the effects of the initialization and the λ_i are mixed. Figure 2.7 (**a**), (**e**), and (**i**) demonstrated that higher weight parameters λ_i attained lower values of the corresponding loss terms $\mathcal{L}_i$, which was expected. Another noticeable feature is that the loss term for the residual $\mathcal{L}_r$ increased with the increasing λ_i while λ_r was fixed to one. It was considered that higher weight parameters λ_i for the initial and boundary conditions minimized the emphasis on the residual loss term $\mathcal{L}_r$ (see Fig. 2.7 (**k**) and (**j**)). The increased $\mathcal{L}_r$ lead to less accurate solutions or higher ϵ^θ, which is evident in Fig. 2.7 (**m**) and (**j**). These complicated trends make the PINN approach less consistent than traditional numerical methods. Note that automatic but empirical tuning of λ_i proposed by Wang et al. (2021) did not improve the results in our case, particularly with the L-LAAF, which is shown in Fig. A.2. Because finding the optimal values for λ_i is not our primary purpose, we use the value of ten for all the three weight parameters λ_i for the following analysis.

2.3.1.6 Effects of number of residual and boundary data points

The effects of the number of residual points N_r, where the residual of the RRE is evaluated, were investigated by varying the number $N_r \in \{1000, 3000, 10000, 30000, 100000\}$, which resulted in 50 runs in total. NN architecture was fixed to 5 layers with 50 units with the L-LAAF. As expected, larger error was observed when smaller number of residual points was used (see Fig. 2.8 (**a**)). However, even if we increased the number to 30000 and 100000, the performance of PINNs did not improve. We concluded that this was due to the simultaneous but opposite effect of the number N_r on the loss term $\mathcal{L}_r$, as shown in Fig. 2.8 (**b**). This was because increasing N_r is equivalent to minimizing the importance of each residual point randomly selected in the spatial and temporal domain. Note that we tested the residual-based adaptive refinement algorithm proposed by Lu et al. (2021b), where additional residual points are added while training NNs based on the distribution of the residual values. As shown in Fig. A.1 (**a**), the algorithm seemed to improve the performance of PINNs, but the effectiveness was minor. These findings demonstrate the difficulty in finding an optimal strategy to distribute residual points for PINNs to learn solutions to PDEs with high accuracy.

Also, the effects of the number of upper boundary data points N_{ub} were studied, where $N_{ub} = \{100, 300, 1000, 3000, 10000\}$. NN architecture and training algorithms were set to the same as Sect. 2.3.1.2. Figure 2.9 (**a**) showed that more than 300 upper boundary data points N_{ub} are necessary for PINNs to learn the solution well. This is because PINNs required enough upper boundary data points to capture the surface flux in particular near the initial condition. However, increasing N_{ub} from 300 did not improve the performance of PINNs. At the same time, we observed the

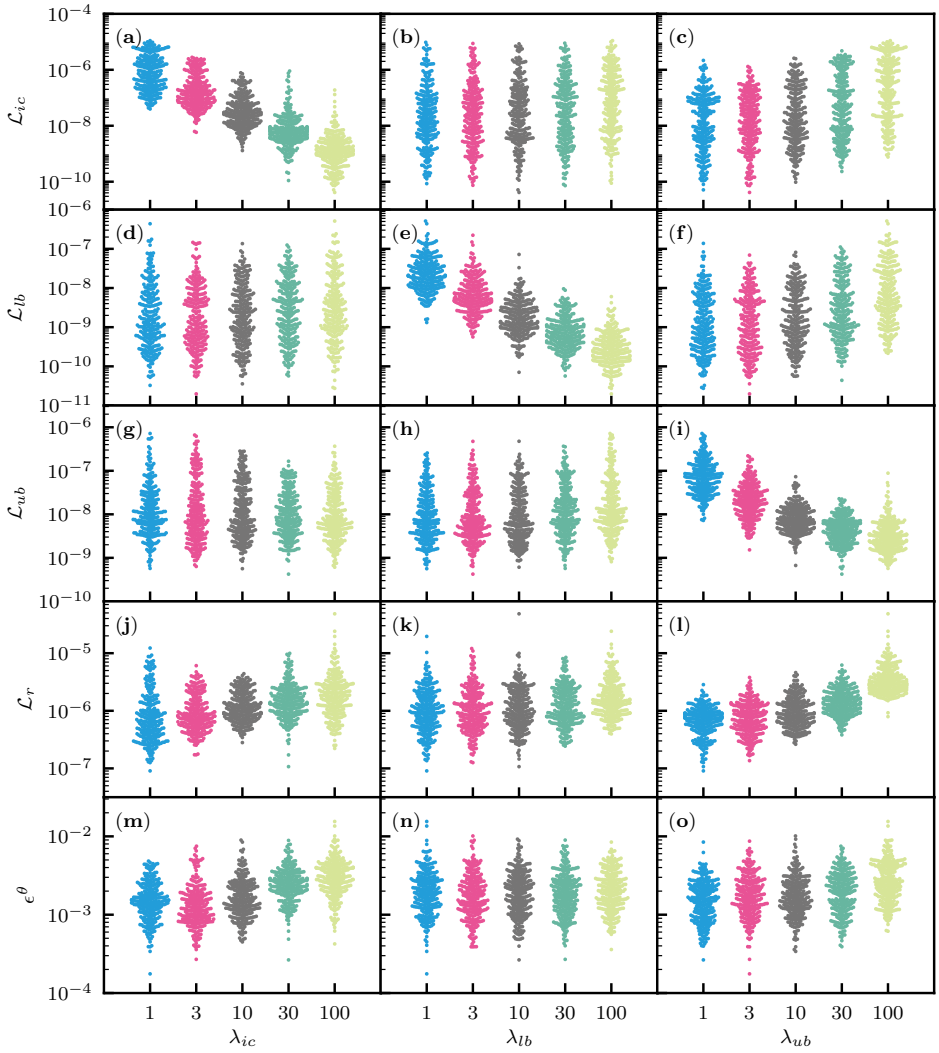

Figure 2.7: Homogeneous soil. The effects of the weight parameters in the loss function (Eq. 2.17) for the initial condition λ_{ic}, the lower boundary condition λ_{lb}, and the upper boundary condition λ_{ub} on the loss terms $\mathcal{L}_{ic}$, $\mathcal{L}_{lb}$, $\mathcal{L}_{ub}$, $\mathcal{L}_r$ for the initial and boundary conditions and the residual of the PDE, respectively, and the relative squared error in terms of volumetric water content ϵ^θ. λ_r was fixed to one.

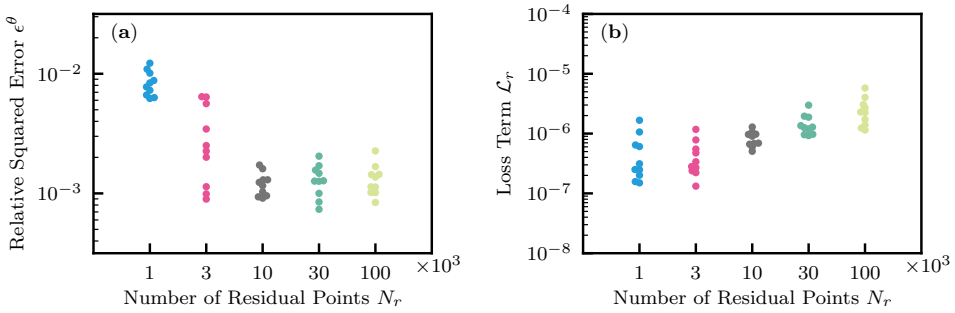

Figure 2.8: Homogeneous soil. (**a**): The effect of the number of residual points N_r on the relative squared error ϵ^θ. (**b**): The effect on the loss term for the residual $\mathcal{L}_r$.

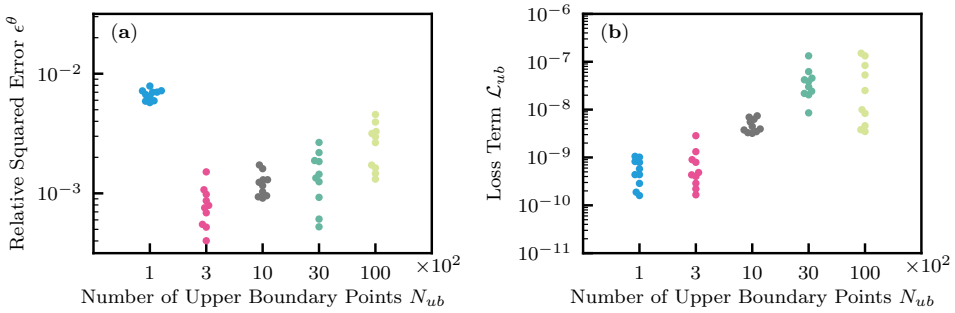

Figure 2.9: Homogeneous soil. (**a**): The effect of the number of upper boundary data N_{ub} on the relative squared error ϵ^θ. (**b**): The effect on the loss term for the upper boundary condition $\mathcal{L}_{ub}$.

increase in the loss term for the upper boundary condition $\mathcal{L}_{ub}$, as shown in 2.9 (**b**). This observation is similar to the case for the residual points and makes it difficult to determine optimal N_{ub}. The difficulty in tuning parameters for PINNs is further explained in the next section.

2.3.1.7 Toward more consistent performance of PINNs

We demonstrated that PINNs can approximate the solution to the RRE with accuracy comparable to the FDM. However, a significant amount of effort was needed to tune the NN architecture and parameters, and those optimal settings depend on problems of interest, which makes it very challenging for PINNs to be consistent numerical solvers of PDEs. To understand why the performance of PINNs is not consistent, we investigated the relationships between $\mathcal{L}_i$ and ϵ^θ by compiling the results from the previous sections. The left column of Fig. 2.10 corresponds to a fixed NN (5 hidden layers with 50 units with the L-LAAF), and the center column is for NNs with varying architecture (the number of hidden layers and units) with the L-LAAF

(from Sect. 2.3.1.4), while the right column is for a fixed NN with varying weight parameters λ_i in the loss function (from Sect. 2.3.1.5). Note that the dimension of the loss function (i.e., the number of adjustable parameters) is the same for the first and third columns because the NN architecture is the same, while the shape or landscape of the loss function is different because of the varying weight parameters λ_i.

As for the first column, the PINN solution was consistent regardless of the random seeds used for the NN initialization. Although there were some differences in the accuracy, a detailed examination of the PINN solutions revealed that the errors mainly came from near the upper boundary. Thus, we concluded that we obtained consistent PINN solutions for this problem once we determined the NN architecture. However, determining NN architecture is still empirical and depends on problems. Based on other studies and our experiences, NNs consisting of less than ten hidden layers appear to be enough to approximate the solution to PDEs. However, the application of PINNs to PDEs with large spatial and temporal domains requires more investigation.

Figure 2.10 (**k**) and (**l**) demonstrates that the residual loss $\mathcal{L}_r$ is well correlated with ϵ^θ, which coincides with the theoretical study by Mishra and Molinaro (2022). This observation might indicate that smaller residual loss $\mathcal{L}_r$ is more important than other loss terms. If this speculation is true, this implies the possibility of transfer learning, where NNs are pre-trained with only the residual of PDEs without initial and boundary conditions and later fine-tuned by them, which could drastically reduce the computational work and needs more investigation.

2.3.2 Heterogeneous Soil

In this section, we simulated a one-dimensional infiltration into a two-layered soil with a length of 20 cm. Because each layer has a distinct saturated hydraulic conductivity K_s, the solution to the RRE is not differentiable at the boundary of the layers. Thus, we implemented the domain decomposition method (see Sect. 2.2.4.4) by dividing the spatial domain into the upper domain Ω_U ($-10 \leq z \leq 0$ cm) and the lower domain Ω_L ($-20 \leq z \leq 10$ cm). NNs $\mathcal{N}_U(z, t; \Theta_U)$ and $\mathcal{N}_L(z, t; \Theta_L)$ were assigned to Ω_U and Ω_L, respectively. The two NNs interact with each other through the interface conditions described in Sect. 2.2.4.4 and were trained simultaneously, although separate training is also possible, as in Jagtap and Karniadakis (2020).

We compared the PINN solution with an FEM solution obtained by HYDRUS-1D (Šimůnek et al., 2013). FEMs are similar to PINNs in that both methods use some basis functions to approximate the solution of PDEs. While PINNs use NNs as the basis function, the FEM implemented in HYDRUS-1D uses a linear finite element as the basis function. Although HYDRUS-1D implements a variable time step, we used a constant time for the comparison. Because WRCs and HCFs defined by Eq. 2.5 and 2.6 are not implemented in HYDRUS-1D, we used a lookup table feature to provide HUDRUS-1D with the WRC and HCF manually.

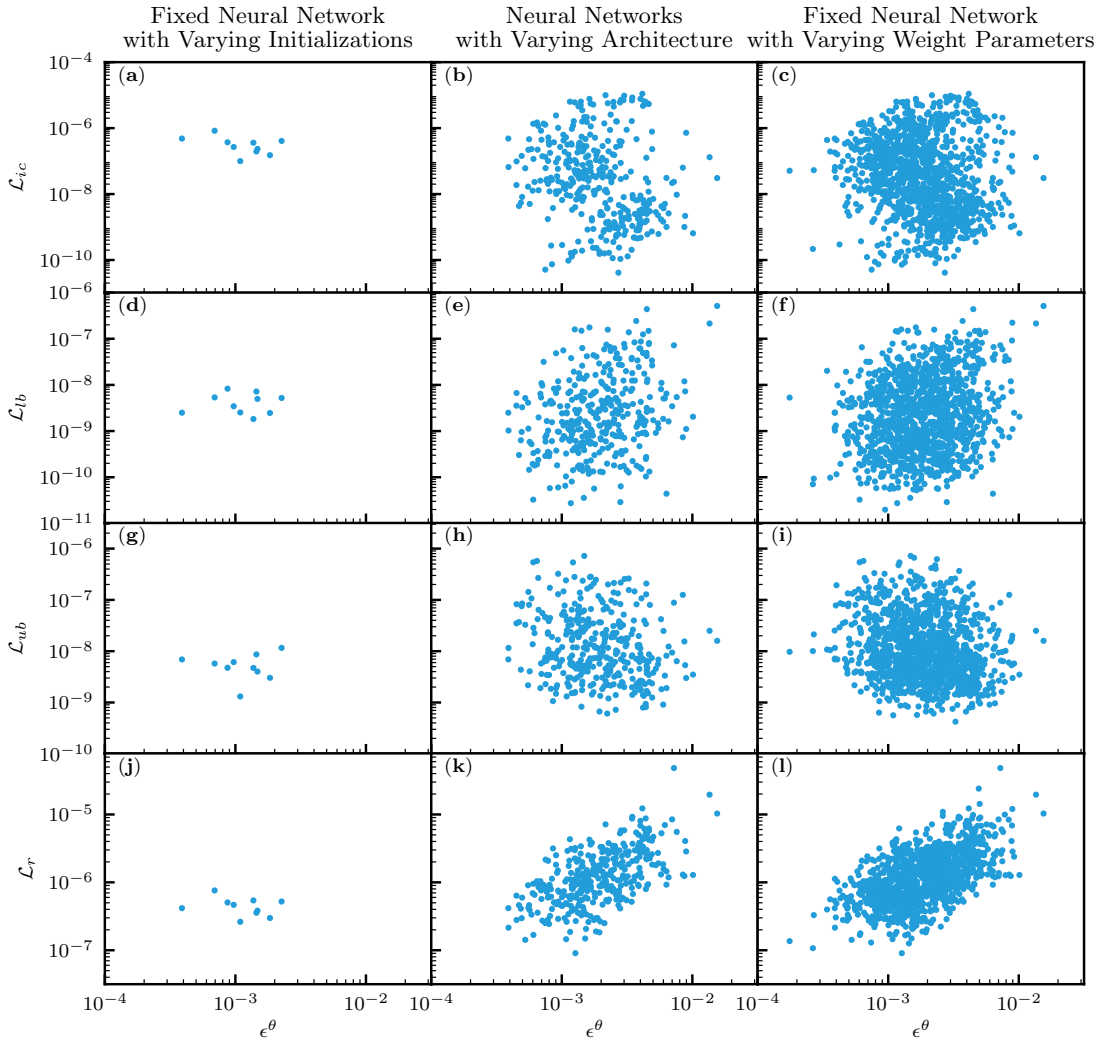

Figure 2.10: Homogeneous soil. The relationships between loss terms $\mathcal{L}_i$ and the relative squared error in terms of volumetric water content ϵ^θ for a fixed neural network (NN) with different NN initializations (left column), NNs with varying architecture (center column), and a fixed NN with varying weight parameters λ_i in the loss function (right column). $\mathcal{L}_{ic}$, $\mathcal{L}_{lb}$, $\mathcal{L}_{ub}$, and $\mathcal{L}_r$ are the loss terms for the initial condition, lower and upper boundary conditions, and the residual of the PDE, respectively.

2.3.2.1 Problem Setup

WRCs and HCFs relationships for the soils are the same for the homogeneous case. The saturated conductivity K_s is 10.0 and 1.0 cm s^{-1} for the upper layer (from $z = -10$ cm to $z = 0$ cm) and the lower layer (from $z = -20$ cm to $z = -10$ cm), respectively. Other parameters θ_s, θ_r, and α for the two layers as well as the initial and boundary conditions are the same as the homogeneous case.

2.3.2.2 Characteristics of PINN solution

Figure 2.11 (**a**) shows the PINN solution with the analytical solution introduced in Sect. 2.2.2.2. Both NNs $\mathcal{N}_U$ and $\mathcal{N}_L$ consisted of 5 hidden layers with 50 units with the L-LAAF, and β was set to one. Randomly sampled 10000 residual points and equally spaced 101 initial data points were used for both NNs. The upper and lower boundary conditions were given as the case for the homogeneous soil. To connect the two NNs, randomly sampled 1000 points were used for the three interface continuity conditions: the water flux $\mathcal{L}_{I_q}$ (Eq. 2.27); the residual $\mathcal{L}_{I_r}$ (Eq. 2.28); the NN output $\mathcal{L}_{I_\mathcal{N}}$ (Eq. 2.29). All the weight parameters in the loss function λ_i were set to ten while λ_r for the lower layer was set to one. Figure 2.11 (**b**) showed the FEM solution obtained using HYDRUS-1D with $dz = 0.1$ cm and $dt = 0.01$ h, which is comparable to the temporal resolution of the upper boundary data points given to the PINNs. The PINN solution was superior to the HYDRUS-1D solution ($\epsilon^\theta = 3.99 \times 10^{-3}$ for the PINN and $\epsilon^\theta = 1.67 \times 10^{-2}$ for the FEM solution, respectively), while the HYDRUS-1D solution underestimated the volumetric water content in the upper layer near the boundary, which coincides with Brunone et al. (2003). This is because only the matric potential continuity is guaranteed in HYDRUS-1D, while both matric potential and water flux continuity conditions were imposed for the PINNs. Also, HYDRUS-1D consistently overestimated the volumetric water content at the wetting front in the lower layer, while consistent errors were not observed for the PINN solution. From this comparison, we concluded that PINNs with domain decomposition can approximate the solution of the RRE for a two-layered soil with discontinuous hydraulic conductivity.

2.3.2.3 Training PINNs

The left column of Fig. 2.12 shows the evolution of the loss terms. All the loss terms for both layers remained higher than those for the homogeneous case (see Fig. 2.4), which demonstrated the difficulty in training PINNs for the layered soil case. While the Adam algorithm resulted in a well-approximated solution for the homogeneous case, the L-BFGS-B algorithm played an important role for the heterogeneous case, particularly in reducing the loss term for the upper boundary condition $\mathcal{L}_{ub}$ and the residual $\mathcal{L}_r$ for the upper layer. Figure 2.12 (**c**) illustrated that the three interface conditions were satisfied well.

The right column of Fig. 2.12 shows how the adaptive parameters sa for the L-LAAF changed during the training. We expected sa for the lower layer to be

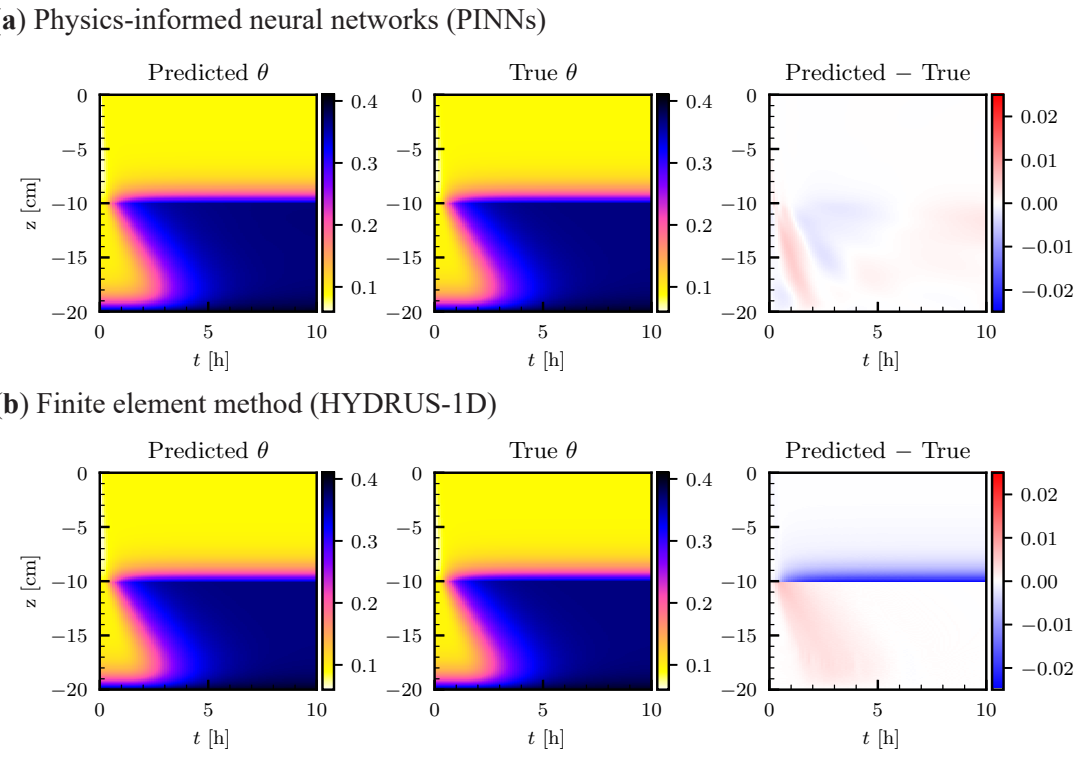

Figure 2.11: Heterogeneous soil. The saturated conductivity K_s is 10.0 and 1.0 cm s^{-1} for the upper and lower layer, respectively. (**a**): Physics informed neural network (PINN) solution in terms of volumetric water content θ [-] obtained by two neural networks of 5 hidden layers with 50 units with the layer-wise adaptive activation function (left column). True analytical solution (center column) is given by Srivastava and Yeh (1991) (see Sect. 2.2.2.2), and the difference between the PINN and true solutions are shown in the right column. (**b**): Numerical solution by a finite element method was obtained with a spatial mesh of $dz = 0.1$ cm and a time step $dt = 0.01$ h using HYDRUS-1D (Šimůnek et al., 2013).

similar to the homogeneous case because the solution for the lower layer is similar to the homogeneous case. However, Fig. 2.12 (**b**) showed that *sa* for the lower layer was much smaller than that for the homogeneous case while similar trends of *sa* for different hidden layers were observed (i.e., layer 2 was the highest). *sa* for all the layers of both layers reached their limiting values after approximately 20000 to 30000 iterations of Adam algorithm, which coincided with the homogeneous case and Jagtap et al. (2020). This indicated that if we can find a better initial guess of *sa* for each layer, we may be able to speed up the training of PINNs, which requires further research.

Figure 2.13 demonstrated how PINNs learned the solution. At the initialization (Fig. 2.13 (**a**)), there is discontinuities at the boundary, which is evident for $t = 10$ h. The continuity conditions and the lower boundary condition were quickly met. The PINNs started to capture the flow of soil moisture at the 20000 iterations of the Adam algorithm (Fig. 2.13 (**e**)), which coincided with when the adaptive parameters *sa* for the L-LAAF reached their limiting values (see the right column of Fig. 2.12). Even at the end of the Adam algorithm, there are large errors in the PINN solution near the surface and wetting fronts in the lower layer (Fig. 2.13 (**g**)). Those errors were further minimized by the L-BFGS-B algorithm (Fig. 2.13 (**h**)). This demonstrated that a second-order method such as the L-BFGS-B algorithm is necessary to train PINNs when the solution to PDEs is complicated.

2.3.2.4 Effects of number of interface points and weight parameters in loss function

We investigated the effects of the number of interface points N_I on PINN performance. The number varied from 100, 300, 1000, 3000, and 10000, and ten different random seeds were used for each setting. Figure 2.14 (**a**), (**b**), and (**c**) showed that the effects on the loss terms for the interface conditions were negligible. The relative squared error ϵ^θ decreased with the number N_I from 100 to 300, but the effect was negligible for larger N_I (see Fig. 2.14 (**d**)). Thus, we concluded that the effects N_I were minor and used 1000 interface points in the following analysis.

We also investigated the effects of the weight parameters λ_i in the loss function. We fixed λ_r for the lower layer to be one and λ_i for the initial condition and the lower boundary condition for the lower layer to be ten while λ_i for the upper layer and the interface conditions were varied from 1, 10, and 100 (i.e., λ_i for the different loss terms in the upper layer are the same). Ten different initializations were conducted. Figure 2.15 (**a**) shows PINNs were more likely trapped by a local minimum of the loss function when λ_i for the upper layer was smaller, indicated by the cloud of the data points. However, the best PINN solution appeared not to be affected by λ_i. Similar observations were made for the effects on the loss terms corresponding to the upper layer (see Fig. A.4). Figure 2.15 (**b**) illustrated the effects of λ_i for the interface conditions. When λ_i were larger, the PINNs produced worse solutions, and it was evident that PINNs suffered from a local minimum of the loss function. Similar conclusions were made from the effects on the loss terms for the upper layer (see Fig.

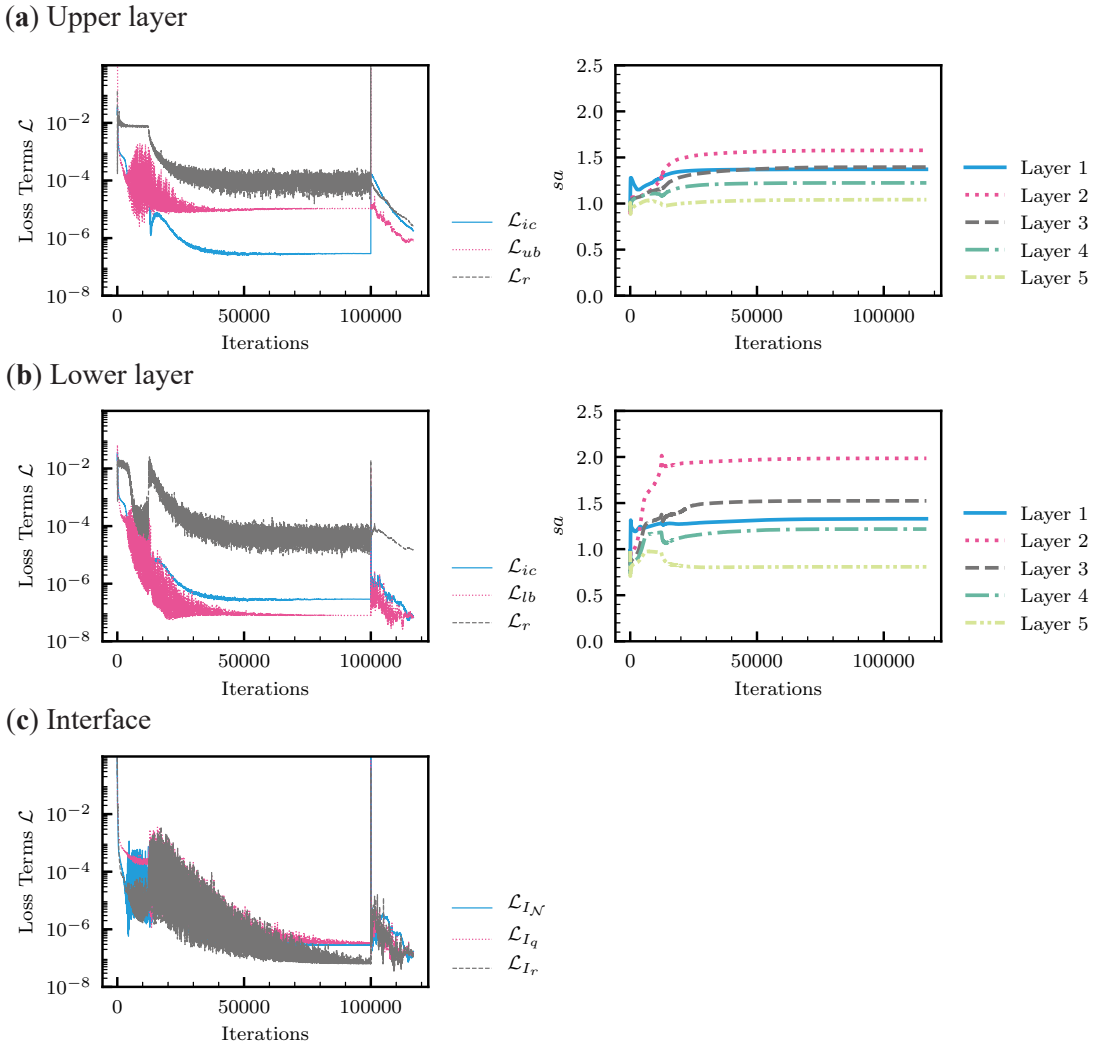

Figure 2.12: Heterogeneous soil. (**a**): The evolution of the loss terms in the loss function (left column) and adaptive parameters *sa* for the layer-wise locally adaptive activation function (Eq. 4.78) for each hidden layer (right column) during the Adam (100000 iterations) and the following L-BFGS-B training for the neural network for the upper layer. Here, Layer 1 is next to the input layer. (**b**) Those for the lower layer. (**c**): The evolution of the loss terms for the interface conditions.

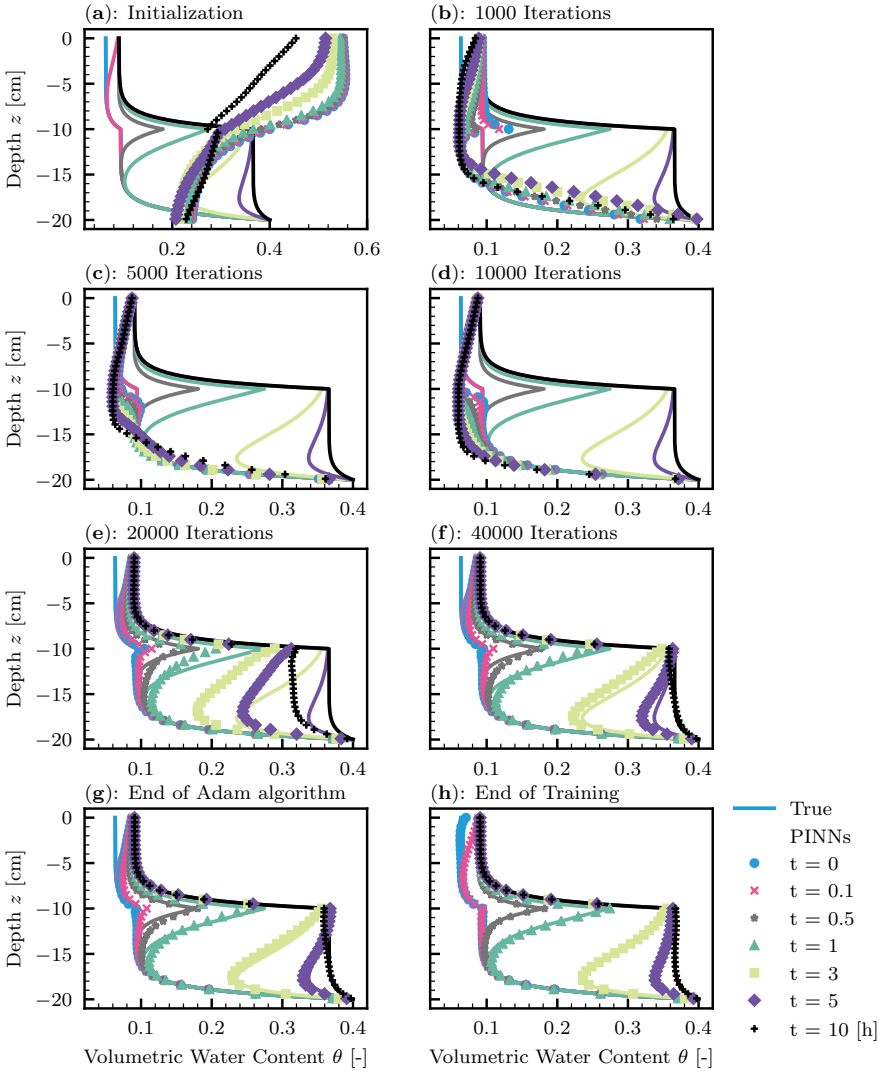

Figure 2.13: Heterogeneous soil. The evolution of the PINN solution during the training. (**a**): Initialization of the PINNs. (**b**) to (**f**): 1000, 5000, 10000, 20000, 40000 iterations of the Adam algorithm. (**g**): The end of 100000 iterations of the Adam algorithm. (**h**): The end of the L-BFGS-B algorithm.

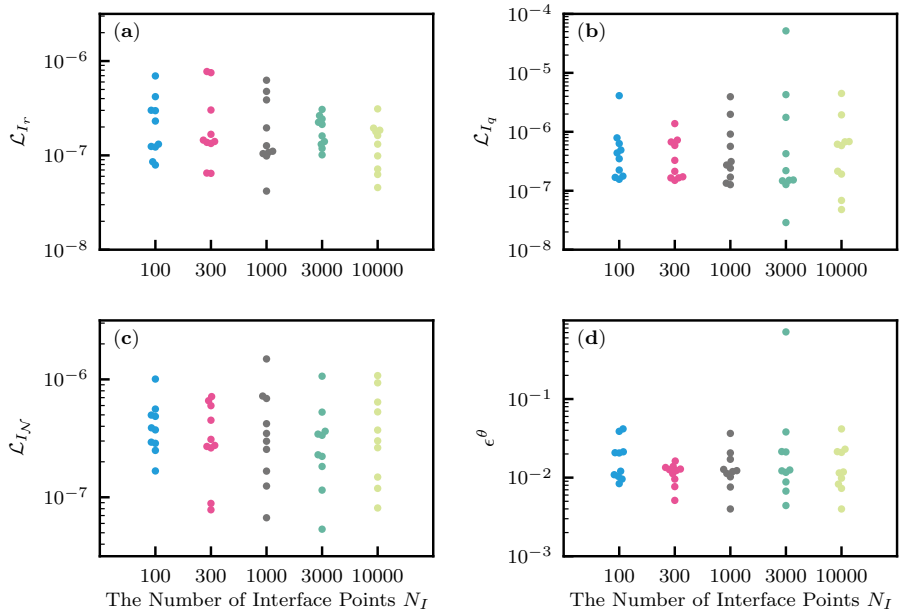

Figure 2.14: Heterogeneous soil. The effects of the number of interface data points N_I. (**a**): The loss term for the continuity of the residual $\mathcal{L}_{I_r}$. (**b**): The loss term for the continuity of the water flux $\mathcal{L}_{I_q}$. (**c**): The loss term for the continuity of the the neural network output $\mathcal{L}_{I_N}$. (**d**): The relative squared error with respect to the volumetric water content ϵ^θ.

A.4). These observations led us to conclude that choosing the right weight parameters λ_i in the loss function is very important and challenging for the heterogeneous case to achieve accurate and consistent solutions to the RRE.

2.4 Inverse modeling

In this section, we demonstrate that inverse modeling can be easily implemented using PINNs. Here, we aim to estimate a surface water flux (i.e., upper boundary condition) from near-surface moisture measurements in a layered soil. Surface water flux is the result of precipitation, evaporation, and surface runoff and thus essential information for land surface modeling and groundwater management. Although rainfall measurements can be used as surface water flux, the measurements are generally spatially scarce and noisy. Therefore, it is important to estimate surface water flux from near-surface soil moisture data. In this line of research, Sadeghi et al. (2019) employed the analytical solution of the linearized RRE, and Li et al. (2021a) proposed a deterministic inverse algorithm to estimate surface water flux given past surface water flux. Brocca et al. (2013) used a simple soil water balance equation to estimate rainfall. These studies assumed soil hydraulic properties are homogeneous

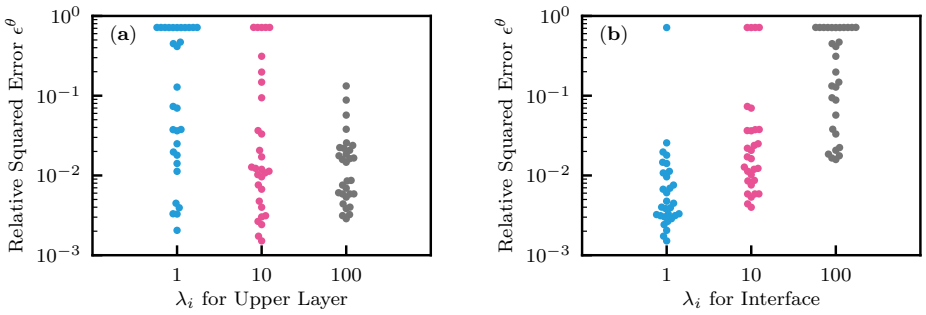

Figure 2.15: Heterogeneous soil. The effects of weight parameters λ_i in the loss function on the performance of PINNs with respect to the relative squared error of the volumetric water content ϵ^θ. (**a**): Upper layer. (**b**): Interface conditions.

(Labolle and Clausnitzer, 1999). Here, we present an inverse framework based on PINNs to estimate surface water flux from near-surface soil moisture measurements in a two-layered soil, as an extension of our previous work (Bandai and Ghezzehei, 2021).

2.4.1 Problem setup

We consider a one-dimensional soil moisture dynamics in a layered soil. The upper layer is a 10 cm depth of loam soil (0 to -10 cm), and the lower layer is a 10 cm depth of sandy loam (-10 to -20 cm). The WRCs and HCFs of the soils for $\psi < 0$ are represented by the van-Genuchten Mualem model (Mualem, 1976; van Genuchten, 1980):

$$\theta = \theta_r + \frac{\theta_s - \theta_r}{(1 + (-\alpha_{VG}\psi)^{n_{VG}})^m}, \tag{2.31}$$

$$K = K_s S_e^l (1 - (1 - S_e^{1/m})^m)^2, \tag{2.32}$$

where θ_r is the residual water content [L^3 L^{-3}]; θ_s is the saturated water content [L^3 L^{-3}]; α_{VG} [L^{-1}] and n_{VG} [-] determine the shape of the WRC; K_s is the saturated hydraulic conductivity [L T^{-1}]; l is the tortuosity parameter; $m = 1 - 1/n_{VG}$; S_e is the effective saturation defined as

$$S_e := \frac{\theta - \theta_r}{\theta_s - \theta_r}. \tag{2.33}$$

The parameters for the two soils were set in the following way: $\theta_r = 0.078$, $\theta_s = 0.43$, $n_{VG} = 1.56$, $K_s = 1.04$, and $l = 0.5$ for the loam soil (upper layer); $\theta_r = 0.065$, $\theta_s = 0.41$, $n_{VG} = 1.89$, $K_s = 4.42$, and $l = 0.5$ for the sandy loam soil (lower layer). The lower boundary is a constant pressure head set to $\psi = -1000$ cm, and the upper boundary condition is a variable surface water fluxes as follows: -0.3 cm h^{-1} from

$t = 0$ to $t = 8$ h; 0.02 cm h^{-1} from $t = 8$ to $t = 12$ h; -0.2 cm h^{-1} from $t = 12$ to $t = 20$ h. The positive and negative values represent evaporation and infiltration, respectively. The initial condition was set to $\psi = -1000$ cm for all the depths. To generate synthetic data for the abovementioned scenario, we employed HYDRUS-1D (Šimůnek et al., 2013) to compute the numerical solution of the RRE with $dt = 0.0001$ h and $dz = 0.02$ cm. The numerical solution by HYDRUS-1D is not necessarily accurate because it may contain undesirable numerical errors near the interface, as observed in Section 3.2 although we confirmed the global mass balance of the solution. We further added a Gaussian noise with a mean of 0 and a standard deviation of 0.005 to the numerical solution. We sampled the simulated noisy synthetic data at predetemined locations to mimic soil moisture measurements by in-situ sensors. We tested three patterns of the measurement locations z_m [cm]: $z_m \in \{-5, -15\}$; $z_m \in \{-3, -7, -13, -17\}$; $z_m \in \{-1, -5, -9, -13, -17\}$. The temporal resolution of the measurements was 0.1 h.

2.4.2 PINNs inverse solution

To infer the surface water flux upper boundary condition, we constructed PINNs with domain decomposition. The two NNs consisted of 5 hidden layers with 50 units, as in Sect. 2.3.2, and $\beta = 0$ was used for the output of both NNs (Eq. 2.15). Unlike the forward modeling, the initial and boundary data points were not used, and the sampled synthetic data points and randomly sampled collocations points were only used to train the NNs. Therefore, the loss function is the sum of the loss term for the measurement data $\mathcal{L}_m$, the residual of the RRE $\mathcal{L}_r$, and the three interface conditions ($\mathcal{L}_{I_r}$, $\mathcal{L}_{I_q}$, and $\mathcal{L}_{I_\mathcal{N}}$). As for the weight parameters λ_i for each loss term, $\lambda_i = 10$ for the measurement data for both layers and the residual loss for the upper layer and $\lambda_i = 1$ for the interface conditions, while $\lambda_i = 1$ for the residual loss for the lower layer as a reference. We tested ten different NN initializations for each measurement scheme, and thus a total of 30 simulations were conducted. Note that the surface water flux $i(t)$ was estimated by evaluating Eq. 2.14 with the solution of the RRE by PINNs.

Figure 2.16 showed the best-recovered solution and estimated surface upper boundary condition for the three measurement schemes. As expected, more accurate recovered solutions were obtained from dense soil moisture measurements (see also Fig. A.7). However, interestingly, NNs more likely were trapped by bad solutions when more measurement data were given (see Fig. A.7). This means a large amount of data does not necessarily lead to a good performance of PINNs because large data make the training of PINNs more difficult. This is a practically important point and requires further investigation. As for the two measurement location scheme (Fig. 2.16 (**a**)), the wetting front reached the top measurement point ($z_m = -5$ cm) at approximately $t = 3$ h, which coincided with when the estimated surface water flux i was reasonable. After that time, both the recovered solution and estimated surface water flux were quite reasonable. Similar trends were observed for the other two measurement schemes (Fig. 2.16 (**b**) and (**c**)). This suggested that we need soil moisture

measurement data closer to the surface ($z = 0$ cm) to capture the wetting front and the infiltration rate. Figure S8 in the Appendix showed the evolution of the loss terms $\mathcal{L}$ corresponding to the measurement data, residual, and the interface conditions. Although the direct comparison between the forward and inverse modeling is not possible because the problem settings are different, we observed smaller residual loss terms $\mathcal{L}_r$ for the inverse modeling. This observation and our experiences indicate that PINNs are more effective for the inverse problem than the forward modeling because data points inside the spatial and temporal domain are more informative than initial and boundary conditions for NNs to find the solution to PDEs.

We note that soil hydraulic parameters are known for both layers in this test, which is not the case for field applications. Depina et al. (2021) implemented PINNs with a global optimization algorithm to estimate the van-Genuchten parameters of a homogeneous soil (K_s, α_{VG}, and n_{VG}) from soil moisture measurements, and the framework was tested against for both synthetic and laboratory infiltration experiment data. They demonstrated that PINNs with a global optimization algorithm could determine the van-Genuchten parameters for a homogeneous soil. In fact, the current study was motivated by the need to verify a PINN approach to estimate such soil hydraulic parameters of a layered soil for field applications. Our next research objective is to implement PINNs that can estimate both the upper surface boundary condition and soil hydraulic parameters (e.g., van-Genuchten parameters) for layered soils and test them with soil moisture and surface water flux data measured in a lysimeter. Note that the estimation of surface water flux was reasonable when the value is positive (i.e., evaporation) in the example, but it would probably not be the case for field applications because evaporation requires coupled heat and water transport models. Applying PINNs to multi-physics in unsaturated hydrology is also our next research step.

2.5 Advantages and disadvantages of PINNs

Regardless of the potential of PINNs to solve PDEs, several studies reported their failures and limitations (Fuks and Tchelepi, 2020; Sun et al., 2020; Wang et al., 2021). Although there are some theoretical studies on the convergence and error analysis of PINNs (e.g., Mishra and Molinaro, 2022; Shin et al., 2020), theoretical understanding of PINNs is still in its infancy (Karniadakis et al., 2021). We summarize the advantages and disadvantages of PINNs compared to traditional numerical methods (e.g., finite difference, finite element, and finite volume methods) to potentially use the method to solve essential questions in hydrology, including large-scale forward and inverse modeling.

One main drawback of PINNs for forward modeling is their computational time. For our case studies, it took approximately 30 and 90 min for the homogeneous and heterogeneous forward modeling using a desktop computer with GPU (NVIDIA GeForce RTX 2060), while it took less than 1 min for HYDRUS-1D to solve the heterogeneous problem. PINNs might be more competitive for large-scale hydrology

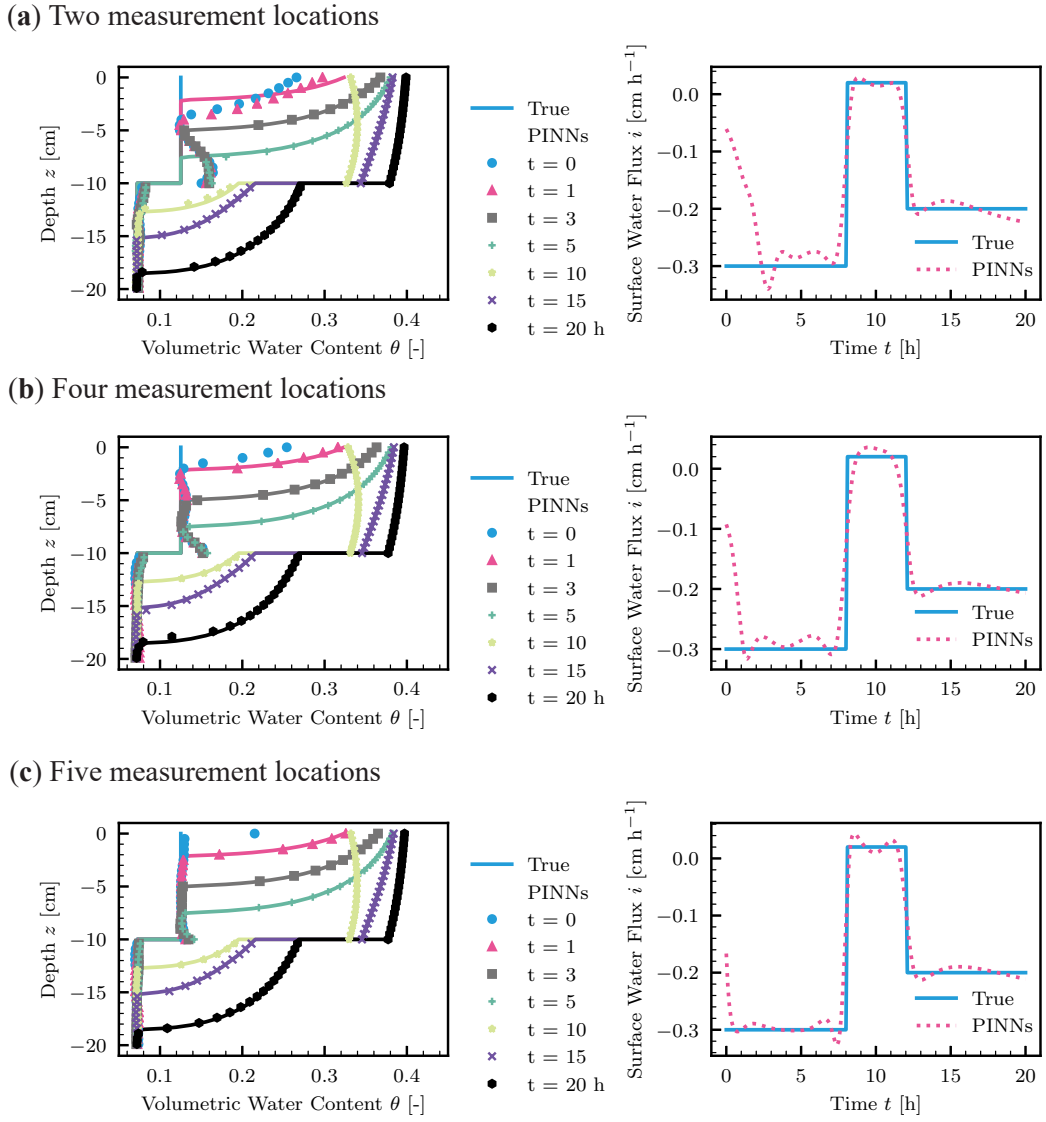

Figure 2.16: Inverse modeling to estimate surface water flux from soil moisture measurements in a layered soil (upper layer: loam soil; lower layer: sandy loam soil). True solution generated by HYDRUS-1D and the recovered solution by PINNs (left column) and the true and estimated upper surface water flux boundary condition (right column) for different measurement locations z_m [cm]. (**a**): $z_m \in \{-5, -15\}$. (**b**): $z_m \in \{-3, -7, -13, -17\}$. (**c**): $z_m \in \{-1, -5, -9, -13, -17\}$.

problems, which needs further investigation.

For forward modeling, PINNs can treat initial and boundary conditions as data points. This feature is advantageous over traditional numerical methods because the accurate initial and boundary conditions are virtually impossible to obtain in practical conditions. On the other hand, traditional methods can take into account the uncertainties of the initial and boundary conditions through the Bayesian approach, while it requires solving the forward problem many times. Although uncertainty quantification through PINNs is open and challenging questions (Psaros et al., 2022), the capability of PINNs to deal with noisy and incomplete initial and boundary conditions is noteworthy.

Traditional numerical methods require mesh generation, which can be tedious when the spatial domain is complicated. On the other hand, PINNs can be easily modified to accommodate such complicated geometries (Raissi et al., 2020). However, it is challenging to correctly impose boundary conditions on PINNs while they are imposed softly in the loss function in this study. Regarding hydrology application, system-dependent boundary conditions such as ponding and evaporation conditions are challenging to implement because PINNs solve PDEs in spatial and temporal domains simultaneously, rather than sequentially, as in traditional time-stepping methods. This difficulty may be a technical issue, but the loss function would be more complicated with such system-dependent boundary conditions, and thus training NNs would be more difficult.

One of the main challenges of PINNs is training PINNs for large-scale modeling. In particular, a long-term simulation such as wetting and drying cycles requires PINNs to approximate very complicated functions. We may sequentially train PINNs in time but lose the ability of PINNs to solve PDEs simultaneously in space and time, and numerical and optimization errors accumulate with time stepping.

The application of PINNs to multi-scale and multi-physics problems is currently challenging, although there are some pioneering studies in hydrology (He et al., 2020). It is known that the solution of PINNs to a multi-scale problem is not always accurate, even for simple problems, because NNs tend to learn "easy" or low-frequency parts of the solution (Wang et al., 2022). Although the study was only concerned with water flow in unsaturated soils, near-surface soil moisture dynamics is essentially coupled heat and water transport. Therefore, further research is needed for the application of PINNs to multi-physics simulations in unsaturated soils.

One advantage of PINNs specific to the RRE is that there is no need for temporal discretization, which results in mass balance issue (Celia et al., 1990). Also, PINNs solutions are differentiable and thus can be used to derive water flux easily without post-processing, as in Scudeler et al. (2016). Furthermore, PINNs can store the solutions of PDEs efficiently with a smaller number of degrees of freedom, particularly for high dimensions (Karniadakis et al., 2021). Those merits can make PINNs a good candidate for a numerical solver of large-scale modeling based on the RRE. Nevertheless, these advantages depend on how accurately PINNs satisfy the RRE.

In terms of inverse modeling, PINNs have some interesting features. First, PINNs do not have to solve the forward modeling to solve the inverse problem. On the

other hand, standard inverse methods require solving the forward modeling many times to adjust parameters of interest. This feature makes the computation of PINNs efficient. However, as shown in the study, PINNs do not precisely impose PDEs constraints as traditional methods, where the forward modeling is actually solved. Further research is needed to minimize the residual loss term so that known physics is precisely imposed. Second, when estimating boundary conditions as in the study or initial condition from data, PINNs do not require the discretization of those target functions. In traditional methods, it is common to represent the target functions as a linear combination of some basis functions (e.g., finite elements) and estimate the coefficients of the basis functions. In PINN framework, such discretization is not necessary, and those target function values can be evaluated directly from NNs. Overall, although PINNs have interesting and attractive characteristics, fully utilizing the potential requires further research. In the next section, future perspectives of PINNs are mentioned.

2.6 Conclusions and future perspectives

We presented a numerical method based on neural networks (NNs), called physics-informed neural networks (PINNs), to solve the Richardson-Richards equation (RRE) to simulate water flow in unsaturated homogeneous and heterogeneous soils. We tested recently proposed PINN algorithms on our problems and found that the layer-wise locally adaptive activation function (L-LAAF) developed by Jagtap et al. (2020) was effective. The L-LAAF changes the slope and the linear regime of the activation functions in NNs and helps PINNs approximate the solution of the RRE well. First, we tested the PINN approach for the homogeneous soil case. By comparing the PINN solution to the analytical solution by Srivastava and Yeh (1991) and the numerical solution by a finite difference method, we demonstrated that "well-trained" PINNs can be competitive in terms of accuracy and memory efficiency. However, training PINNs requires significant efforts to tune various parameters of NNs, including NN architecture and weight parameters in the loss function. We systematically investigated the effects of those parameters on the performance of PINNs and demonstrated that those interrelated effects make PINN approach less consistent. Although some automatic but empirical algorithms to tune those parameters improved the performance of PINNs to some extent, it was difficult for PINNs to consistently obtain solutions to the PDE with high accuracy, and the results were strongly dependent on the initialization of NNs. Our empirical but comprehensive observations provide some suggestions on the choice of the parameters, but we do not think they can be applied to various cases, and thus further studies are necessary.

We tested PINN approach for a layered soil, where hydraulic conductivity is discontinuous across the layer boundary. The analytical solution by Srivastava and Yeh (1991) was used to verify the PINN solution. We demonstrated that PINNs with domain decomposition proposed by Jagtap and Karniadakis (2020) successfully approximated the solution of the RRE for a two-layered soil. The comparison with a

finite element method using popular software, HYDRUS-1D (Šimůnek et al., 2013), was made. The PINN solution was superior to HYDRUS-1D for the problem because the interface conditions on the layer boundary were well imposed for the PINN approach but not for HYDRUS-1D. Nevertheless, the study demonstrated that obtaining PINN solutions for the problem with consistent accuracy was challenging because of the difficulty in choosing the right weight parameters in the loss function, which determines the relative importance of physical constraints for the problem (e.g., initial and boundary conditions).

We further applied the PINNs with domain decomposition to the inverse modeling to estimate a water flux upper boundary condition from noisy sparse soil moisture measurements. The inverse modeling was easily formulated by the PINN approach, and the effects of the measurement schemes were studied. The upper boundary condition was reasonably inverted from the noisy data, in particular when measurement data near the soil surface were available. However, our results demonstrated that a large amount of data do not necessarily lead to a good performance of PINNs because training PINNs is more difficult with more data. Further research is needed to make PINNs learn from a larger amount of data and simultaneously determine both soil hydraulic properties and surface water flux for layered soils.

The PINN algorithm presented here is focused on water flow in unsaturated soils. However, there are many situations where we need to simulate saturated-unsaturated flow, such as ponding conditions and interactions with groundwater. Our preliminary tests gave positive results on the extension of PINNs to saturated-unsaturated cases. Readers may refer to the author comment in the interactive public discussion (Bandai and Ghezzehei, 2022a).

PINNs have the potential to solve issues traditional numerical methods cannot solve by leveraging the capability of NNs to approximate complex functions efficiently. However, the mathematical complexities of the forward and inverse problems are lumped into a complicated non-linear, non-convex minimization problem. Whether PINNs can perform well depends on whether we can solve the resulting minimization problem well. Another difficulty comes from the fact that PINNs have an unusual regularization term in the loss function as the form of the residual of PDEs. This term is very different from standard regularization terms such as L1 and L2 regularizations because it contains the derivative of the output of NNs with respect to their input. The mathematical and exploratory investigation of the minimization problem and the regularization term is necessary for further improvements of PINNs. The investigation may include a vast amount of literature on NNs and PDE constrained optimization. There are many methods and findings that have not been well tested against PINNs, including transfer learning, second-order optimization methods, and the correspondence with adjoint-state methods (Petra and Stadler, 2011). We will investigate those areas to improve the understanding of PINNs and use PINNs for large-scale modeling in hydrology.

Aside from PINNs, the latest research trends have been directed toward learning the "operator" of PDEs rather than their solutions given initial and boundary conditions, as in the study. This new research field has been led by two main groups (Lu

et al., 2021a; Li et al., 2021b), and they aim to develop operator learning methods applicable to general PDEs, of course including the ones in hydrology. Do we wait until they develop general PDE simulators or provide a unique perspective in soil physics and hydrology? We need to consider how we contribute to the rapid progress of the fields as domain scientists (Nearing et al., 2021).

2.7 List of abbreviations

FDMs: Finite difference method
FEMs: Finite element methods
GPUs: Graphics processing units
HCF: Hydraulic conductivity function
L-LAAF: Layer-wise locally adaptive activation function
ML: Machine learning
NNs: Neural networks
PDE: Partial differential equation
PINNs: Physics-informed neural networks
RRE: Richardson-Richards equation
WRC: Water retention curve

2.8 List of notation

superscript
$\hat{\cdot}$: prediction except for used for $\hat{\Theta}$
$\cdot^*$: dimensionless for the analytical solutions

subscript
$\cdot_D$: Dirichlet boundary condition
$\cdot_F$: water flux boundary condition
$\cdot_I$: interface
$\cdot_{I_\mathcal{N}}$: interface condition regarding the continuity of neural network output
$\cdot_{I_q}$: interface condition regarding the continuity of water flux
$\cdot_{I_r}$: interface condition regarding the continuity of the residual of the RRE
$\cdot_{I_\psi}$: interface condition regarding the continuity of water potential
$\cdot_{ic}$: initial condition
$\cdot_{lb}$: lower boundary condition
$\cdot_L$: lower layer
$\cdot_m$: measurement data
$\cdot_r$: residual
$\cdot_{ub}$: upper boundary condition

$\cdot_U$: upper layer

alphabet
a: the collection of trainable parameters for adaptive activation functions for a neural network
$a^{[k]}$: trainable parameter for the element-wise non-linear activation function
b: the collection of bias vectors for a neural network
$\mathbf{b}^{[k]}$: bias vector for the kth hidden layer
dt: time step for finite difference and finite element solutions [T]
dz: spatial mesh for finite difference and finite element solutions [L]
$g(z)$: initial condition
$h(z,t)$: Dirichlet boundary condition
H: total water head [L]
$\mathbf{h}^{[k]}$: vector for the kth hidden layer
$i(z,t)$: water flux boundary condition
K: hydraulic conductivity [L T^{-1}]
K_s: saturated hydraulic conductivity [L T^{-1}]
l: tortuosity parameter [-]
L: the number of hidden layers
$\mathcal{L}$: loss function
$\mathcal{L}_i$: loss term for i constraints
m: a van-Genuchten parameter
$n^{[k]}$: dimension of a vector corresponding to the kth hidden layer
n_{VG}: van-Genuchten parameter [-]
n^x: dimension of input vector $\mathbf{x}$
n^y: dimension of output vector $\hat{\mathbf{y}}$
N_i: the number of points for i constraints
$\mathcal{N}$: neural network functions
o: output functions
q: water flux in the vertical direction (positive downward) [L T^{-1}]
$\mathbf{q}$: water flux in three dimensions [L T^{-1}]
q_A: constant water flux at the surface to determine the initial condition of the analytical solutions [L T^{-1}]
q_B: constant water flux at the surface used in the analytical solutions [L T^{-1}]
$\hat{r}$: the residual of the RRE
s: fixed scaling factor for adaptive activation functions
S: source term [T^{-1}]
t: time [T]
T: final time [T]
$\mathbf{W}$: the collection of the weight matrices for a neural network
$\mathbf{W}^{[k]}$: weight matrix for the kth hidden layer
$\mathbf{x}$: input vector
$\hat{\mathbf{y}}$: output vector
z: vertical coordinate or elevation head (positive upward) [L]

Z: the vertical length of a soil [L]

Greek alphabet
α_G: pore-size distribution parameter [L^{-1}]
α_{VG}: van-Genuchten parameter [L^{-1}]
β: fixed parameter for the output of neural networks
ϵ^θ: relative squared error in terms of volumetric water content
θ: volumetric water content [L^3 L^{-3}]
θ_r: residual volumetric water content [L^3 L^{-3}]
θ_s: saturated volumetric water content [L^3 L^{-3}]
Θ: neural network parameters
$\hat{\Theta}$: update of neural network parameters for each iteration of optimization algorithms
κ_n: infinite sequence to compute the analytical solution
λ_i: weight parameters in the loss function
σ: element-wise non-linear activation function
ψ: water potential in soils [L]
ψ_{lb}: water potential at the bottom boundary [L]
Ω: spatial domain
$\partial\Omega$: spatial boundary

others
$:=$: equal by definition
∇: nabla

References for Chapter 2

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., J., D., Devin, M., Ghemawat, S., I., G., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous distributed systems.

Assouline, S. and Or, D. (2013). Conceptual and parametric representation of soil hydraulic properties: A review. *Vadose Zone J.*, 12(4):1–20.

Babaeian, E., Sadeghi, M., Jones, S. B., Montzka, C., Vereecken, H., and Tuller, M. (2019). Ground, proximal, and satellite remote sensing of soil moisture. *Rev. Geophys.*, 57:530–616.

Bandai, T. and Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resour. Res.*, 57:e2020WR027642.

Bandai, T. and Ghezzehei, T. A. (2022a). Author comment on "Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition" by Toshiyuki Bandai and Teamrat A. Ghezzehei.

Bandai, T. and Ghezzehei, T. A. (2022b). *DD-PINNS-RRE*. Zenodo [code].

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18:1–43.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In Montavon, G., Orr, G. B., and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, second edition.

Brocca, L., Moramarco, T., Melone, F., and Wagner, W. (2013). A new method for rainfall estimation through soil moisture observations. *Geophys. Res. Lett.*, 40(5):853–858.

Brooks, R. H. and Corey, A. T. (1964). Hydraulic properties of porous media. *Hydrology Papers 3, Colorado State University*.

Brunone, B., Ferrante, M., Romano, N., and Santini, A. (2003). Numerical simulations of one-dimensional infiltration into layered soils with the Richards equation using different estimates of the interlayer conductivity. *Vadose Zone J.*, 2(2):193–200.

Buckingham, E. (1907). *Studies on the movement of soil moisture*. Bull. 38. USDA, Bureau of Soils.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *J. Sci. Comput.*, 16(5):1190–1208.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res.*, 26(7):1483–1496.

Clément, J.-B., Golay, F., Ersoy, M., and Sous, D. (2021). An adaptive strategy for discontinuous Galerkin simulations of Richards' equation: Application to multi-materials dam wetting. *Adv. Water Resour.*, 151:103897.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems*, 2:303–314.

Depina, I., Jain, S., Valsson, S. M., and Gotovac, H. (2021). Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow. *Georisk*.

Dijkema, J., Koonce, J. E., Shillito, R. M., Ghezzehei, T. A., Berli, M., van der Ploeg, M. J., and van Genuchten, M. T. (2017). Water distribution in an arid zone soil: Numerical analysis of data from a large weighing lysimeter. *Vadose Zone J.*

Farthing, M. W. and Ogden, F. L. (2017). Numerical solution of Richards' equation: A Review of advances and challenges. *Soil Sci. Soc. Am. J.*, 81:1257–1269.

Fuks, O. and Tchelepi, H. A. (2020). Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1):19–37.

Gardner, W. R. (1958). Some steady-state solutions of the unsaturated moisture flow equation with application to evaporation from a water table. *Soil Sci.*, 85(4):228–232.

Ghezzehei, T. A., Kneafsey, T. J., and Su, G. W. (2007). Correspondence of the Gardner and van Genuchten-Mualem relative permeability function parameters. *Water Resour. Res.*, 43(10):1–7.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, Sardinia, Italy, May 13 - 15.

He, Q., Brajas-solano, D., Tartakovsky, G., and Tartakovsky, A. M. (2020). Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Adv. Water Resour.*, 141:103610.

Hornik, K. (1991). Approximation capabilities of multilayer neural network. *Neural Networks*, 4(1991):251–257.

Jagtap, A. D. and Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun. Comput. Phys.*, 28(5):2002–2041.

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Comput. Phys.*, 404:109136.

Jin, X., Cai, S., Li, H., and Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Comput. Phys.*, 426.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nat. Rev. Phys.*, 3:422–440.

Kingma, D. P. and Ba, J. B. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd international conference on learning representations*, San Diego, CA, USA, May 7 - 9.

Kosugi, K. (1996). Lognormal distribution model for unsaturated soil hydraulic properties. *Water Resour. Res.*, 32(9):2697–2703.

Labolle, E. M. and Clausnitzer, V. (1999). Comment on Russo [1991], Serrano [1990, 1998], and other applications of the water-content-based form of Richards' equation to heterogeneous soils. *Water Resour. Res.*, 35(2):605–607.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw. Learn. Syst.*, 9(5):987–1000.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In Montavon, G., Orr, G. B., and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade*, pages 9–48. Springer, second edition.

Li, N., Yue, X. Y., and Wang, W. K. (2021a). Inverse estimation of spatiotemporal flux boundary conditions in unsaturated water flow modeling. *Water Resour. Res.*, 57:e2020WR028030.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021b). Fourier neural operator for parametric partial differential equations. In *International conference on learning and representations*, Vienna, Austria, May 4.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021a). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Mach. Intell.*, 3:218–229.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021b). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

Mishra, S. and Molinaro, R. (2022). Estimates on the generalization error of physics-informed neural networks for approximating PDEs. *IMA J. Numer. Anal.*, 00:1–43.

Mitra, K. and Vohralík, M. (2021). A posteriori error estimates for the Richards equation. *arXiv [preprint]*.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resour. Res.*, 12(3):513–522.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2021). What role does hydrological science play in the age of machine learning? *Water Resour. Res.*, 57:e2020WR028091.

Paniconi, C. and Putti, M. (2015). Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resour. Res.*, 51(9):7090–7129.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *33rd conference on neural information processing systems*, Vancouver, Canada, December 8 - 14.

Petra, N. and Stadler, G. (2011). Model variational inverse problems governed by partial differential equations. Technical report, The Institute for Computational Engineering and Sciences, The University of Texas at Austin.

Philip, J. R. (1969). Theory of infiltration. *Adv. Hydrosci.*, 5:215–296.

Pries, C. E. H., Castanha, C., Porras, R. C., and Torn, M. S. (2017). The whole-soil carbon flux in response to warming. *Science*, 359:1420–1423.

Psaros, A. F., Meng, X., Zou, Z., Guo, L., and Karniadakis, G. E. (2022). Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *arXiv [preprint]*.

Radu, F. A., Pop, I. S., and Knabner, P. (2008). Error estimates for a mixed finite element discretization of some degenerate parabolic equations. *Numer. Math.*, 109:285–311.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Dickstein, J. S. (2017). On the expressive power of deep neural networks. In *34th International Conference on Machine Learning*, Sydney, Australia, August 6 - 11.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Comput. Phys.*, 378:686–707.

Raissi, M., Yazdani, A., and Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367:1026–1030.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333.

Richardson, L. F. (1922). *Weather prediction by numerical process*. Cambridge University Press.

Robinson, D. A., Campbell, C. S., Hopmans, J. W., Hornbuckle, B. K., Jones, S. B., Knight, R., Ogden, F., Selker, J., and Wendroth, O. (2008). Soil moisture measurement for ecological and hydrological watershed-scale observatories: A review. *Vadose Zone J.*, 7:358–389.

Sadeghi, M., Tuller, M., Warrick, A., Babaeian, E., Parajuli, K., Gohardoust, M. R., and Jones, S. B. (2019). An analytical model for estimation of land surface net water flux from near-surface soil moisture observations. *J. Hydrol.*, 570:26–37.

Scudeler, C., Putti, M., and Paniconi, C. (2016). Mass-conservative reconstruction of Galerkin velocity fields for transport simulations. *Adv. Water Resour.*, 94:470–485.

Sheng, W., Zhou, R., Sadeghi, M., Babaeian, E., Robinson, D. A., Tuller, M., and Jones, S. B. (2017). A TDR array probe for monitoring near-surface soil moisture distribution. *Vadose Zone J.*, 16(4):1–8.

Shin, Y., Darbon, J., and Karniadakis, G. E. (2020). On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. *Commun. Computat. Phys.*, 28(5):2042–2074.

Šimůnek, J., Šejna, M., Saito, H., Sakai, M., and van Genuchten, M. T. (2013). *The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably saturated media, Version 4.17*. Department of Environmental Sciences, University of California Riverside, Riverside.

Srivastava, R. and Yeh, T. C. J. (1991). Analytical solutions for one-dimensional, transient infiltration toward the water table in homogeneous and layered soils. *Water Resour. Res.*, 27(5):753–762.

Sun, L., Gao, H., Pan, S., and Wang, J. (2020). Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.*, 361:112732.

Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., and Barajas-Solano, D. (2020). Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resour. Res.*, 56:e2019WR026731.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J.*, 44:892–898.

Vereecken, H., Huisman, J. A., Bogena, H., Vanderborght, J., Vrugt, J. A., and Hopmans, J. W. (2008). On the value of soil moisture measurements in vadose zone hydrology: A review. *Water Resour. Res.*, 44.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17(3):261–272.

Wang, S., Teng, Y., and Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.*, 43(5):A3055–A3081.

Wang, S., Yu, X., and Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *J. Comput. Phys.*, 449:110768.

Chapter 3

Estimating soil hydraulic properties from complete dryness using shortwave infrared imaging and inverse modeling

Abstract. [1]Accurate simulation of soil moisture dynamics for dry conditions is essential for projecting the impact of droughts. A commonly used soil hydraulic model, the van-Genuchten Mualem (VGM) model, is not suitable for describing water retention and hydraulic conductivity for dry conditions. We investigated whether an alternative model, called the Peters-Durner-Iden (Iden) model, better describes soil moisture dynamics for dry conditions. With that aim, we conducted inverse modeling of the Richardson-Richards equation (RRE) with the VGM and the PDI models. Soil moisture and cumulative infiltration data from upward infiltration experiments for seven soils with distinctive textures were used. Soil moisture data measured by a shortwave infrared (SWIR) imaging camera enabled us to monitor the advancement of wetting fronts clearly, where soil hydraulic properties for dry conditions play essential roles. We demonstrated that the RRE with the PDI model captured soil moisture dynamics better than that with the VGM model. However, the inverse modeling was only successful for sandy loam and loam soils, and we faced difficulties for sandy soils due to the violation of the one-dimensional flow assumption and clay-rich soils because of the heterogeneity in soil hydraulic properties.

[1]A preliminary study of the current chapter was presented as Bandai, T., Sadeghi, M., Babaeian, E., Tuller, M., Jones, S. B., Ghezzehei, T. A. Characterization of unsaturated water flow in soils using short-wave infrared imaging through inverse modeling. ASA, CSSA, SSSA International Annual Meeting 2021, Salt Lake City, USA (November 2021).

3.1 Introduction

Faced with droughts in arid regions and their projected increase, we need more accurate simulations of soil moisture dynamics in dry soils. While water flows in large pores for medium to wet conditions, for dry conditions, water molecules are adsorbed onto the surface of soil mineral and organic particles and flow as thin films. Unfortunately, current climate and hydrological models do not consider such processes well.

The situation has also been true for the state of the practice in soil physics. Conceptually, such processes have been well recognized since the beginning of the field (Buckingham, 1907). When simulating soil moisture dynamics using the Richardson-Richards equation (RRE) (Richardson, 1922; Richards, 1931), we need a water retention curve (WRC) or the relation between the volumetric water content θ and the water potential ψ as well as a hydraulic conductivity function (HCF) that relates the water potential ψ to the hydraulic conductivity K. Commonly, the van Genuchten (VG) model (van Genuchten, 1980) is used to represent the WRC of soil, and the statistical model of hydraulic conductivity proposed by Mualem (1976) is combined with the VG model to obtain the HCF. While this van Genuchten-Mualem (VGM) framework has been useful when simulating soil moisture dynamics in many situations, it has been criticized on several grounds, including its poor performance for dry conditions (Hills et al., 1989).

Many parametric models for WRCs, including the VG model, describe WRCs using power-law or S-shaped functions. However, they do not adequately represent WRCs for dry conditions, which exhibit a linear relation on a semi-log scale (Campbell and Shiozawa, 1992; Schneider and Goss, 2012). Another issue is a parameter in the VG model called the residual water content θ_r (Nimmo, 1991). If the VG model is used to describe WRCs for conditions from complete dryness, θ_r must be set to zero, which makes it difficult for the model to fit medium soil moisture data. Also, the VG model lacks physical descriptions of adsorptive water for dry conditions. The Mualem model of hydraulic conductivity also has issues for dry conditions. The Mualem model assumes soil water flow is only driven by capillarity in a bundle of cylindrical tubes and neglects film water flow on soil minerals and corner flow in angular pores (Tuller and Or, 2001).

In the last two decades, several physical or semi-physical models have been proposed to describe WRCs and HCFs accounting for film and corner flows in addition to capillary flow (Tuller and Or, 2001; Peters and Durner, 2008; Lebeau and Konrad, 2010; Zhang, 2011; Peters, 2013; Iden and Durner, 2014; Peters, 2014). Tuller and Or (2001) derived WRCs and HCFs by upscaling a pore-scale model assuming that water is held in an angular pore and silt-shaped spaces (Tuller et al., 1999), and they provided a solid foundation for the physical description of capillary and adsorptive water. However, the ability of the model to fit experimental data was limited due to the probability distribution used to represent the pore-size distribution, although this restriction could be loosened by abandoning a closed-form expression of HCFs (Tuller and Or, 2001). Other models (Lebeau and Konrad, 2010; Zhang, 2011; Peters,

2013) were developed by using empirical models for WRCs and modifying the physical model proposed by Tokunaga (2009) for film water flow. Among them, the model by Peters (2013) is simple enough to be combined with the RRE but physically consistent for both capillary and film water from complete oven-dry to saturation. The model was further improved by Iden and Durner (2014) and Peters (2014) and is referred to as the Peters-Durner-Iden (PDI) model.

The PDI model partitions soil water into capillary and non-capillary water (i.e., film and corner water) and can incorporate vapor flow contributions into the hydraulic conductivity. Thus, the PDI model has a distinctive advantage for dry conditions compared to the VGM model and has been tested in such conditions (Luo et al., 2020; Iden et al., 2021). However, the PDI model needs one more parameter to describe the saturated conductivity for non-capillary water K_{snc}. There are no direct methods to measure this parameter, and it needs to be estimated from experimental data. However, measuring hydraulic conductivity for dry conditions, where the effect of non-capillary water is eminent, is time-consuming and prone to noise caused by temperature and evaporation (Tokunaga, 2009). Thus, inverse modeling of the RRE with soil moisture dynamics data is more promising. However, such studies are limited and only concerned with evaporation, not infiltration.

To fill the research gap, we estimated WRCs and HCFs from oven-dry to complete saturation by inverse modeling of the RRE with the PDI model using upward infiltration experimental data for seven soils with distinctive textures by Sadeghi et al. (2017). Sadeghi et al. (2017) used a shortwave infrared (SWIR) imaging camera with a spatial resolution (sub mm-scale) to measure volumetric water content based on the physical model by Sadeghi et al. (2015). The SWIR framework provides higher-resolution soil moisture data on both space and time scales than conventional methods, such as time domain reflectometry (TDR). This advantage enables us to monitor the advancement of wetting fronts more accurately, where we expect to observe the effect of non-capillary flow. Using a similar approach, Babaeian et al. (2021) estimated the VGM parameters for the wetting branch using inverse modeling of the RRE and SWIR water content estimates, but the fitting was not sufficient for fine-textured soils and the dry end of the WRC. We expect that using the PDI model improves the fitting for dry conditions. We compared the performance of the PDI model with that of the VGM model and demonstrated that the PDI model is better at capturing soil moisture dynamics for dry conditions. In addition, we observed the heterogeneity of soil hydraulic properties in the artificially packed soils and discussed the effects on the inverse modeling.

3.2 Methods

3.2.1 Upward infiltration experimental data with shortwave infrared imaging

We used upward infiltration experimental data measured at Utah State University for the inverse modeling. Because the experimental data were presented in Sadeghi et al. (2017), we briefly summarize the experimental setup and their framework to use shortwave infrared (SWIR) imaging to obtain soil moisture data.

Sadeghi et al. (2017) conducted constant-head upward infiltration experiments for seven different soils collected in Arizona. The physical and chemical properties of the soils were measured at the University of Arizona and summarized in Table 3.1. Fig. 3.1 shows the water retention curve (WRC) for each soil measured by using Tempe Cells (Soilmoisture Equipment Corp., USA) and WP4 Dewpoint Potentiometer (METER Group, Inc. USA). The saturated hydraulic conductivity was measured by constant head method using a cylinder with a diameter of 66 mm and a height of 69 mm. Bulk densities for the upward infiltration experiments, WRC, and saturated hydraulic conductivity measurements are slightly different due to packing, as shown in Table 3.1. We refer the water retention data (Fig. 3.1) and the saturated conductivity measurement data as the independently measured WRC and K_s, respectively, to avoid confusing them with WRCs and HCFs estimated from the upward infiltration experimental data. Note that Sadeghi et al. (2018) measured particle size distributions and mineral compositions for the soils, except for AZ2 soil.

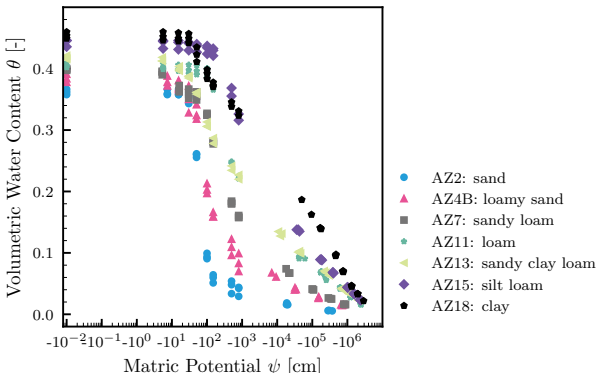

Figure 3.1: Water retention curves (WRCs) for the soils used in the upward infiltration experiments. These data were measured independently from the upward infiltration experiments.

Each soil was oven-dried, sieved with a 2 mm sieve, and uniformly packed into a Hele-Shaw cell (referred to as the experimental box hereafter) with a 10 cm width, a 10 cm height, and a 5 cm thickness (transmission coefficient: 98%). A metal mesh

Table 3.1: The physical and chemical properties of soils used in the study. The bulk densities are for the upward infiltration experiments (UI), the water retention curve measurements (WRC), and the saturated hydraulic conductivity measurements (K_s), respectively.

Soil	Texture	Sand %	Silt %	Clay %	Bulk density for UI, WRC, K_s g cm^{-3}	Particle density g cm^{-3}	Saturated hydraulic conductivity cm s^{-1}	Organic matter % in mass	Electrical conductivity μS cm^{-1}
AZ2	Sand	96.3	3.07	0.62	1.64, 1.63, 1.64	2.60	4.57×10^{-3}	0.87	663.5
AZ4B	Loamy Sand	80.4	14.2	5.45	1.52, 1.59, 1.61	2.63	8.94×10^{-3}	2.30	844.6
AZ7	Sandy Loam	58.5	32.0	9.49	1.53, 1.50, 1.50	2.57	1.51×10^{-4}	1.06	829.8
AZ11	Loam	38.5	40.1	21.3	1.53, 1.51, 1.51	2.70	1.91×10^{-4}	1.74	495.2
AZ13	Sandy Clay Loam	58.1	15.4	26.5	1.49, 1.49, 1.49	2.55	5.50×10^{-4}	2.75	835.6
AZ15	Silt Loam	3.62	73.4	23.0	1.27, 1.40, 1.40	2.46	2.37×10^{-5}	3.44	3016.7
AZ18	Clay	29.2	18.7	52.2	1.40, 1.40, 1.43	2.61	3.22×10^{-4}	3.98	760.8

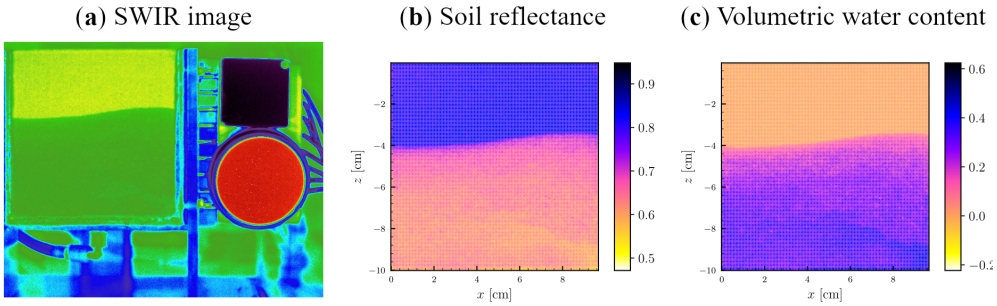

Figure 3.2: A shortwave infrared (SWIR) image taken for AZ7 soil at the time $t = 100$ min. (**a**): SWIR image; (**b**): Soil reflectance [-]; (**c**): Volumetric water content [-].

with a 3 mm thickness was placed at the bottom of the experimental box to enhance uniform water entry. The top of the experimental box was open to the atmosphere. A time domain reflectometry (TDR) array (Sheng et al., 2017) was inserted into the experimental box during the packing, though we did not used the volumetric water content measured by the TDR array in the study. The established bulk density for the upward infiltration experiment for each soil is summarized in Table 3.1. To induce upward infiltration, they applied the zero pressure head at the bottom of the soil column by connecting a Mariotte bottle to the experimental box. The soil column was weighed during the experiments by a laboratory scale with a resolution of 0.01 g to obtain cumulative infiltration data, which were used for the inverse modeling.

To obtain soil moisture data on high spatial and temporal resolutions, they used a SWIR imaging camera (SWIR 640 P-Series camera; Infrared Cameras Inc., USA) to monitor the experiments. The soil reflectance for a wavelength between 900 – 1700 nm was measured by the camera with two standard panels with known reflectance: a panel with Magic Black coating (ACKTAR Ltd., Israel) with 2% reflectance in the SWIR domain and a white Spectralon Diffuse Reflectance Standard panel (Labsphere Inc., USA) with an average reflectance of 99% in the SWIR range. The spatial resolution was approximately 0.03 cm. An example of the SWIR image and the soil reflectance measurement taken for AZ7 soil is shown in Fig. 3.2, which demonstrates the soil reflectance captured the wetting front clearly. The SWIR imaging and the scale measurement were conducted with a predetermined interval for each soil (0.25 minutes for AZ2, AZ4B, and AZ13 soils, and 1 minute for the other soils).

The soil reflectance measured by the SWIR camera was converted into two-dimensional soil moisture data using a physics-based model proposed by Sadeghi et al. (2015). The model is based on the Kubelka and Munk (1931) theory of re-

flectance, where soil reflectance R [-] is related to the light absorption coefficient of soil k [L^{-1}] and the light scattering coefficient of soil s [L^{-1}]:

$$r := \frac{k}{s} = \frac{(1-R)^2}{2R}. \tag{3.1}$$

Here the ratio r is commonly called the "transformed reflectance." Sadeghi et al. (2015) assumed that the scattering of water in soils is negligible compared to that for an oven-dry soil, which may be true for a SWIR range, and derived a simple expression for the volumetric water content θ of a soil:

$$\theta = \theta_s \frac{r - r_d}{r_s - r_d}, \tag{3.2}$$

where θ_s is the saturated volumetric water content [-], r_s and r_d are the transformed reflectance for saturated and oven-dry soils, respectively. This linear model can be calibrated by measuring r, r_d, and θ. Sadeghi et al. (2018) proved that the point-scale model (Eq. 3.2) can be extended to a soil column scale provided that r_s, r_d, and θ_s are uniform in the soil column. This proof underpins the validity of calibrating the model by replacing θ by the averaged volumetric water content of the soil column $\bar{\theta}$, which can be measured by weighing the soil column, and r by the averaged transformed reflectance $\bar{r}$. We took the same approach, and the result of the calibration for each soil is shown in Fig. 3.3, which demonstrates that the linear relationship was held well. However, we observed some deviations from the linear relationship for near-saturated conditions particularly for clay rich soils (e.g., AZ13 and AZ18 soils). This is due to the neglect of the Fresnel reflectance, which may not be negligible when soils are near-saturation (Sadeghi et al., 2015). Using the calibration lines, we converted the soil reflectance into the two-dimensional volumetric water content data for each SWIR image (see Fig. 3.2 (**c**)).

The volumetric water content θ at the beginning of each upward experiment was assumed to be zero. This may not be true because the soils must have absorbed water from the laboratory air. Assuming a 293 K laboratory room temperature and a 40% relative humidity, the water potential corresponding to the vapor water is about -10^6 cm according to the Kelvin equation. Based on the WRC measurements (see Fig. 3.1), θ would be approximately 0.05 for dry AZ18 soil for the water potential. We did not correct this error because the exact water potential of the laboratory was unknown, and the correction would not change the study's conclusion. Thus, the measured volumetric water content at the beginning of the experiments was assumed to be zero for all the soils.

The pixel-wise volumetric water content data were noisy, and we computed the average of the volumetric water content at each depth. We computed the average in two ways: average over a point and a horizontal line. First, we define the coordinate system (x, z), where x is the horizontal coordinate with $x = 0$ being at the left of the experimental box and z is the vertical coordinate with $z = -10$ cm set to the bottom of the experimental box (see Fig. 3.2 (**c**)). As for the point average, we define a rectangular box (x, z) such that $x \in [x_0 - \delta_x, x_0 + \delta_x]$ and $z \in [z_0 - \delta_z, z_0 + \delta_z]$ for a

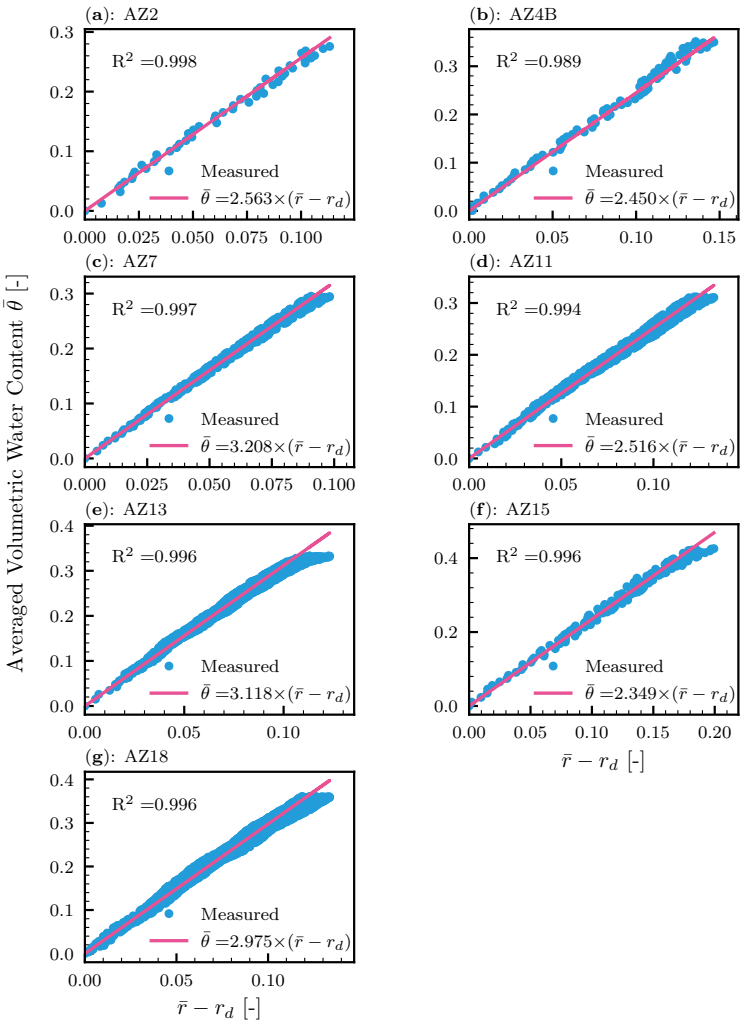

Figure 3.3: The calibration of the linear model (Eq. 3.2) for each soil, where the point-wise volumetric water content θ is replaced by the averaged volumetric water content $\bar{\theta}$, and the transformed reflectance r is replaced by the averaged transformed reflectance in the soil column $\bar{r}$.

point $(x, z) = (x_0, z_0)$ and computed the average of the volumetric water content at pixels contained in the rectangular box. We set $\delta_x = \delta_z$ and $x_0 = 5$ cm. As for the horizontal average, we also define a rectangular box (x, z) such that $z \in [z_0 - \delta_z, z_0 + \delta_z]$ for a depth $z = z_0$ and all $x \in [0, 10]$ and computed the average of the volumetric water content at pixels contained in the rectangular box. For both cases, we set the threshold value $\delta_z = 0.25$ cm, which corresponds to 8 pixels. Thus, the point average is the average of about 64 pixels, and the horizontal average is the average over 2664 pixels. Fig. 3.4 shows the point and the horizontal average at $z = -4.5$ cm for AZ7 soil. The breakthrough curve for the point averaged θ was sharper than that for the

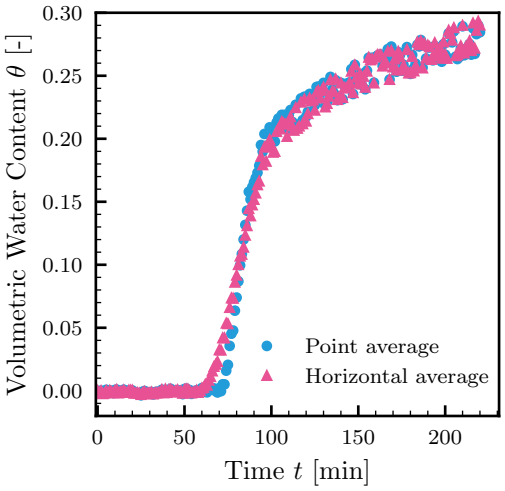

Figure 3.4: The point and the horizontal average of the volumetric water content for AZ7 soil at a depth $z = -4.5$ cm.

horizontal averaged θ. In the inverse modeling, we used both data to estimate soil hydraulic parameters.

3.2.2 Forward modeling

3.2.2.1 Richardson-Richards equation

One-dimensional water transport in isotropic, isothermal, and rigid soils can be described by the Richardson-Richards equation (Richardson, 1922; Richards, 1931):

$$\frac{\partial \theta(\psi)}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right] \quad \text{for} \quad z \in (-Z, 0) \quad \text{and} \quad t \in (0, T), \tag{3.3}$$

where θ is the volumetric water content [L^3 L^{-3}], z is the vertical coordinate (positive upward) [L], t is the time [T], ψ is the water potential [L], K is the hydraulic conductivity [L T^{-1}], Z is the length of the soil column [L], T is the final time [T]. θ and K are both functions of ψ, and several parametric models have been proposed (Assouline and Or, 2013), which will be described in Sect. 3.2.2.2. We consider the following initial and boundary conditions:

$$\psi(z, 0) = \psi_i(z), \quad z \in [-Z, 0], \tag{3.4}$$

$$\psi(-Z, t) = \psi_{lb}, \quad t \in (0, T), \tag{3.5}$$

$$q(z, t) := -K(z, t)\left(\frac{\partial \psi(z, t)}{\partial z} + 1\right) = 0, \quad z = 0, \quad t \in (0, T), \tag{3.6}$$

where ψ_i is the initial condition, ψ_{lb} is the water potential at the lower boundary, $q(z,t)$ is the water flux in the vertical direction. ψ_i was set to $-10^{6.8}$ cm because the soil was (assumingly) oven-dry at the beginning of each upward infiltration experiment (Schneider and Goss, 2012), and the lower boundary condition was $\psi_{lb} = -0.01$ cm. Because we only used the experimental data before water reached the top of the experimental box, the upper water flux boundary condition was set to 0 for all t.

3.2.2.2 Soil hydraulic functions

The van Genuchten-Mualem (VGM) model (Mualem, 1976; van Genuchten, 1980) and Peter-Durner-Iden (PDI) model (Peters, 2013; Iden and Durner, 2014; Peters, 2014) were used to represent the soil hydraulic functions $\theta(\psi)$ and $K(\psi)$. In the VGM model, $\theta(\psi)$ is written as (van Genuchten, 1980):

$$\theta(\psi) = \theta_r + (\theta_s - \theta_r)\Gamma(\psi), \tag{3.7}$$

where θ_r is the residual volumetric water content [-], θ_s is the saturated volumetric water content [-], Γ is referred to as the VG saturation function here and defined as:

$$\Gamma(\psi) := (1 + (-\alpha\psi)^n)^{-m}, \tag{3.8}$$

where α [L^{-1}] and n [-] are van Genuchten fitting parameters, m is defined as $m = 1 - 1/n$. $K(\psi)$ is written as:

$$K(\psi) = K_s\Gamma(\psi)^\tau(1 - (1 - \Gamma(\psi)^{1/m})^m)^2, \tag{3.9}$$

where K_s is the saturated hydraulic conductivity [$L\ T^{-1}$], τ is the tortuosity parameter [-].

Although the VGM model has been widely used, the applicability of the model to dry soils has been questioned. Many studies criticized the non-physical parameter θ_r (Nimmo, 1991). Also, the VGM model ignores film and corner flow in pore space, which underestimates hydraulic conductivity for dry conditions (Tuller and Or, 2001). Several models have been proposed to describe WRCs for the full range of water content (Peters and Durner, 2008; Lebeau and Konrad, 2010; Zhang, 2011). For the inverse modeling, we selected the Peter-Durner-Iden (PDI) model to describe WRCs and HCFs for conditions from oven-dry to full saturation because of its simplicity. In the PDI model below, we did not consider vapor flow. Also, note that the PDI model shares the parameters with the VGM model. Although we did not differentiate them by notations, we ensure that there is no confusion.

The WRC in the PDI model is described by the sum of capillary and non-capillary (i.e., film and corner) water:

$$\theta(\psi) = (\theta_s - \theta_r)S_c(\psi) + \theta_r S_{nc}(\psi), \tag{3.10}$$

where S_c is the capillary saturation function, and S_{nc} is the non-capillary saturation function. Here, the residual water content θ_r is interpreted as the maximum volumetric water content for non-capillary water. To ensure the capillary water becomes zero

at the water potential of oven-dryness ψ_0, the PDI model re-scaled the VG saturation function to obtain the capillary saturation function:

$$S_c(\psi) = \frac{\Gamma(\psi) - \Gamma_0}{1 - \Gamma_0}, \tag{3.11}$$

where $\Gamma_0 = \Gamma(\psi_0)$. We used $\psi_0 = 10^{6.8}$ cm (Schneider and Goss, 2012), which corresponds to the water potential for oven-dry soils. Note that the effect of the rescaling compared to the original VG model is negligible for sandy soil and silt loam but eminent for clay loam soils for dry conditions (Iden and Durner, 2014). The saturation function for the non-capillary water is described by the following empirical equation:

$$S_{nc} = 1 + \frac{1}{\psi'_\alpha - \psi'_0}\left(\psi' - \psi'_\alpha + b\ln\left(1 + \exp\left(\frac{\psi'_\alpha - \psi'}{b}\right)\right)\right), \tag{3.12}$$

where $\psi' = \log_{10}(-\psi)$, $\psi'_\alpha = \log_{10}(\alpha^{-1})$, $\psi'_0 = \log_{10}(-\psi_0)$, and b is a smoothing parameter empirically calculated as

$$b = 0.1 + \frac{0.2}{n^2}\left(1 - \exp\left(-\left(\frac{\theta_r}{\theta_s - \theta_r}\right)^2\right)\right). \tag{3.13}$$

The hydraulic conductivity for the PDI model is the weighted sum of the capillary and non-capillary parts:

$$K(\psi) = K_c(\psi) + K_{nc}(\psi), \tag{3.14}$$

where K_c and K_{nc} are hydraulic conductivity for capillary water and non-capillary water, respectively. The hydraulic conductivity of the capillary flow is given as

$$K_c(\psi) = K_{sc}S_c^\tau\left(1 - \frac{\left(1 - \Gamma(\psi)^{1/m}\right)^m}{\left(1 - \Gamma_0^{1/m}\right)^m}\right)^2, \tag{3.15}$$

where K_{sc} is the saturated hydraulic conductivity for the capillary flow [L T^{-1}]. Note that the τ must be larger than -1 to satisfy the monotonicity and concavity constraints of HCFs (Peters, 2014).

The conductivity for non-capillary flow is obtained by the Tokunaga's model (Tokunaga, 2009):

$$K_{nc}(\psi) = K_{snc}\left(\frac{\psi_0}{\psi_\alpha}\right)^{a(1-S_{nc}(\psi))}, \tag{3.16}$$

where K_{snc} is the saturated hydraulic conductivity for the non-capillary flow [L T^{-1}], $\psi_\alpha = -\alpha^{-1}$, and a is the slope of $K_{nc}(\psi)$ in the log-log plot [-].

3.2.2.3 Finite element methods

The RRE with the initial and boundary conditions given soil hydraulic functions were solved by a finite element method. The solution to the RRE, $\psi(z,t)$, was approximated by the discrete solution $\psi_h(z,t)$:

$$\psi_h(z,t) := \sum_{j=1}^{N_s} \psi_j(t)\phi_j(z), \tag{3.17}$$

where N_s is the number of nodes, ϕ_j for $j = 1, ..., N_s$ are the linear basis functions, $\psi_j(t)$ is the water potential at the node j at the time t. Discrete volumetric water content θ_h and hydraulic conductivity K_h were also approximated by the linear basis functions, and the nodal volumetric water content $\theta_j(t)$ and hydraulic conductivity $K_j(t)$ were computed from the water potential at the same node $\psi_j(t)$ using the WRC and the HCF. Using the standard Galerkin method, we obtain the discrete variational problem, where we seek ψ_h such that

$$\int_{-Z}^{0} \frac{\partial \theta_h}{\partial t}\phi_i \, dz + \int_{-Z}^{0} K_h \left(\frac{\partial \psi_h}{\partial z} + 1 \right) \frac{\partial \phi_i}{\partial z} \, dz = 0 \tag{3.18}$$

for $i = 1, ..., N_s$. The temporal derivative was discretized by the backward Euler method by introducing the times t^n for $n = 0, ..., N_t$, such that

$$0 = t^0 < t^1 \cdots < t^n < \cdots < t^{N_t} = T, \tag{3.19}$$

and the corresponding time steps $\Delta t^n = t^n - t^{n-1}$ for $n = 1, ..., N_t$, where the superscript n represents the time $t = t^n$. For the time $t = t^n$ with $n = 1, ..., N_t$, we obtain the discrete variational problem, where we seek ψ_h^n such that

$$\int_{-Z}^{0} \frac{\theta_h^n - \theta_h^{n-1}}{\Delta t^n}\phi_i \, dz + \int_{-Z}^{0} K_h^n \left(\frac{\partial \psi_h^n}{\partial z} + 1 \right) \frac{\partial \phi_i}{\partial z} \, dz = 0 \tag{3.20}$$

for $i = 1, ..., N_s$. Mass-lumping was used for the temporal derivative term (the first integral in Eq. 3.20) to obtain a smooth and non-oscillatory solution (Neuman, 1973; Celia et al., 1990). The resulting non-linear variational problem needs to be solved with iterative methods. Although the modified Picard method proposed by Celia et al. (1990) is commonly used to solve the non-linear system, we used the Newton method with Armijo line search (Kelley, 2018). The non-linear variational problem above can be written as the systme of non-linear equations (readers may find Scudeler et al. (2016) useful):

$$\mathbf{F}(\boldsymbol{\psi}^{n,k}) = 0, \tag{3.21}$$

where $\mathbf{F}$ denote the system of the non-linear equations, $\boldsymbol{\psi}^{n,k} := [\psi_1^{n,k}, \psi_2^{n,k}, ..., \psi_{N_s}^{n,k}]^T$ is the solution at the kth Newton iteration with $\boldsymbol{\psi}^{n,0} = \boldsymbol{\psi}^{n-1}$. For each Newton iteration, the Newton direction $\mathbf{d}$ was determined by solving the Newton system:

$$\mathbf{F}'(\boldsymbol{\psi}^{n,k})\mathbf{d}^k = -\mathbf{F}(\boldsymbol{\psi}^{n,k}), \tag{3.22}$$

where the Jacobian matrix $\mathbf{F}'$ was computed with analytical $\frac{d\theta}{d\psi}$ and $\frac{dK}{d\psi}$, and the linear system was solved by Gaussian elimination. The Newton step size is determined by Armijo line search:

$$\boldsymbol{\psi}^{n,k+1} = \boldsymbol{\psi}^{n,k} + \lambda \mathbf{d}^k, \tag{3.23}$$

with the initial $\lambda = 1$, and λ was reduced by a factor 0.5 when the sufficient decrease condition

$$||\mathbf{F}(\boldsymbol{\psi}^{n,k+1})|| < (1 - c\lambda)||\mathbf{F}(\boldsymbol{\psi}^{n,k})||, \tag{3.24}$$

where $c = 10^{-4}$, was not met. The Newton iteration was terminated when

$$||\mathbf{F}(\boldsymbol{\psi}^{n,k})|| \leq \tau_a + \tau_r||\mathbf{F}(\boldsymbol{\psi}^{n,0})||, \tag{3.25}$$

where $\tau_a = 10^{-7}$ and $\tau_r = 10^{-7}$.

While the modified Picard method only needs to evaluate the derivative of the volumetric water content $\frac{\partial\theta}{\partial\psi}$, the Newton method requries the derivative of the hydraulic conductivity $\frac{\partial K}{\partial\psi}$ in addition to $\frac{\partial\theta}{\partial\psi}$. As a result, the computational efficiency we can expect for the Newton method is often diminished by the additional evaluation of $\frac{\partial K}{\partial\psi}$, as reported by Lehmann and Ackerer (1998). However, we experienced that the Newton method is more efficient than the modified Picard method for simulating the upward infiltration when using the PDI model, while the two methods were comparable for the VGM model. Because we did not investigate the efficiency of the Newton method for other situations, we do not discuss which method is more efficient for the PDI model.

3.2.2.4 Cumulative flux calculation

Because we used the cumulative infiltration data for the inverse modeling, we need to compute the water flux at the lower boundary. Although we could compute the flux at the lower boundary by discretizing the Buckinghan Darcy law (Eq. 3.6), the computed flux may not be accurate when using a linear finite element method. Therefore, we computed the lower boundary flux based on the mass balance with the known upper boundary flux (Vogel et al., 1996). By integrating the continuity equation, we obtain

$$q_{lb} = q_{ub} + \int_{-Z}^{0} \frac{\partial\theta}{\partial t}\, dz, \tag{3.26}$$

where q_{lb} and q_{ub} are water fluxes at the lower and upper boundaries, respectively. Because q_{ub} was set to zero, the cumulative water flux at the lower boundary $I(t)$ (or cumulative infiltration) can be calculated as

$$I(t) = \int_0^t q_{lb}\, dt = \int_0^t \int_{-Z}^{0} \theta(z,t)\, dz\, dt. \tag{3.27}$$

Here, the integral was evaluated by the trapezoidal rule.

3.2.3 Inverse modeling

The numerical solution to the RRE with the initial and boundary conditions (Eq. 3.3) was fitted to the experimental data by varying parameters in the VGM and PDI models. As for the VGM model, there are six parameters, θ_r, θ_s, α, n, K_s, and τ, while there is one additional parameter K_{snc} in the PID model. Ideally, all the parameters should be estimated simultaneously for each model. However, the experimental data contained limited information on those parameters, which makes the inverse problem ill-posed (i.e., non-unique solution and stability against noise). We dealt with the ill-posedness by three methods.

First, instead of using a local optimization method, we used a global optimization method called the shuffled complex evolution developed by the University of Arizona (SCE-UA) (Duan and Gupta, 1992). This method combines the Nelder-Mead method (Nelder and Mead, 1965) with evolutionary algorithms. This method allows us to define the range of the target parameters, which helps to deal with the ill-posedness, too. Readers should refer to Duan and Gupta (1992) and Duan et al. (1994) for the detail.

Second, we limited the number of parameters to be estimated, and fixed values were used for θ_r and τ for both models. As for θ_r, $\theta_r = 0$ was used for the VGM model. The θ_r for the PDI model was based on the θ_r value estimated by fitting the PDI model to the independently measured WRC (Fig. 3.1) with a correction due to bulk density differences. Fitting the VGM and the PDI models to the experimental WRC was conducted by using the SCE-UA algorithm. Fig. 3.5 shows the fitted models, and Table 3.2 shows the estimated parameters. $\tau = 0.5$ was used for both models. We should note that the tortuosity parameter τ for the VGM model should be treated as a fitting parameter (Schaap and Leij, 2000). This parameter greatly changes the shape of the HCF and can increase the hydraulic conductivity K for dry conditions if the value is small. Although this could improve the fitting of the VGM model to the experimental data, we decided not to allow τ to vary because of the high correlation with K_s and the unphysical negative value of the parameter. Therefore, we fixed the parameter $\tau = 0.5$ and focused on the comparison between the VGM and the PDI models.

Third, we added a regularization term regarding the parameter n in an objective function to deal with the ill-posedness. This was because of the high correlation between n and K_s parameters for both models (K_{sc} for the PDI model). We used a n value estimated from the independently measured WRC as prior information. Therefore, we defined the objective function $\mathcal{J}$ as

$$\mathcal{J}(\Theta) = \sum_{i=1}^{N^\theta} \left(\hat{\theta}^i(\Theta) - \theta^i_{\text{obs}} \right)^2 + \sum_{i=1}^{N^I} \left(\hat{I}^i(\Theta) - I^i_{\text{obs}} \right)^2 + \gamma \left(\log_{10} n - \log_{10} n' \right)^2, \quad (3.28)$$

where N^θ and N^I are the numbers of observations for the volumetric water content θ_{obs} and cumulative infiltration I_{obs}, Θ is a set of parameters to be estimated (i.e., $\Theta = \{\theta_s, \alpha, n, K_s\}$ for the VGM model and $\Theta = \{\theta_s, \alpha, n, K_{sc}.K_{snc}\}$ for the PDI

Table 3.2: Estimated soil hydraulic parameters for the VGM and the PDI models from the independently measured water retention curve data, shown in Fig. 3.1.

Soil	Soil Hydraulic Model	θ_r -	θ_s -	α cm^{-1}	n -
AZ2	VGM	0.027	0.364	0.0176	3.629
AZ2	PDI	0.049	0.363	0.0181	4.152
AZ4B	VGM	0.035	0.388	0.0184	1.820
AZ4B	PDI	0.106	0.379	0.0155	2.696
AZ7	VGM	0.017	0.391	0.0117	1.416
AZ7	PDI	0.101	0.387	0.0107	1.602
AZ11	VGM	0.023	0.411	0.0061	1.351
AZ11	PDI	0.194	0.403	0.0041	2.315
AZ13	VGM	0.000	0.422	0.0296	1.215
AZ13	PDI	0.242	0.419	0.0297	1.510
AZ15	VGM	0.000	0.444	0.0027	1.276
AZ15	PDI	0.242	0.440	0.0027	1.744
AZ18	VGM	0.000	0.452	0.0084	1.196
AZ18	PDI	0.188	0.460	0.0454	1.005

model), $\hat{\theta}$ and $\hat{I}$ are the calculated volumetric water content and the cumulative flux at the lower boundary by solving the forward model given the parameters Θ, γ is the regularization parameter set to one, n' is the parameter n estimated from the independently measured WRCs (i.e., values in Table 3.2). We used the volumetric water content measurement at 8 depths ($z = -8.5, -7.5, -6.5, -5.5, -4.5, -3.5, -2.5, -1.5$ cm) and the cumulative infiltration I at all the times where the SWIR iamges were taken until the wetting front reached $z = -0.5$ cm (i.e., 0.5 cm below the soil surface).

The objective function can be interpreted as a least-square estimation with a regularization term. Here, we assumed that the variance of the discrepancy between the model and the experimental data was the same for the volumetric water content and the cumulative infiltration. In fact, it turned out that they were similar when the fitting was appropriate regardless of the difference in the range of the values. This was because the cumulative infiltration data were less noisy than the volumetric water content data. The regularization term can be interpreted as prior information. We assumed the parameter n is similar to the n estimated from the independently measured WRC data. Because this may not be accurate, we need to be careful when interpreting the estimated parameters; the estimated parameters here are dependent on the experimental data and all the assumptions we made.

3.2.4 Implementations

The inverse modeling requires a robust implementation of the forward modeling and the optimization algorithm. Particularly, the forward modeling must be robust to enable extensive parameter searches. The finite element method described above was implemented using FEniCS library (Logg et al., 2012), which consists of a C++ implementation with a Python interface. Soil hydraulic functions (i.e., VGM and PDI model) were implemented in C++ to speed up the computation, and the rest

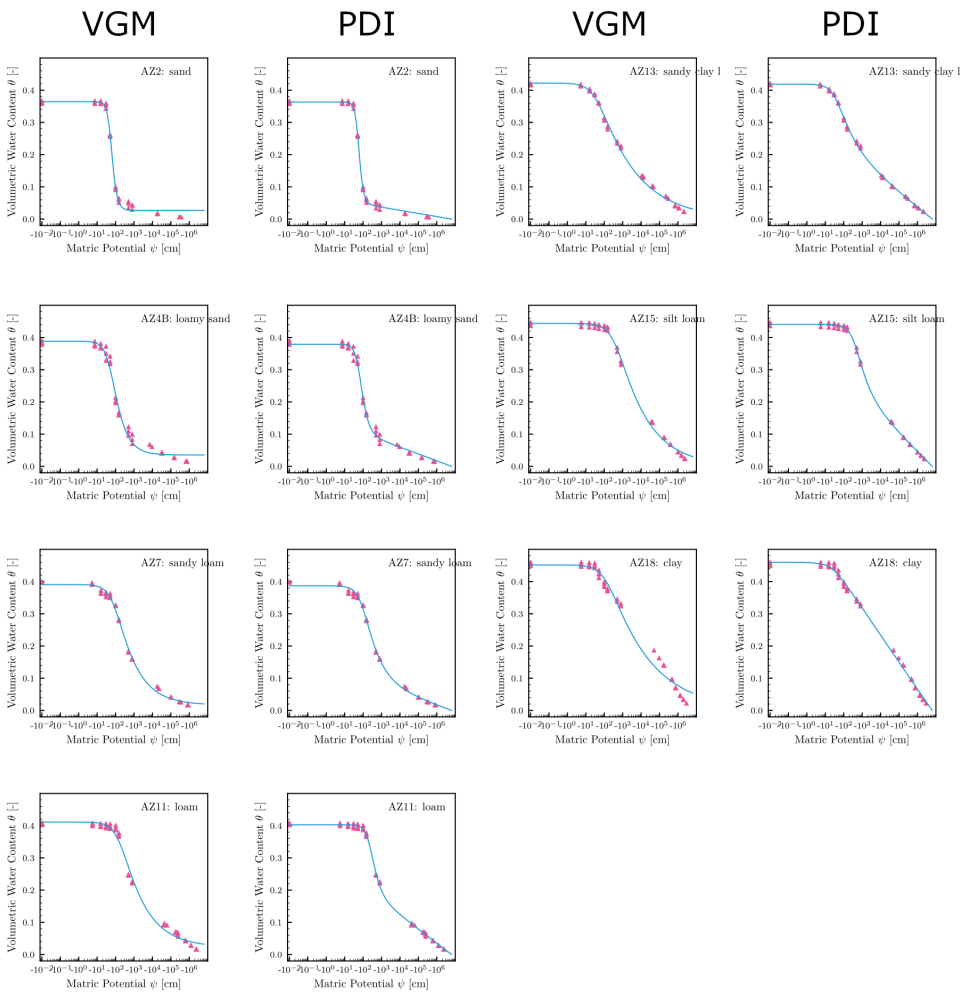

Figure 3.5: The van Genuchten-Mualem (VGM) model and the Peter-Durner-Iden (PDI) model fitted to the independently measured water retention curves for all soils.

was written in Python. An adaptive time-stepping method based on the number of the Newton iterations was used to speed up the forward modeling (Vogel et al., 1996). Solving the forward problem is not straightforward because the experimental data require us to simulate soil moisture dynamics from oven-dry to near-saturation conditions. As for very dry conditions, we enforced θ and K to be zero for $\psi < \psi_0$ for the PDI model, but no such corrections were made for the VGM model. The Newton method with the Armijo line search enabled us to use a larger time step than the modified Picard method without convergence issues. Contrary to our expectations, we were faced with convergence issues for near-saturation conditions. This occurred during the inverse modeling when a parameter n was approximately larger than six or the saturated hydraulic conductivity K_s was high, which induced fast water flow

and saturated the soil very quickly. In such a situation, both the modified Picard and Newton method experienced convergence issues due to the zero slope of WRCs. This could be alleviated by introducing an artificial slope into WRCs. Because it was not desirable to introduce such an artificial error, we stopped the forward modeling when we detected the increase in the volumetric water content at the top of the experimental box. Note that θ, $\frac{\partial\theta}{\partial\psi}$, K, and $\frac{\partial K}{\partial\psi}$ were set to θ_s, 0, the maximum conductivity (K_s for the VGM model and $K_{sc}+K_{snc}$ for the PDI model), and 0 for $\psi \geq 0$ for both models. Also, a large n value made the computation of $\frac{\partial K}{\partial\psi}$ ill-defined when ψ was very close to zero because the VG saturation function Γ became one even when $\psi < 0$ for both models due to the limited number of digits. Thus, we forced $\Gamma < 1 - 10^{-12}$. Finally, we stopped the forward modeling when the Newton iteration or the line search failed within the predetermined maximum iterations (100 for both). This occurred when the soil became saturated, and it is reasonable to ignore such cases.

Inverse modeling was implemented by combining the forward modeling explained above with the SCE-UA algorithm implemented in Python. Parameters α, n, K_s (K_{sc} for the PDI model), and K_{snc} were transformed as follows:

$$\alpha^t = \log_{10}(\tfrac{1}{\alpha}), \tag{3.29}$$

$$n^t = \log_{10}(n), \tag{3.30}$$

$$K_s^t = \log_{10}(K_s), \tag{3.31}$$

$$K_{snc}^t = \log_{10}(K_{snc}), \tag{3.32}$$

where α^t, n^t, K_s^t, and K_{snc}^t are the transformed parameters. The range of the parameters is summarized in Table 3.3. The range of θ_s was determined based on the measured volumetric water content data. The other ranges were determined by trial and error (i.e., diagnosing the convergence of each parameter in the SCE-UA algorithm).

There exist tuning parameters in the SCE-UA algorithm, and we used the default setting recommended in Duan et al. (1994) with the number of complexes of 4. We allowed the maximum evaluations of the forward modeling to be 5000 until convergence criteria were met. The convergence criteria are twofold. The first one is the geometric mean of the relative range of the parameters to be less than 0.0001. The second one is that the objective function does not improve by 0.1% in the last ten evolution loops. The same tuning parameters were used for the WRC fitting.

3.2.5 Evaluations

The results of the inverse modeling were evaluated by computing the relative error ϵ for the volumetric water content and the cumulative water flux defined as:

$$\epsilon^\delta := \left(\frac{\frac{1}{N^\delta}\sum_i^{N^\delta}(\hat{\delta}^i - \delta_{\text{obs}}^i)^2}{\frac{1}{N^\delta}\sum_i^{N^\delta}(\delta_{\text{obs}}^i)^2} \right)^{1/2} \tag{3.33}$$

for $\delta = \theta, I$, where δ_{obs} is the measurement data, $\hat{\delta}$ is the predicted value by the forward modeling, N^δ is the number of the measurement data.

Table 3.3: The range of soil hydraulic parameters for the VGM and the PDI models searched by the SCE-UA algorithm. K_s^* is the independently measured saturated conductivity shown in Table 3.1.

Parameter	Units	Soil	Min	Max
θ_s	-	AZ2	0.24	0.34
θ_s	-	AZ4B	0.31	0.41
θ_s	-	AZ7	0.32	0.42
θ_s	-	AZ11	0.35	0.45
θ_s	-	AZ13	0.33	0.43
θ_s	-	AZ15	0.40	0.65
θ_s	-	AZ18	0.35	0.45
α	cm^{-1}	-	0.00001	0.1
n	-	-	1.0001	7.5
K_s or K_{sc}	cm s^{-1}	-	$K_s^* \times 10^{-4}$	$K_s^* \times 10^2$
K_{snc}	cm s^{-1}	-	$K_s^* \times 10^{-8}$	$K_s^* \times 10$

3.3 Results and discussions

Fig. 3.6 to 3.9 show the fitted numerical solutions with the VGM and the PDI models to the horizontal and point soil moisture and the cumulative water flux for all the soils, and Table 3.4 shows the results of the inverse modeling. Fig. 3.10 shows the estimated the wetting branch of the WRCs and HCFs from the inverse modeling with the drying branch of those estimated from the independently measured WRCs and saturated hydraulic conductivity. We present and discuss the results of the inverse modeling by dividing the seven soils into three groups: AZ2 and AZ4B soils, AZ7 and AZ11 soils, AZ13, AZ15, and AZ18 soils.

3.3.1 AZ2 and AZ4B soils: Validity of one-dimensional flow

Fig. 3.6 (**a**) and (**b**) show that the numerical solutions with the VGM model could not describe the cumulative water flux for AZ2 and AZ4B soils, respectively. In these two coarse-textured soils, we observed the violation of the one-dimensional flow assumption, as shown in Fig. 3.11 (**a**) and (**b**), respectively. The fitting of the numerical solutions with the VGM and the PDI models to the soil moisture data (both horizontal and point average data) was also not good for the two soils. We, therefore, concluded that the experimental data for AZ2 and AZ4B soils were not compatible with the one-dimensional RRE regardless of the soil hydraulic functions. The sand percentage for the two soils was over 80%, and the saturated hydraulic conductivities were both high (see Table 3.1), which induced non-one-dimensional flow near the bottom of the experimental column at the beginning of the experiments. Future studies should pay special attention to the design of the inlet of the experimental box to prevent two or three-dimensional flows for coarse-textured soils. We also note that Šimůnek et al. (2001) attributed the failure of the one-dimensional RRE to describe cumulative infiltration data to a non-equilibrium flow between inter- and intra-aggregates. We did not observe such phenomena even though some aggregates were present. For

Table 3.4: Estimated soil hydraulic parameters for the VGM and the PDI models obtained by the inverse modeling with the upward infiltration experimental data. ϵ^θ and ϵ^I are the L_2 relative error defined by Eq 3.33. The column "reliability" shows the qualitative evaluation of the fittings by three ranks (reliable, middling, and failed).

Soil	Soil Hydraulic Model	Data type	ϵ^θ -	ϵ^I -	Reliability	θ_s -	α cm^{-1}	n -	K_s or K_{sc} $cm\ s^{-1}$	K_{snc} $cm\ s^{-1}$
AZ2	VGM	point	2.10×10^{-1}	1.32×10^{-1}	failed	0.272	3.99×10^{-5}	3.437	7.61×10^{-7}	
AZ2	VGM	horizontal	2.51×10^{-1}	1.31×10^{-1}	failed	0.272	3.49×10^{-5}	3.250	6.78×10^{-7}	
AZ2	PDI	point	2.02×10^{-1}	1.31×10^{-1}	failed	0.268	7.35×10^{-5}	3.864	1.29×10^{-6}	1.44×10^{-8}
AZ2	PDI	horizontal	2.41×10^{-1}	1.31×10^{-1}	failed	0.268	2.78×10^{-5}	3.490	4.89×10^{-7}	1.02×10^{-8}
AZ4B	VGM	point	2.21×10^{-1}	1.62×10^{-1}	failed	0.326	2.41×10^{-5}	1.624	9.03×10^{-7}	
AZ4B	VGM	horizontal	2.66×10^{-1}	1.62×10^{-1}	failed	0.316	3.60×10^{-5}	2.002	9.01×10^{-7}	
AZ4B	PDI	point	2.28×10^{-1}	1.62×10^{-1}	failed	0.311	4.42×10^{-5}	2.226	9.14×10^{-7}	1.35×10^{-8}
AZ4B	PDI	horizontal	2.57×10^{-1}	1.62×10^{-1}	failed	0.321	8.12×10^{-5}	2.277	1.47×10^{-6}	4.14×10^{-8}
AZ7	VGM	point	9.33×10^{-2}	9.51×10^{-3}	reliable	0.333	4.57×10^{-2}	2.006	1.21×10^{-4}	
AZ7	VGM	horizontal	8.02×10^{-2}	8.32×10^{-3}	reliable	0.335	5.16×10^{-2}	2.140	1.29×10^{-4}	
AZ7	PDI	point	6.95×10^{-2}	8.17×10^{-3}	reliable	0.376	4.89×10^{-1}	1.107	1.08×10^{-4}	1.79×10^{-4}
AZ7	PDI	horizontal	4.80×10^{-2}	6.90×10^{-3}	reliable	0.375	3.85×10^{-1}	1.134	2.05×10^{-3}	1.20×10^{-4}
AZ11	VGM	point	8.82×10^{-2}	1.95×10^{-2}	reliable	0.374	6.62×10^{-2}	1.639	2.10×10^{-4}	
AZ11	VGM	horizontal	7.02×10^{-2}	1.82×10^{-2}	reliable	0.382	7.00×10^{-2}	1.646	2.20×10^{-4}	
AZ11	PDI	point	8.23×10^{-2}	1.54×10^{-2}	reliable	0.398	6.83×10^{-1}	1.069	1.88×10^{-2}	1.83×10^{-4}
AZ11	PDI	horizontal	6.78×10^{-2}	1.51×10^{-2}	reliable	0.379	1.43×10^{-1}	1.286	1.05×10^{-3}	9.11×10^{-6}
AZ13	VGM	point	1.02×10^{-1}	1.69×10^{-2}	failed	0.394	1.31×10^{-1}	1.204	4.14×10^{-3}	
AZ13	VGM	horizontal	9.18×10^{-2}	1.45×10^{-2}	middling	0.401	1.59×10^{-1}	1.204	5.51×10^{-3}	
AZ13	PDI	point	1.24×10^{-1}	1.99×10^{-2}	failed	0.358	1.32×10^{-1}	1.392	1.65×10^{-3}	6.05×10^{-6}
AZ13	PDI	horizontal	1.10×10^{-1}	1.68×10^{-2}	middling	0.368	1.79×10^{-1}	1.355	2.83×10^{-3}	1.17×10^{-5}
AZ15	VGM	point	2.14×10^{-1}	3.17×10^{-2}	failed	0.476	5.92×10^{-5}	1.486	6.06×10^{-7}	
AZ15	VGM	horizontal	1.63×10^{-1}	3.04×10^{-2}	failed	0.459	1.18×10^{-3}	1.347	1.76×10^{-5}	
AZ15	PDI	point	2.01×10^{-1}	3.40×10^{-2}	failed	0.430	7.02×10^{-3}	2.751	2.85×10^{-5}	1.03×10^{-7}
AZ15	PDI	horizontal	1.57×10^{-1}	3.10×10^{-2}	failed	0.426	3.80×10^{-5}	3.087	1.35×10^{-7}	1.35×10^{-9}
AZ18	VGM	point	1.44×10^{-1}	2.34×10^{-2}	failed	0.408	5.11×10^{-2}	5.487	2.72×10^{-5}	
AZ18	VGM	horizontal	1.12×10^{-1}	1.76×10^{-2}	failed	0.433	6.74×10^{-2}	1.399	1.78×10^{-4}	
AZ18	PDI	point	1.30×10^{-1}	2.19×10^{-2}	failed	0.410	5.11×10^{-2}	1.530	8.99×10^{-5}	1.34×10^{-7}
AZ18	PDI	horizontal	1.03×10^{-1}	1.66×10^{-2}	failed	0.422	9.21×10^{-2}	1.313	3.09×10^{-4}	1.54×10^{-6}

the other five soils, the estimated cumulative water flux was reasonably compatible with the measured data, which implied that the one-dimensional flow assumption was valid for the five soils. We also confirmed the one-dimensional flow assumption by the SWIR images, as exemplified by the one for AZ7 soil shown in Fig. 3.11 (**c**).

3.3.2 AZ7 and AZ11 soils: Advantage of PDI model

For AZ7 soil (sandy loam soil), the RRE with the VGM model well captured the horizontal averaged soil moisture data, as shown in Fig. 3.6 (**c**), except for very dry and near-saturation conditions. For very dry conditions, the numerical solution with the PDI model could fit the horizontal averaged soil moisture data better than that with the VGM model (see Fig. 3.7 (**c**)). This demonstrates that the PDI model is better than the VGM model for describing soil moisture dynamics for dry conditions. However, the numerical solution with the PDI model overestimated the horizontal averaged water content for near-saturation conditions. This was attributed to the fact that the PDI model overestimates water retention for near saturation. Vogel et al. (2001) investigated the effect of the shape of WRCs for near-saturation conditions on upward infiltration and demonstrated that introducing an air entry value to the VGM model significantly changed the timing of the arrival of wetting fronts. Nevertheless,

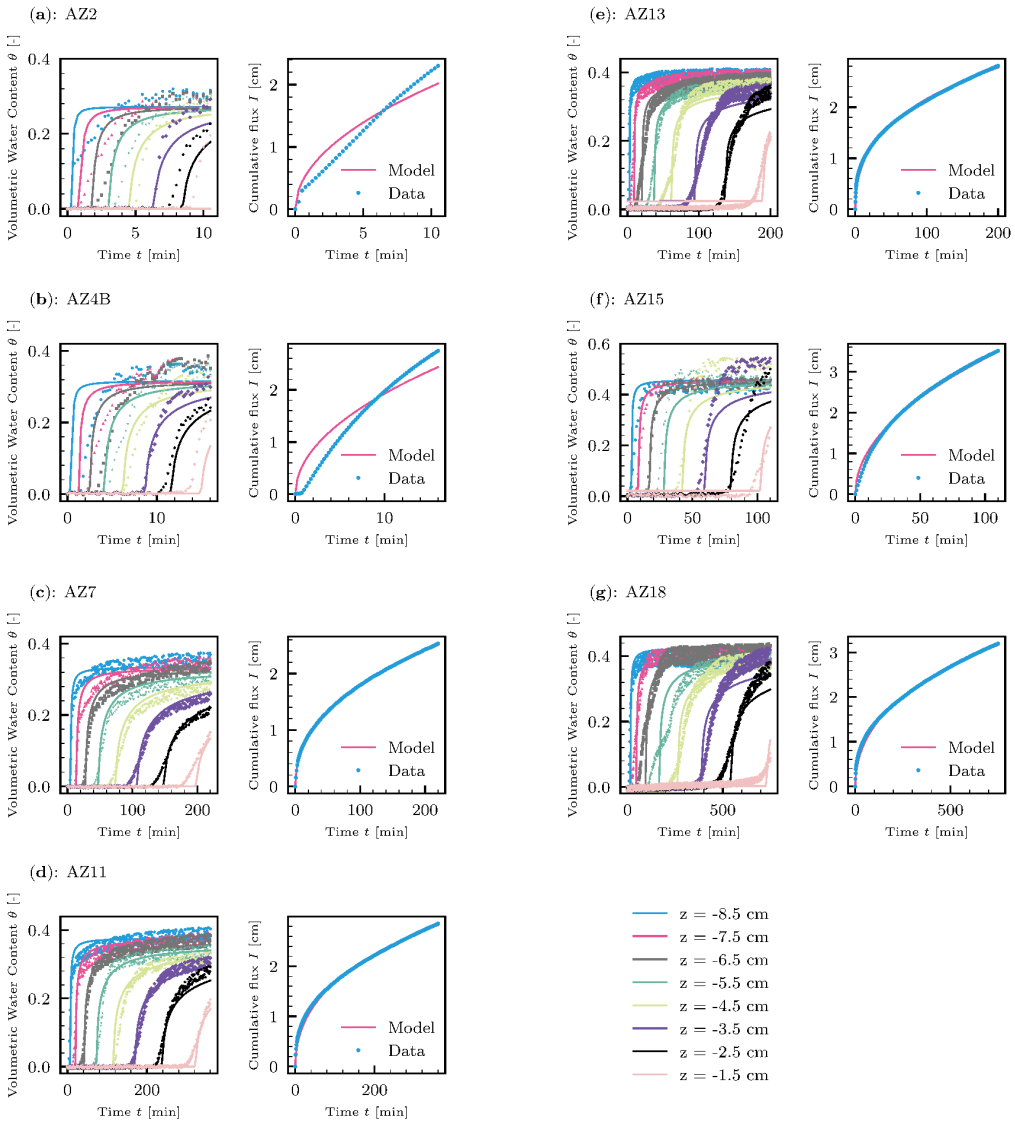

Figure 3.6: Numerical solutions with the VGM model fitted to the horizontal average volumetric water content data from the upward infiltration experiments.

the modified VGM model with an air entry value would not describe the gradual increase in the volumetric water content observed in the upward experiments here. Future studies should address this issue by, for example, introducing a finite slope near saturation.

Compared to the horizontal averaged data, the point averaged data showed sharper wetting fronts (see Fig. 3.8 (**c**)). The fitting of the numerical solutions was better for the horizontal average than the point average for the VGM and PDI models for AZ7 soil (see Table 3.4). This was because the point average data were more af-

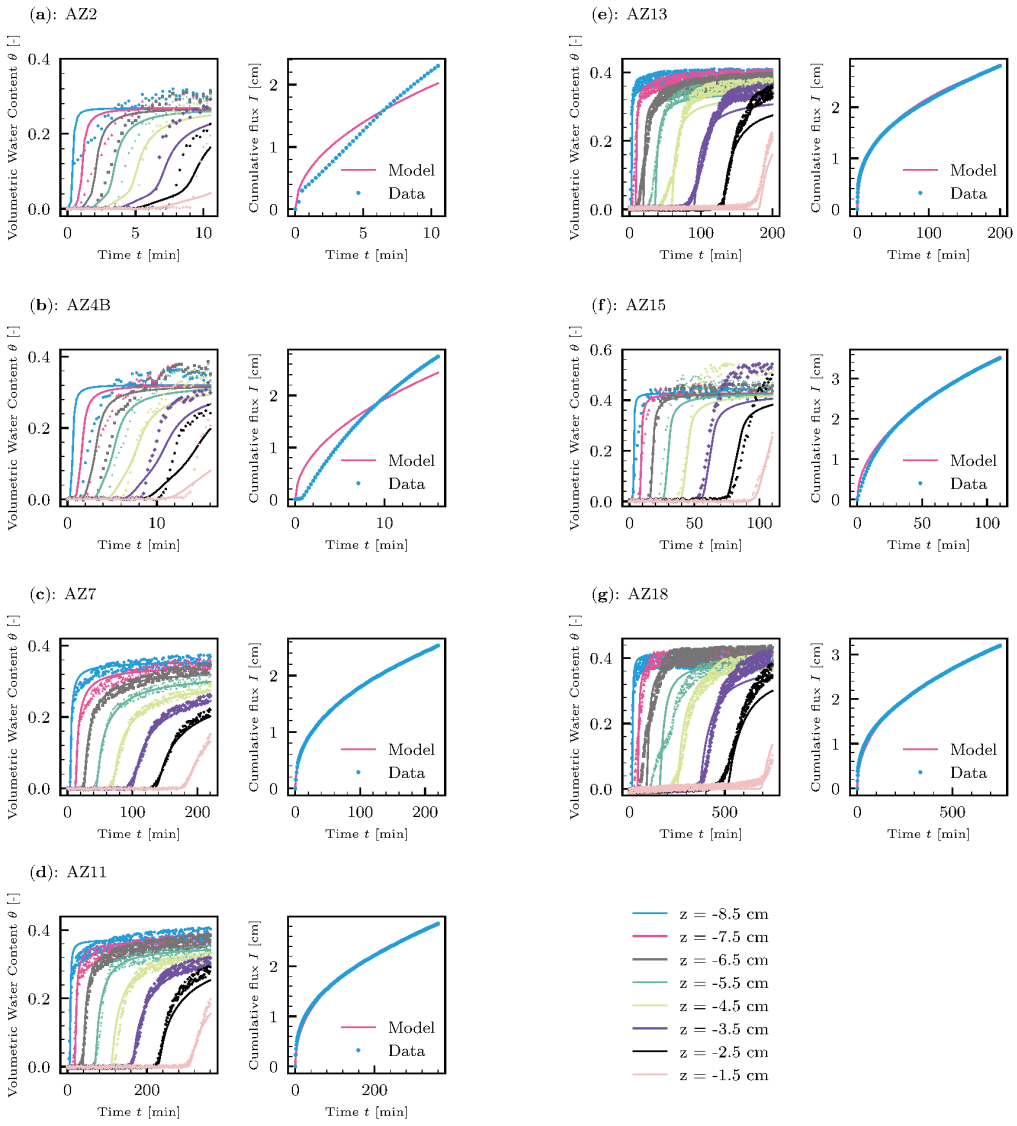

Figure 3.7: Numerical solutions with the PDI model fitted to the horizontal average volumetric water content data from the upward infiltration experiments.

fected by the heterogeneity of soils, which contradicts the RRE assuming the soil is homogeneous. In the next section, we discuss the effects of the heterogeneity of soils.

In terms of the estimated WRCs, Fig. 3.10 (**c**) demonstrate that there was no significant difference between the horizontal and point averaged soil moisture data for AZ7 soil. This was because the heterogeneity of the soil was relatively small, and the water flow was uniform. The estimated WRCs for the VGM model were quite different from those for the PDI model (see Fig. 3.10 (**c**)). The PDI model suggested more water retention for dry conditions ($\psi < -10^3$ cm) than the VGM

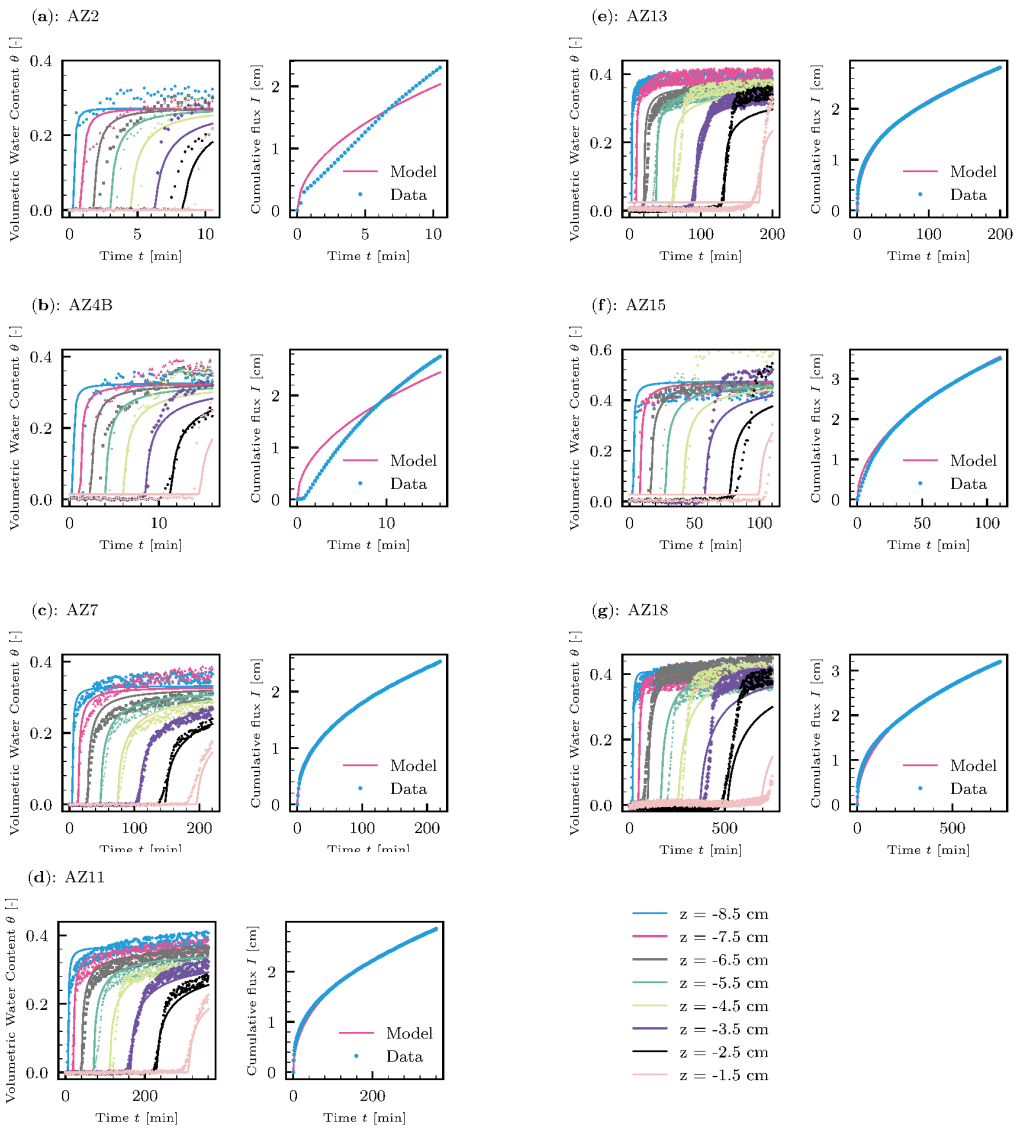

Figure 3.8: Numerical solutions with the VGM model fitted to the point average volumetric water content data from the upward infiltration experiments.

model. Unfortunately, the wetting branch of water retention data is not available, and we could not verify the validity of the estimated WRCs. However, the WRCs with the PDI model appeared to close a hysteresis loop with the independently measured WRCs. This should be expected considering the effect of hysteresis is minor for dry conditions (i.e., adsorptive water regime). This observation increased the reliability of the estimated WRCs by the PDI model than those with the VGM model.

The saturated hydraulic conductivity K_s of AZ7 soil for the VGM model was about 1.2×10^{-4} cm s^{-1}, which is in the same magnitude as the independently mea-

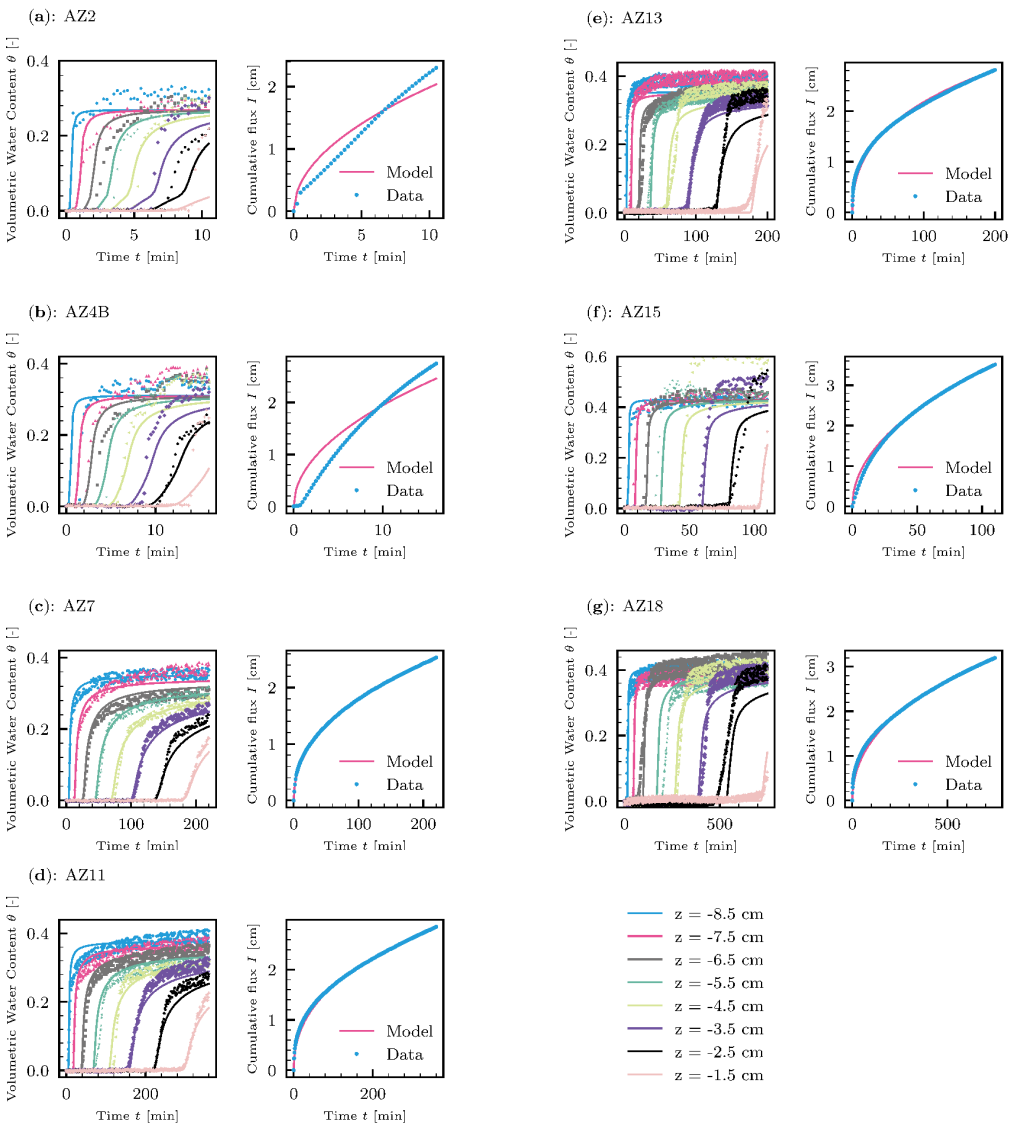

Figure 3.9: Numerical solutions with the PDI model fitted to the point average volumetric water content data from the upward infiltration experiments.

sured K_s (Table 3.1). For the PDI model, the estimated K_{sc} from the point average data was comparable to the independently measured K_s. However, K_{sc} for the PDI model estimated from the horizontal average soil moisture data was higher than independently measured K_s by one order magnitude, although the estimated HCFs for the horizontal and point average data were almost indistinguishable (see Fig. 3.10 (**c**)). We attributed this to the limited information on soil hydraulic functions near saturation in the upward infiltration experimental data. If we applied a pressure heat above zero at the bottom of the experimental box, we would gain more information

on soil hydraulic functions for near saturation (Latorre and Moret-Fernández, 2019).

The results for AZ11 soil were comparable to those for AZ7 soil (see Table 3.4). AZ7 and AZ11 soils exhibited the lowest heterogeneity among the soils, which was evident in the SWIR images. We concluded that the inverse modeling with the one-dimensional RRE using upward infiltration experimental data is most effective for sandy loam and loam soils.

3.3.3 AZ13, AZ15, and AZ18 soils: Effects of heterogeneity

For the other soils, AZ13, AZ15, and AZ18 soils, there was difficulty in fitting the numerical solutions due to the heterogeneity of the soils. For AZ13, the numerical solution with the PDI model captured the overall trend of the horizontal averaged soil moisture data, as shown in Fig. 3.7 (**e**), but not for the point average data (see Fig. 3.9 (**e**)). We attributed this to the heterogeneity of the soil due to local variances of WRCs and HCFs. Also, the heterogeneity in soil optical properties could introduce errors in the SWIR-based volumetric water content data. The linear model (Eq. 3.2) assumed that the soil optical properties r_s and r_d are constant in the soil column. However, this assumption may be violated in AZ13, AZ15, and AZ18 soils, adding uncertainty to the estimated volumetric water content data. The SWIR images demonstrated the existence of local heterogeneity of soil hydraulic properties in AZ13 soil, as exemplified in Fig. 3.11 (**d**). Also, the soil moisture data at $z = -2.5$ cm in Fig. 3.9 (**e**) became larger than that at $z = -3.5$ cm after the arrival of the wetting front, which demonstrates the heterogeneity of the porosity in the packed soil. The estimated WRC with the PDI model agreed well with the WRCs fitted to the independently measured WRCs for dry conditions. Nevertheless, we have less confidence in the estimated WRCs because the quality of the fitting was not good as AZ7 and AZ11 soils.

For AZ15 and AZ18 soils, more significant effects of the heterogeneity of soil hydraulic properties were observed. Also, the Fresnel reflectance might have contributed to the heterogeneity of the retrieved volumetric water content for these clay rich soils (Sadeghi et al., 2015). As a result, the numerical solutions with both models could not adequately describe the soil moisture data except for the timing of the arrival of the wetting fronts. The heterogeneity of soil hydraulic properties was due to the high percentage of clay materials in these two soils. Therefore, we concluded that the current framework of the inverse modeling with the upward infiltration experiment data could not reliably estimate soil hydraulic properties of soils with many clay materials due to the resulting heterogeneity.

3.4 Conclusions and perspectives

Water retention curves (WRCs) and hydraulic conductivity functions (HCFs) are necessary to simulate soil moisture dynamics based on the Richardson-Richards equation (RRE). A commonly used van Genuchten-Mualem (VGM) model is not suitable for

describing WRCs and HCFs for dry conditions. An alternative model, the Peter-Durner-Iden (PDI) model, provides WRCs and HCFs from oven-dryness to saturated conditions. We investigated whether the RRE with the PDI model can describe upward infiltration into very dry soils with distinctive textures through inverse modeling. We used upward infiltration experiment data presented in Sadeghi et al. (2017), where a shortwave infrared (SWIR) camera was used to monitor soil moisture dynamics with high resolution in space and time, along with cumulative infiltration (i.e., water flux at the lower boundary) measured by a laboratory scale.

Numerical solutions to the one-dimensional RRE were incompatible with soil moisture and cumulative water flux data for two soils with a very high sand percentage (over 80%). The SWIR images allowed us to confirm the two-(or possibly three) dimensional flow near the bottom of the experimental box, which contradicts the one-dimensional assumption of the RRE. For the other soils, the cumulative water flux was well described by the numerical solutions, and the SWIR images confirmed the one-dimensional flow.

For sandy loam and loam soils (AZ7 and AZ11 in Table 3.1), the numerical solutions with the PDI model better captured soil water dynamics for dry conditions than those with the VGM model. As a result, the wetting branch of the estimated WRCs with the PDI model showed more water-holding capacity for dry conditions than those with the VGM model. Although the wetting branch of the WRCs was not independently measured, the estimated WRCs with the PDI model were consistent with the independently measured WRCs for the drying branch in that they formed a hysteresis loop. The estimated saturated hydraulic conductivity had a reasonable agreement with the one independently measured. We concluded that the current framework of the inverse modeling with the upward infiltration experimental data is suitable for sandy loam and loam soils. Nevertheless, the fitting was not adequate for near-saturation conditions. For future studies, we should introduce a finite slope in WRCs near saturation. Also, rather than parametric models, such as the VGM and the PDI models, more flexible functions should be used to describe WRCs and HCFs. For example, previous studies used polynomial functions (Bitterlich et al., 2004) and neural networks (Bandai and Ghezzehei, 2021) to describe them. By using such flexible functions, we might be able to "discover" WRCs and HCFs that cannot be described by commonly used parametric models.

For soils with a large clay percentage, the fittings of the numerical solutions to the measured soil moisture data were not satisfactory due to the heterogeneity of the soils due to local variations in soil hydraulic properties (i.e., porosity and saturated hydraulic conductivity). The SWIR images enabled us to confirm the existence of non-uniform flow due to such heterogeneity. Dealing with such micro-heterogeneity of soils in the inverse modeling of soil moisture is challenging. For forward modeling, Miller and Miller's similar media theory (Miller and Miller, 1956; Sadeghi et al., 2016) has commonly been used to simulate soil moisture dynamics in heterogeneous soils. However, we are not aware of studies estimating parameters in such models with inverse modeling. Solving such a problem would be the next step of inverse modeling in soil moisture dynamics.

3.5 List of abbreviations

HCF: Hydraulic conductivity function
PDI: Peter-Durner-Iden
RRE: Richardson-Richards equation
SWIR: Shortwave infrared
TDR: Time domain reflectometry
UI: Upward infiltration
VG: van Genuchten
VGM: van Genuchten-Mualem
WRC: Water retention curve

3.6 List of notations

Superscript
$\cdot^t$: logarithmic transformation for the inverse modeling
$\hat{\cdot}$: compute values obtained by solving the forward problem
Subscript
$\cdot_j(t)$: finite element coefficient at node j at time t [L]
Alphabet
a: slope parameter in the PDI model [-]
b: smoothing parameter in the PDI model [-]
c: parameter for line search
$\mathbf{d}$: Newton direction
$f(t)$: water flux at the top boundary at time t [L T^{-1}]
$\mathbf{F}$: system of non-linear equations
$\mathbf{F}'$: Jacobian matrix
I: cumulative flux at the lower boundary [L]
k: light absorption coefficient [L^{-1}]
K: hydraulic conductivity [L T^{-1}]
K_c: hydraulic conductivity for capillary water [L T^{-1}]
K_{nc}: hydraulic conductivity for non-capillary water [L T^{-1}]
K_s: saturated hydraulic conductivity [L T^{-1}]
K_s^*: independently measured saturated hydraulic conductivity [L T^{-1}]
K_{sc}: saturated hydraulic conductivity for capillary water [L T^{-1}]
K_{snc}: saturated hydraulic conductivity for non-capillary water [L T^{-1}]
$\mathcal{J}$: objective function
m: van Genuchten parameter [-]
n: van Genuchten parameter [-]
n': van Genuchten parameter n estimated from the independently measured WRCs [-]
N^I: number of observations for the cumulative infiltration [-]
N_s: number of spatial nodes [-]

N_t: number of temporal steps [-]
N^θ: number of observations for the volumetric water content [-]
q: water flux [L T^{-1}]
r: transformed reflectance [-]
$\bar{r}$: averaged transformed reflectance [-]
r_d: transformed reflectance for oven-dry soils [-]
r_s: transformed reflectance for saturated soils [-]
R: soil reflectance [-]
S_c: capillary saturation function [-]
S_{nc}: non-capillary saturation function [-]
s: light scattering coefficient [L^{-1}]
t: time [T]
T: final time [T]
x: horizontal coordinate [L]:
z: vertical coordinate [L]:
Z: length of soil column [L]:
Greek alphabet
α: van Genuchten parameter [L^{-1}]
γ: regularization parameter
Γ: van Genuchten saturation function [-]
δ_z: threshold value to compute average values [L]
Δt^n: time step at the time t^n [T]
ϵ^δ: relative error for the variable δ
θ: volumetric water content [-]
$\bar{\theta}$: averaged volumetric water content of the soil column [-]
θ_r: residual volumetric water content [-]
θ_s: saturated volumetric water content [-]
Θ: set of parameters in the VGM model or the PDI model estimated in the inverse modeling
λ: step size for line search
τ: tortuosity parameter [-]
τ_a: absolute threshold for Newton iterations [-]
τ_r: relative threshold for Newton iterations [-]
ψ: water potential [L]
ψ': log transformed water potential ($\psi' = \log_{10}(-\psi)$)
ψ_0: water potential of oven-dryness [L]
ψ_i: initial condition of the water potential [L]
ψ_{lb}: water potential at the lower boundary [L]
ϕ_j: linear basis function for a node j

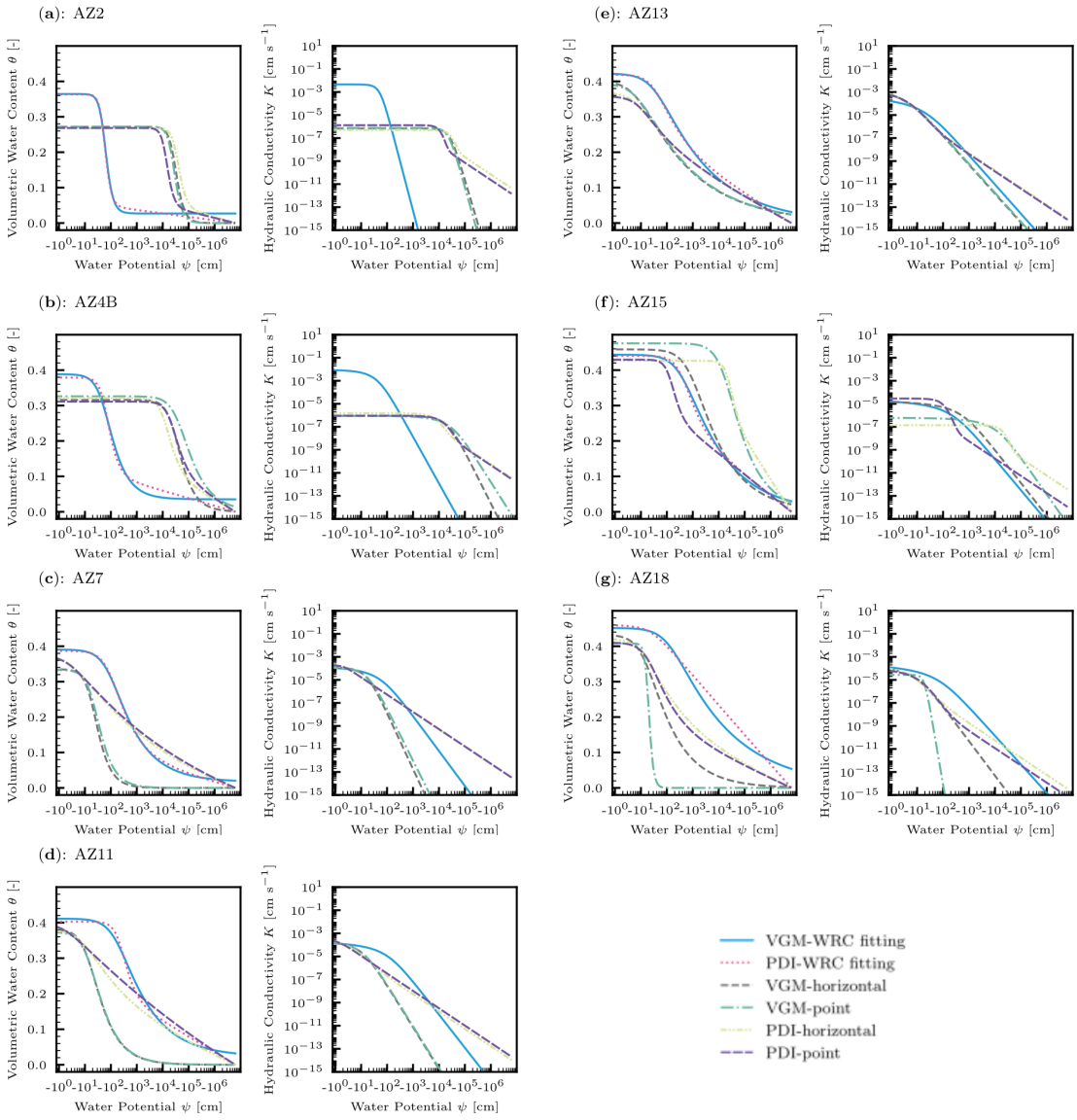

Figure 3.10: Estimated water retention curves (WRCs) and hydraulic conductivity functions (HCFs) from the inverse modeling with the horizozan and the point averaged soil moisture data. WRCs fitted to the independently measured WRCs are also shown and denoted as WRC fitting. The VGM-WRC fitting lines for HCFs were computed by the VGM model fitted to the independently measured WRCs and saturated conductivity.

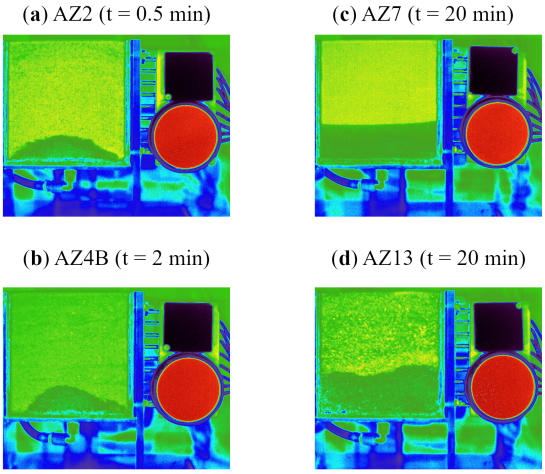

Figure 3.11: Shortwave infrared (SWIR) images. (**a**) and (**b**): The violation of one-dimensional flow for AZ2 and AZ4B soils, respectively. (**c**): Uniform one-dimensional flow observed for AZ7 soil. (**d**): Non-uniform flow caused by the heterogeneity of AZ13 soil.

References for Chapter 3

Assouline, S. and Or, D. (2013). Conceptual and parametric representation of soil hydraulic properties: A review. *Vadose Zone Journal*.

Babaeian, E., Arthur, E., Sadeghi, M., Effati, M., Gohardoust, M. R., Jones, S. B., and Tuller, M. (2021). The feasibility of shortwave infrared imaging and inverse numerical modeling for rapid estimation of soil hydraulic properties. *Vadose Zone Journal*, pages 1–16.

Bandai, T. and Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*, 57.

Bitterlich, S., Durner, W., Iden, S. C., and Knabner, P. (2004). Inverse estimation of the unsaturated soil hydraulic properties from column outflow experiments using free-form parameterizations. *Vadose Zone Journal*, 3:971–981.

Buckingham, E. (1907). *Studies on the movement of soil moisture*, volume 38 of *Bulletin*. Washington, DC.

Campbell, G. S. and Shiozawa, S. (1992). Prediction of hydraulic properties of soils using particle-size distribution and bulk density data. In van Genuchten, M., Leij, F. J., and Lund, L. J., editors, *International Workshop on" Indirect Methods for Estimating the Hydraulic Properties of Unsaturated Soils "*, pages 317–328.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, 26(7):1483–1496.

Duan, Q. and Gupta, V. (1992). Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research*, 28(4):1015–1031.

Duan, Q., Sorroshian, S., and Gupta, V. K. (1994). Optimal use of the SCE-UA global optimization method for calibrating watershed models. *Journal of Hydrology*, 158:265–284.

Hills, R. G., Hudson, D., Porro, I., and Wierenga, P. J. (1989). Modeling one-dimensional infiltration into very dry soils 2. Estiamtion of the soil water parameters and model predictions. *Water Resources Research*, 25(6):1271–1282.

Iden, S. C., Diamantopoulos, E., and Durner, W. (2021). Capillary, film, and vapor flow in transient bare soil evaporation (2): Experimental identification of hydraulic conductivity in the medium to dry moisture range. *Water Resources Research*, 57:e2020WR028514.

Iden, S. C. and Durner, W. (2014). Comment on "Simple consistent models for water retention any hydraulic conductivity in the complete moisture range" by A. Peters. *Water Resources Research*, 50:7530–7534.

Kelley, C. T. (2018). Numerical methods for nonlinear equations. *Acta Numerica*, 27:207–287.

Kubelka, P. and Munk, F. (1931). Ein beitrag zur optik der farbanstriche. *Zeitschrift für Technische Physik*, 12:593–601.

Latorre, B. and Moret-Fernández, D. (2019). Simultaneous estimation of the soil hydraulic conductivity and the van Genuchten water retention parameters from an upward infiltration experiment. *Journal of Hydrology*, 572(March):461–469.

Lebeau, M. and Konrad, J. M. (2010). A new capillary and thin film flow model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 46:W12554.

Lehmann, F. and Ackerer, P. H. (1998). Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media. *Transport in Porous Media*, 31:275–292.

Logg, A., Mardal, K. A., and Wells, G. N. (2012). *Automated solution of differential equations by the finite element method*. Springer.

Luo, Y., Ghezzehei, T. A., Yu, Z., and Berli, M. (2020). Modeling near-surface water redistribution in a desert soil. *Vadose Zone Journal*, 19.

Miller, E. E. and Miller, R. D. (1956). Physical theory for capillary flow phenomena. *Journal of Applied Physics*, 27(4):324–332.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Neuman, S. P. (1973). Saturated-unsaturated seepage by finite elements. *Jounral of the Hydraulic Division*, 99(12):2233–2250.

Nimmo, J. R. (1991). Reply [to "Comment on the treatment of residual water content in 'a consistent set of parametric models for the two-phase flow of immiscible fluids in the subsurface' by L. Luckner et al."]. *Water Resources Research*, 27(4):661–662.

Peters, A. (2013). Simple consistent models for water retention and hydraulic conductivity in the complete moisture range. *Water Resources Research*, 49:6765–6780.

Peters, A. (2014). Reply to comment by S. Iden and W. Durner on "Simple consistent for water retention and hydraulic conductivity in the complete moisture range". *Water Resources Research*, 50:7535–7539.

Peters, A. and Durner, W. (2008). A simple model for describing hydraulic conductivity in unsaturated porous media accounting for film and capillary flow. *Water Resources Research*, 44:W11417.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333.

Richardson, L. F. (1922). *Weather prediction by numerical process*. Cambridge University Press, Cambridge, United Kingdom.

Sadeghi, M., Babaeian, E., Tuller, M., and Jones, S. B. (2018). Particle size effects on soil reflectance explained by an analytical radiative transfer model. *Remote Sensing of Environment*, 210:375–386.

Sadeghi, M., Ghahraman, B., Warrick, A. W., Tuller, M., and Jones, S. B. (2016). A critical evaluation of the Miller and Miller similar media theory for application to natural soils. *Water Resources Research*, 52:3829–3846.

Sadeghi, M., Jones, S. B., and Philpot, W. D. (2015). A linear physically-based model for remote sensing of soil moisture using short wave infrared bands. *Remote Sensing of Environment*, 164:66–76.

Sadeghi, M., Sheng, W., Babaeian, E., Tuller, M., and Jones, S. B. (2017). High-resolution shortwave infrared imaging of water infiltration into dry soil. *Vadose Zone Journal*, 16(13).

Schaap, M. G. and Leij, F. J. (2000). Improved prediction of unsaturated hydraulic conductivity with the Mualem-van Genuchten model. *Soil Sci. Soc. Am. J.*, 64:843–851.

Schneider, M. and Goss, K.-U. (2012). Prediction of the water sorption isotherm in air dry soils. *Geoderma*, 170:64–69.

Scudeler, C., Putti, M., and Paniconi, C. (2016). Mass-conservative reconstruction of Galerkin velocity fields for transport simulations. *Advances in Water Resources*, 94:470–485.

Sheng, W., Zhou, R., Sadeghi, M., Babaeian, E., Robinson, D. A., Tuller, M., and Jones, S. B. (2017). A TDR array probe for monitoring near-surface soil moisture distribution. *Vadose Zone Journal*, 16(4):1–8.

Šimůnek, J., Wendroth, O., Wypler, N., and Van T, M. (2001). Non-equilibrium water flow characterized by means of upward infiltration experiments. *European Journal of Soil Science*, 52(1):13–24.

Tokunaga, T. K. (2009). Hydraulic properties of adsorbed water films in unsaturated porous media. *Water Resources Research*, 45:W06415.

Tuller, M., Dani, O., and Dudley, L. M. (1999). Adsorption and capillary condensation in porous media: Liquid retention and interfacial configurations in angular pores. *Water Resources Research*, 35(7):1949–1964.

Tuller, M. and Or, D. (2001). Hydraulic conductivity of variably saturated porous media: Film and corner flow in angular pore space. *Water Resources Research*, 37(5):1257–1276.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America*, 44:892–898.

Vogel, T., Huang, K., Zhang, R., and van Genuchten, M. T. (1996). Hydrus code for simulating one-dimensional water flow, solute transport, and heat movement in variably-saturated media: Version 5.0. Technical Report August, U. S. Salinity Laboratory, Agricultural Research Service, U. S. Department of Agriculture, Riverside, CA, USA.

Vogel, T., Van Genuchten, M. T., and Cislerova, M. (2001). Effect of the shape of the soil hydraulic functions near saturation on variably-saturated flow predictions. *Advances in Water Resources*, 24(2):133–144.

Zhang, Z. F. (2011). Soil water retention and relative permeability for conditions from oven-dry to full Saturation. *Vadose Zone Journal*, 10:1299–1308.

Chapter 4

Estimation of surface water flux from soil moisture measurement: Comparison between physics-informed neural networks and adjoint method

Abstract. [1]Surface water flux is an important variable for water resource management. Accurate measurement of surface water flux is only possible by building costly lysimeters. In this work, we seek to use near-surface soil moisture measurements to estimate surface water flux. Surface water flux can be interpreted as a boundary condition for a non-linear time-dependent Richardson-Richards equation (RRE), a partial differential equation (PDE) that models water transport in soils. Thus, estimating surface water flux can be framed as estimating this boundary condition of the PDE from near-surface soil moisture measurements. Solving this PDE-constrained optimization problem is computationally challenging due to the large parameter dimension stemming from discretization. Here, we used first and second-order adjoint-based derivative information and applied Newton's method to minimize the objective function. We compared the performance of this approach with physics-informed neural networks. Both methods gave comparable estimates of surface water flux in three numerical experiments. However, the adjoint method was more robust in terms of the reconstructed soil moisture profile under a data-limited situation.

[1]The preliminary work of the current chapter was presented as Bandai, T., Ghezzehei, T. A., Petra, N. Estimation of rainfall rates from soil moisture measurements through PDE-constrained optimization. HydroML Symposium, Pennsylvania, USA, (May 2022).

4.1 Introduction

In the last few decades, measuring soil moisture has become routine in hydrological and meteorological field stations. Recently, the networks of soil moisture measurements have been organized to foster their usage (Cosh et al., 2021; Dorigo et al., 2021). Most soil moisture monitoring programs were established to verify soil moisture estimates provided by satellite remote sensing. Also, soil moisture data enable us to assess drought conditions. While these direct usages of soil moisture measurement will continue to be valuable, we should aim to extract more information from soil moisture measurements. There is no doubt that soil moisture data contain information on water flow and soil hydraulic properties. While data-driven approaches might be popular at the moment, it is indispensable to develop physics-based algorithms to extract information from soil moisture data for interpretability and extrapolation capability (Willcox et al., 2021).

Brocca et al. (2013) developed an algorithm based on the mass conservation of water to estimate surface water flux (i.e., infiltration and evaporation) from soil moisture measurements. However, the mass conservation of water is not enough to describe the strong dependency of water flow on the wetness of soils. By combining the Buckingham-Darcy law (Buckingham, 1907), which states that water flow in soils is driven by the gradient of the potential energy of water, with the mass conservation, a non-linear time-dependent partial differential equation (PDE) called the Richardson-Richards equation (RRE) was established to simulate water flow in soils (Richardson, 1922; Richards, 1931). While the RRE has been successfully used to analyze soil moisture data from laboratory experiments, the application of the RRE to field soil moisture measurements is still challenging. This is because of the uncertainty in soil hydraulic properties and boundary conditions. In the last two decades, attention has been focused on estimating soil hydraulic properties by inverse modeling based on the RRE. While the constant (or zero water potential gradient) lower boundary condition might be justifiable in field conditions, the upper boundary condition must be precisely specified. However, rainfall rates measured by tipping bucket rain gauges are affected by wind and contain many measurement errors (Pollock et al., 2018). Also, the eddy covariance method to estimate evaporation is costly and does not necessarily provide appropriate estimates because of scale mismatch. The only avaialble way to accurately measure the upper boundary condition is to build lysimeters that capture all water gain or loss. In fact, soil moisture dynamics in fields were successfully analyzed by the RRE if well-defined boundary conditions were given by lysimeters (Dijkema et al., 2017; Groh et al., 2018). However, building and maintaining lysimeters are expensive and time-consuming, and standard hydrological and meteorological stations are not equipped with lysimeters. Therefore, we need to optimize the upper boundary condition in addition to soil hydraulic properties by soil moisture data. To achieve this goal, Sadeghi et al. (2019) employed an analytical solution to the linearized RRE. Although the analytical model has shown promising results, its accuracy and applicability for extreme wet and dry conditions are limited, probably due to the linearization.

The inverse problem to estimate the upper boundary (i.e., surface water flux) is a large-scale optimization problem because the number of parameters is the number of time steps for the RRE. To overcome this challenge, Bandai and Ghezzehei (2021) and Bandai and Ghezzehei (2022) applied a neural network (NN) based inverse method called physics-informed neural networks (PINNs) (Raissi et al., 2019). PINNs approximate the solution to PDEs based on the so-called universal approximation capability of NNs (Cybenko, 1989; Hornik, 1991). PINNs have been used to solve the forward and inverse problems governed by various PDEs, such as Navier-Stokes equation (Jin et al., 2021). In the PINN framework, forward and inverse problems governed by PDEs are translated into a minimization problem, where gradient-based optimization methods are used. The success of PINNs hinges on the exact evaluation of the gradient of the objective function and the partial derivatives in PDEs by automatic differentiation (AD) implemented on highly efficient machine learning libraries, such as TensorFlow (Abadi et al., 2015). Although traditional methods, such as finite difference and finite element methods, are more stable and efficient for forward modeling, PINNs are quite promising for inverse modeling because PINNs do not require well-defined initial and boundary conditions. Bandai and Ghezzehei (2021) demonstrated that PINNs could be used to estimate surface water flux from near-surface soil moisture measurements in a homogeneous soil using synthetic data. In their framework, soil hydraulic properties are approximated by NNs, and surface water flux is estimated by the Buckingham Darcy law with the estimated hydraulic conductivity and the water potential gradient at the surface computed by AD. Although they demonstrated the potential of PINNs, the results depend on the amount of available data and extensive hyperparameter tunings. Thus, in their following study, the capability of PINNs to solve the RRE given soil hydraulic properties was systematically investigated by analytical solutions to the RRE for homogeneous and two-layer soils (Bandai and Ghezzehei, 2022). They demonstrated that the layer-wise locally adaptive activation function proposed by Jagtap et al. (2020) was effective in training NNs among other methods. More comprehensive reviews on PINNs are given by Karniadakis et al. (2021).

Behind the increasing attention and popularity of machine learning, a more classical method called an adjoint method from optimal control theory had been steadily improved to solve large-scale inverse problems governed by PDEs in the last few decades (Ghattas and Willcox, 2021). In the adjoint method, the inexact Newton method (Eisenstat and Walker, 1996) is used to solve the minimization problem from the inverse problem. The linear system resulting from the Newton method has the gradient of the objective function and the Hessian. The linear system is solved by a conjugate gradient method, which only requires the Hessian-vector product, not the Hessian itself. This is the key to the application of the adjoint method to large-scale inverse problems because the construction of the Hessian itself is intractable. In the framework, the gradient and the Hessian-vector product are computed by solving a so-called adjoint PDE and the linearized forward and adjoint PDEs. The adjoint method has been successfully applied to large-scale inverse problems in geosciences, such as ice sheets in Antarctica (Isaac et al., 2015). However, the application of

adjoint methods to the RRE has not been achieved except for a few studies, such as Santos et al. (2006), where an adjoint method was used to estimate saturated hydraulic conductivity field from water potential measurements. Nevertheless, the study used only the gradient, not the Hessian information.

Because a close relation between adjoint methods and AD is recognized (Haber and Ruthotto, 2018), there must be similarities between them. By comparing the two methods, we can improve our understanding of the advantage of each method and their limitations. Mowlavi and Nabi (2022) compared the two methods and demonstrated that both methods gave comparable results for various inverse problems (i.e., Laplace, Burgers, Kuromoto-Sivashinsky, and Navier-Stokes equation). In the study, the adjoint method only computes the gradient, not the Hessian-vector product.

The objective of the study is to compare the two methods for the inverse problem governed by the RRE in the context of the estimation of surface water flux from soil moisture measurements. Specifically, we investigate the accuracy of the estimated surface water flux in the presence of noise in the data, the effect of regularization parameters, and computational demand for a large-scale problem through three numerical examples. While soil hydraulic properties are also critical sources of uncertainty, we assumed they were known and focused on the estimation of the upper boundary condition to study the characteristics of the two methods.

4.2 Forward problem: Vertical water flow in soils

Vertical soil moisture dynamics in a rigid, isotropic, homogeneous soil on a spatial domain $\Omega := (-Z, 0)$ can be described by the Richardson-Richards equation (RRE):

$$\frac{\partial \theta(\psi)}{\partial t} = \frac{\partial}{\partial z}\left(K(\psi)\frac{\partial \psi}{\partial z} + K(\psi)\right) \quad \text{in} \quad \Omega \times (0, T), \tag{4.1}$$

where t is the time [T], T is the final time [T], z is the spatial coordinate with positive upward with $z = 0$ set to the surface of the soil, Z is the length of the soil [L], ψ is the water potential [L], θ is the volumetric water content [L^3 L^{-3}], $K(\psi)$ is the hydraulic conductivity [L T^{-1}]. Physical concepts behind Equation 4.1 stem from the extension of Darcy's law for saturated porous media to unsaturated flow by Buckingham (1907) and the subsequent formulation by Richardson and Richards (Richardson, 1922; Richards, 1931). The volumetric water content θ and hydraulic conductivity K are both non-linear functions of the water potential ψ and represent the macroscopic water transport properties of the soil. These two functions are referred to as the water retention curve (WRC) and hydraulic conductivity function (HCF). The two functions are affected by water retention and flow processes in the complicated pore geometry (Assouline and Or, 2013) and are commonly determined by laboratory experiments or inverse modeling. Note that WRCs $\theta(\psi)$ exhibit hysteresis under wetting and drying (i.e., θ is not a unique function of ψ but also depends on its trajectory). However, we neglected the effect of hysteresis in this study.

Several parametric models are available to represent WRCs $\theta(\psi)$ and HCFs $K(\psi)$ (Brooks and Corey, 1964; Mualem, 1976; van Genuchten, 1980; Kosugi, 1996). A key concept is that both θ and K depend on the geometry of pore spaces in soils, and they are not independent (i.e., share parameters). We selected the Brooks and Corey (BC) model (Brooks and Corey, 1964) for simplicity. However, the mathematical framework presented here can be applied to other models, such as the van Genuchten- Mualem model (Mualem, 1976; van Genuchten, 1980). WRCs $\theta(\psi)$ for the BC model is

$$\theta(\psi) = \theta_r + (\theta_s - \theta_r)S_e(\psi) = \theta_r + \left(\frac{\psi}{\psi_c}\right)^{-\lambda} \quad \text{for} \quad \psi < \psi_c, \tag{4.2}$$

$$\theta(\psi) = \theta_s \quad \text{for} \quad \psi \geq \psi_c, \tag{4.3}$$

where θ_r is the residual volumetric water content [-], θ_s is the saturated volumetric water content [-], S_e is the effective saturation [-], ψ_c is the air-entry value [L], λ is the pore-size distribution parameter [-]. Usually, λ ranges between 0.3 (wide pore-size distribution) and 10.0 (uniform pore-size distribution). For $\psi < \psi_c$, the first and second derivatives of $\theta(\psi)$ are as follows:

$$\frac{\partial\theta}{\partial\psi} = -\frac{(\theta_s - \theta_r)\lambda}{\psi}\left(\frac{\psi}{\psi_c}\right)^{-\lambda}, \tag{4.4}$$

$$\frac{\partial^2\theta}{\partial\psi^2} = \frac{(\theta_s - \theta_r)\lambda(\lambda+1)}{\psi^2}\left(\frac{\psi}{\psi_c}\right)^{-\lambda}. \tag{4.5}$$

HCFs $K(\psi)$ for the BC model is

$$K(\psi) = K_s S_e^{l+2+2/\lambda} \quad \text{for} \quad \psi < \psi_c, \tag{4.6}$$

$$K(\psi) = K_s \quad \text{for} \quad \psi \geq \psi_c, \tag{4.7}$$

where K_s is the saturated hydraulic conductivity [L T^{-1}], l is the tortousity parameter [-]. For $\psi < \psi_c$, the first and second derivatives of $K(\psi)$ are as follows:

$$\frac{\partial K}{\partial\psi} = -\frac{K_s(l+2+2\lambda)\lambda}{\psi_c} S_e^{l+3+2/\lambda}, \tag{4.8}$$

$$\frac{\partial^2 K}{\partial\psi^2} = \frac{K_s(l+2+2\lambda)(l+2+3\lambda)\lambda^2}{\psi_c^2} S_e^{l+2+4/\lambda}. \tag{4.9}$$

This study used the BC parameters for sandy loam soil provided in Rawls et al. (1982). Also, we changed the parameter λ to demonstrate the effect of the non-linearity of WRCs. Table 4.1 shows the BC parameters for the original and modified values, which are referred to as Sandy Loam A and B, respectively. The corresponding WRCs $\theta(\psi)$ and HCFs $K(\psi)$ and their first and second derivatives are shown in Figure 4.1, where Sandy Loam A exhibits stronger non-linearity than Sandy Loam B.

To simulate vertical movement of water in soils, we imposed a Neumann flux boundary condition on the top boundary $\partial\Omega_N := \{z|z = 0\}$ while the bottom boundary $\Omega_D := \{z|z = -Z\}$ is the constant Dirichlet boundary condition along with the

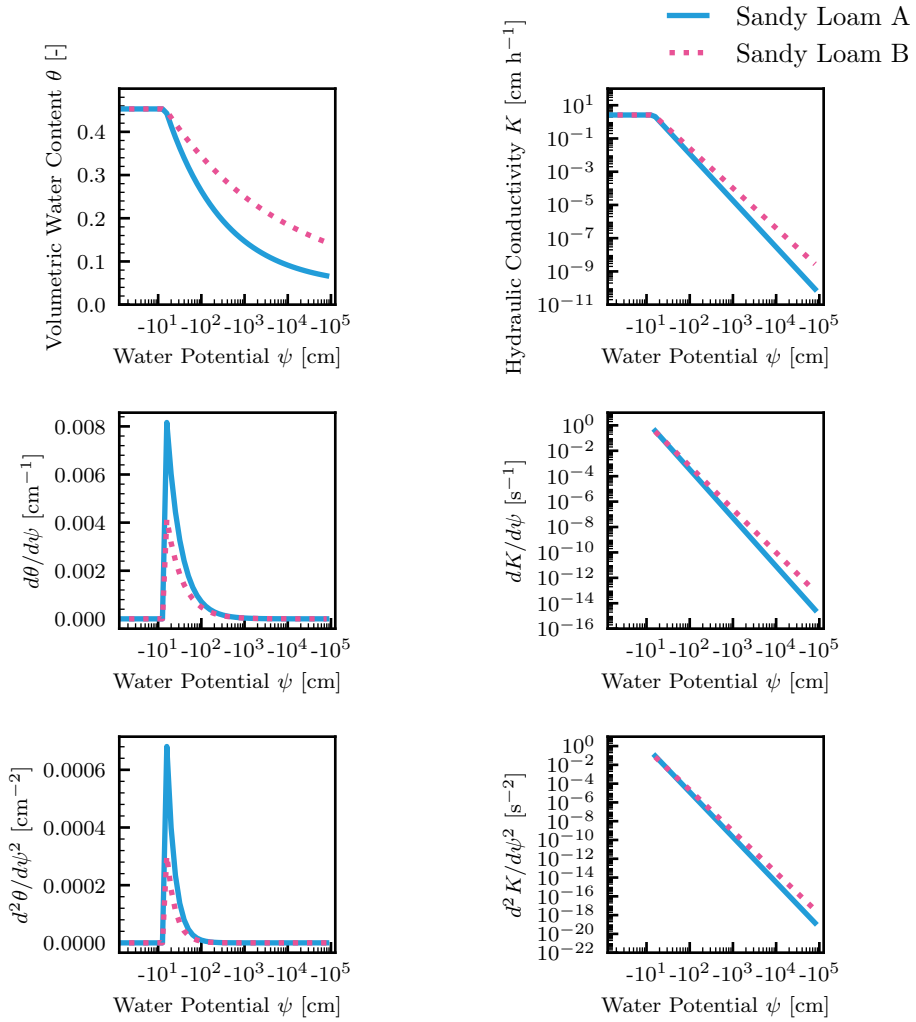

Figure 4.1: Water retention curves $\theta(\psi)$ and hydraulic conductivity functions $K(\psi)$ and their first and second derivatives for Sandy Loam A and Sandy Loam B (see Table 4.1). The derivatives were set to zero for $\psi \geq \psi_c$ and thus not visible for the derivatives of K.

Table 4.1: Brooks and Corey parameters for the soils used in the study (Rawls et al., 1982).

Soil Type	θ_s [-]	θ_r [-]	ψ_c [cm]	λ [-]	K_s [cm h^{-1}]	l [-]
Sandy Loam A	0.453	0.041	-14.66	0.322	2.59	0.5
Sandy Loam B	0.453	0.041	-14.66	0.100	2.59	0.5

initial condition:

$$\psi(z,0) = \psi_0(z), \quad \text{on} \quad \Omega, \tag{4.10}$$

$$\psi(-Z,t) = \psi_D(t), \quad \text{on} \quad \Omega_D \quad t \in (0,T), \tag{4.11}$$

$$q(z,t) := -K(z,t)\left(\frac{\partial \psi(z,t)}{\partial z} + 1\right) = -m(t), \quad \text{on} \quad \Omega_F, \quad t \in (0,T), \tag{4.12}$$

where $\psi_0(x)$ is the initial condition [L], q is the water flux [L T^{-1}], $m(t)$ is the Neumann flux boundary condition representing the water flux into the soil [L T^{-1}], which is defined to be positive when water flows into the domain, ψ_D is the water potential at the Dirichlet boundary [L].

4.2.1 Finite element solution to forward problem

We solved the RRE (Equation 4.1) with the initial and boundary conditions (Equations 4.10 – 4.12) by a finite element method with mass lumping (Celia et al., 1990; Guermond and Pasquetti, 2013). Let us introduce trial and test spaces $\mathcal{V}$ and $\hat{\mathcal{V}}$, respectively, where

$$\mathcal{V} = \{v \in H^1(\Omega) : v = \psi_D(t) \quad \text{on} \quad z \in \partial\Omega_D \quad \text{for each} \quad t \in (0,T)\}, \tag{4.13}$$

$$\hat{\mathcal{V}} = \{v \in H^1(\Omega) : v = 0 \quad \text{on} \quad z \in \partial\Omega_D \quad \text{for each} \quad t \in (0,T)\}. \tag{4.14}$$

Here, $H^1(\Omega)$ is the Sololev space containing functions v such that v^2 and $|\nabla v|^2$ have finite integrals over Ω. We multiplied the residual of the PDE by a test function $p \in \hat{\mathcal{V}}$ and integrated it by parts in space, which results in a continuous variational problem, where we seek $\psi \in \mathcal{V}$ such that

$$\int_\Omega \frac{\partial \theta(\psi)}{\partial t} p \, dz + \int_\Omega K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\frac{\partial p}{\partial z}\, dz - \int_{\Omega_F} m(t)p(z)\, ds = 0 \tag{4.15}$$

for all $p \in \hat{\mathcal{V}}$. We dicretized the continuous variational problem into a discrete variational problem by introducing finite dimensional trial and test spaces $\mathcal{V}_h \in \mathcal{V}$ and $\hat{\mathcal{V}}_h \in \hat{\mathcal{V}}$, respectively. Linear finite elements with N_s nodes were used for the basis functions of $\mathcal{V}_h$ and $\hat{\mathcal{V}}_h$. The volumetric water θ and the hydraulic conductivity K were also approximated by the linear functions with coefficients computed from the water potential ψ at each node with the given WRC and HCF (Scudeler et al., 2016). Thus,

$$\psi(x,t) \approx \psi_h(z,t) = \sum_{i=1}^{N_s} \psi_i(t)\phi_i(z), \tag{4.16}$$

$$\theta(z,t) \approx \theta_h(z,t) = \sum_{i=1}^{N_s} \theta(\psi_i(t))\phi_i(z) = \sum_{i=1}^{N_s} \theta_i(t)\phi_i(z), \tag{4.17}$$

$$K(z,t) \approx K_h(z,t) = \sum_{i=1}^{N_s} K(\psi_i(t))\phi_i(z) = \sum_{i=1}^{N_s} K_i(t)\phi_i(z), \tag{4.18}$$

where N_s is the number of the spatial nodes, ϕ_i is the linear function at node i. Then, the discrete variational problem for time t becomes to find $[\psi_1(t), \psi_2(t), ..., \psi_{N_s}(t)]^T$ such that

$$\int_\Omega \frac{\partial \theta_h}{\partial t} p_h \, dz + \int_\Omega K_h \left(\frac{\partial \psi_h}{\partial z} + 1 \right) \frac{\partial p_h}{\partial z} \, dz - \int_{\Omega_F} m(t) p_h(z) \, ds = 0 \tag{4.19}$$

for all $p_h \in \hat{\mathcal{V}}_h$. The temporal derivative was approximated by the backward Euler method. For that purpose, we introduce the times t^n for $n = 0, 1, ..., N_t$, such that

$$0 = t^0 < t^1 \cdots < t^n < \cdots < t^{N_t} = T, \tag{4.20}$$

and the corresponding time steps $\Delta t^n = t^n - t^{n-1}$ for $n = 1, ..., N_t$, where the superscript n represents the time $t = t^n$. Constant time stepping was used for simplicity in the study (i.e., $\Delta t^n = \Delta t$ for $n = 1, ..., N_t$.). As for the boundary condition m, let us introduce another functional space

$$\mathcal{M} = \{v \in H^1(0, T)\}, \tag{4.21}$$

and m is discretized on by a discrete space $\mathcal{M}_h \in \mathcal{M}$. Linear finite elements were used to represent the discretized m:

$$m(t) \approx m_h(t) = \sum_{i=0}^{N_t} m_i \chi_i(t), \tag{4.22}$$

where $\chi_i(t)$ is the linear function at node i.

ψ_h^0 was obtained by interpolating the initial condition ψ_0. For the time $t = t^n$ with $n = 1, ..., N_t$, the discrete variational problem is finding $\boldsymbol{\psi}^n := [\psi_1^n, \psi_2^n, ..., \psi_{N_s}^n]^T$ such that

$$\int_\Omega \frac{\theta_h^n - \theta_h^{n-1}}{\Delta t^n} p_h \, dz + \int_\Omega K_h^n \left(\frac{\partial \psi_h^n}{\partial z} + 1 \right) \frac{\partial p_h}{\partial z} \, dz - \int_{\Omega_F} m_h(t^n) p_h(z) \, ds = 0, \tag{4.23}$$

for all $p_h \in \hat{\mathcal{V}}_h$. The variational problem above is non-linear in terms of $\boldsymbol{\psi}^n$ because of the non-linear functions θ and K. The resulting system of non-linear equations was solved by Newton's method with Armijo line search (Armijo, 1966; Kelley, 2018). After replacing the test function p_h by ϕ_i for $i = 1, 2, ..., N_s$ and using mass lumping on the mass matrix in the temporal derivative term to obtain a smooth solution (Celia et al., 1990; Guermond and Pasquetti, 2013), the system of non-linear equations for $t = t^n$ can be written as:

$$\mathbf{F}^n(\boldsymbol{\psi}^n) = \mathbf{0}, \tag{4.24}$$

where $\mathbf{F}^n := [F_1^n(\boldsymbol{\psi}^n), F_2^n(\boldsymbol{\psi}^n), ..., F_{N_s}^n(\boldsymbol{\psi}^n)]^T$ for $n = 1, 2, ..., N_t$ with

$$F_i^n(\boldsymbol{\psi}^n) = \mathbf{M}^L(\boldsymbol{\theta}^n - \boldsymbol{\theta}^{n-1}) + \Delta t(\mathbf{K}\boldsymbol{\psi}^n + \mathbf{b}_1 - \mathbf{b}_2) \tag{4.25}$$

where $\boldsymbol{\theta}^n := [\theta_1^n, \theta_2^n, ..., \theta_{N_s}^n]^T$, $\mathbf{M}^L$ is the lumped mass matrix whose ijth element is

$$\mathbf{M}_{ij}^L = \left(\sum_j \mathbf{M}_{ij} \right) \delta_{ij}, \tag{4.26}$$

where $\mathbf{M}$ is the mass matrix with its ijth element $\mathbf{M}_{ij} = \int_\Omega \phi_i \phi_j \, dz$, δ_{ij} is the Kronecker symbol, $\mathbf{K}$ is the stiffness matrix with a non-linear coefficient, whose ijth element is

$$\mathbf{K}_{ij} = \int_\Omega K_h \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z} \, dz, \tag{4.27}$$

$\mathbf{b}_1$ is the vector with its ith element

$$\mathbf{b}_{1,i} = \int_\Omega K_h \frac{\partial \phi_i}{\partial z} \, dz, \tag{4.28}$$

$\mathbf{b}_2$ is the vector with its ith element

$$\mathbf{b}_{2,i} = \int_{\Omega_F} m_h(t^n) \phi_i \, ds. \tag{4.29}$$

The integrals for $\mathbf{K}$ and $\mathbf{b}_1$ were evaluated by the Gauss integration with one point and unit weight, while other integrals were evaluated exactly (Scudeler et al., 2016). Because the initial condition ψ_0 was given and interpolated to obtain ψ_i^0 for $i = 1, 2, ..., N_s$, we define $\mathbf{F}^0(\boldsymbol{\psi}^0) := [F_1^0(\boldsymbol{\psi}^0), F_2^0(\boldsymbol{\psi}^0), ..., F_{N_s}^0(\boldsymbol{\psi}^0)]^T = \mathbf{0}$ with

$$F_i^0(\boldsymbol{\psi}^0) = \psi_i^0 - \psi_0(z = z_i), \tag{4.30}$$

where z_j is the spatial coordinate corresponding to the node i.

We assemble all $\mathbf{F}^n$ for $n = 0, 1, ..., N_t$ as $\bar{\mathbf{F}} := [(\mathbf{F}^0)^T, (\mathbf{F}^1)^T, ..., (\mathbf{F}^{N_t})^T]^T \in \mathbb{R}^{N_s(N_t+1)}$. In the same manner, we assemble the state variable as $\bar{\boldsymbol{\psi}} := [(\boldsymbol{\psi}^0)^T, (\boldsymbol{\psi}^1)^T, ..., (\boldsymbol{\psi}^{N_t})^T]^T \in \mathbb{R}^{N_s(N_t+1)}$. They will be used to derive the adjoint equation in the next section.

For $t = t^n$ with $n = 1, 2, ..., N_t$, the system of non-linear equations $\mathbf{F}^n$ was solved by the Newton method. Let $\boldsymbol{\psi}^{n,k}$ denote the solution at the kth Newton iteration for $k = 0, 1, 2, ...$ with $\boldsymbol{\psi}^{n,0} = \boldsymbol{\psi}^{n-1}$. For the kth Newton iteration, the Newton direction $\mathbf{d}^k$ is determined by solving the Newton system:

$$\mathbf{F}'(\boldsymbol{\psi}^{n,k}) \mathbf{d}^k = -\mathbf{F}^n(\boldsymbol{\psi}^{n,k}), \tag{4.31}$$

where the Jacobian matrix $\mathbf{F}' := \frac{\partial \mathbf{F}}{\partial \boldsymbol{\psi}}$ (see Sect. B.1.1) was computed by analytical $\theta'(\psi)$ and $K'(\psi)$, and the linear system was solved by the Gaussian elimination. The Newton step size is determined by Armijo line search:

$$\boldsymbol{\psi}^{n,k+1} = \boldsymbol{\psi}^{n,k} + \lambda \mathbf{d}^k, \tag{4.32}$$

with the initial $\alpha = 1$, and α is reduced by a factor 0.5 when the sufficient decrease condition

$$||\mathbf{F}(\boldsymbol{\psi}^{n,k+1})|| < (1 - c\alpha)||\mathbf{F}(\boldsymbol{\psi}^{n,k})||, \tag{4.33}$$

where $c = 10^{-4}$, is not met. The Newton iteration was terminated when

$$||\mathbf{F}(\boldsymbol{\psi}^{n,k})|| \leq \tau_a + \tau_r ||\mathbf{F}(\boldsymbol{\psi}^{n,0})||, \tag{4.34}$$

where $\tau_a = 10^{-7}$, $\tau_r = 10^{-7}$.

4.3 Inverse modeling: Estimation of surface water flux from soil moisture measurements

In this section, we describe the inverse problem to estimate surface water flux from soil moisture measurements based on the RRE. We employed two methods: adjoint methods and physics-informed neural networks (PINNs). In the following subsections, adjoint methods and PINNs for the RRE are explained, and the comparison between the two algorithms is discussed at the end of this section. We may unavoidably use the same notation for adjoint methods and PINNs, but we will ensure that there is no ambiguity.

4.3.1 Adjoint methods

In adjoint methods, there are two approaches to derive the gradient and the Hessian information (Hessian-vector product): discretize-then-optimize (DTO) and optimize-then-discretize (OTD) approaches Chavent (2010). The gradient and the Hessian-vector product obtained by the two approaches do not always coincide (Wilcox et al., 2015). Our implementation is based on the DTO approach because of the stabilized finite element method (i.e., mass lumping) to solve the forward problem. However, the OTD approach provides cleaner and more insightful expressions, and we present both derivations and remark on the difference between the two approaches when appropriate.

4.3.1.1 Regularized inverse problem

The inverse problem is to estimate surface water flux m given (possibly noisy) soil moisture data θ_{obs} measured at $z \in \{z^1_{\text{obs}}, .., z^{N_{\text{obs}}}_{\text{obs}}\}$ by N_{obs} sensors, which are sparsely located but continuously measure volumetric water content θ. This inverse problem can be formulated as minimizing the following functional:

$$\min_m \mathcal{J}(m) := \frac{1}{2} \int_0^T \int_\Omega (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})^2 \, dzdt + \mathcal{R}(m), \tag{4.35}$$

where ψ is the solution to the forward problem given the WRC $\theta(\psi)$ and HCF $K(\psi)$, $\mathcal{B}$ is an observational operator that extracts soil moisture at the measurement locations, and $\mathcal{R}$ is an regularization term to penalize oscillatory components of m:

$$\mathcal{R}(m) := \frac{\gamma}{2} \int_0^T \left(\frac{dm(t)}{dt} \right)^2 dt, \tag{4.36}$$

where $\gamma > 0$ is the regularization parameter. This regularization term is required to prevent the inverse solution from being severely degraded by the noise in the data.

The objective functional $\mathcal{J}(m)$ is discretized by approximating ψ, θ, and m in the same way as Sect. 4.2.1. The first term of the objective functional can be approximated as:

$$\frac{1}{2}\int_0^T \int_\Omega (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})^2 \, dzdt \approx \frac{1}{2}\int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t)) \, dt, \quad (4.37)$$

where $\mathbf{B} \in \mathbb{R}^{N_{\text{obs}} \times N_s}$ is a matrix consisting of zero and one, $\boldsymbol{\theta}_{\text{obs}}(t) \in \mathbb{R}^{N_{\text{obs}}}$ is the vector consisting of the volumetric water content measured by N_{obs} sensors. After discretizing the regularization term, we obtain the following discretized objective function:

$$J(\mathbf{m}) := \frac{1}{2}\int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t)) \, dt + \frac{\gamma}{2}\mathbf{m}^T \mathbf{K}^m \mathbf{m}, \quad (4.38)$$

where $\mathbf{m} = [m^0, m^1, ..., m^{N_t}]^T$, $\mathbf{K}^m$ is the stiffness matrix for the basis function χ_i, whose ijth element is $K^m_{i,j} = \int_0^T \nabla\chi_i \cdot \nabla\chi_j \, dt$. The temporal integration in the first term was approximated by a Trapezoidal rule, while the stiffness matrix $\mathbf{K}^m$ was evaluated analytically.

4.3.1.2 Inexact Newton-CG method to solve inverse problem

To minimize the discretized objective function (Eq. 4.38), we employed the inexact Newton-conjugate gradient (Newton-CG) method (Nocedal and Wright, 2006). The Newton-CG method is quadratically convergent near the solution, and globalization was achieved by the Armijo line search (Armijo, 1966). More detailed explanations of the Newton-CG method can be found in Petra et al. (2012) and Ghattas and Willcox (2021).

Let us denote the gradient and the Hessian of the objective function as $\mathbf{g} \in \mathbb{R}^{N_m}$ and $\mathbf{H} \in \mathbb{R}^{N_m \times N_s}$, respectively. The kth Newton iteration is written as

$$\mathbf{H}^k \tilde{\mathbf{m}}^k = -\mathbf{g}^k, \quad (4.39)$$

where the superscript k represents the kth Newton iteration, $\mathbf{H}^k$ and $\mathbf{g}^k$ are the gradient and the Hessian at the kth solution $\mathbf{m}^k$, $\tilde{\mathbf{m}}^k$ is the Newton direction. Then, the Armijo line search follows:

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha\tilde{\mathbf{m}}^k, \quad (4.40)$$

where the initial $\alpha = 1$ and reduced by a factor 0.5 when the sufficient decrease condition

$$J(\mathbf{m}^{k+1}) < J(\mathbf{m}^k) - c\alpha(\mathbf{g}^k)^T \tilde{\mathbf{m}}^k, \quad (4.41)$$

with $c = 10^{-4}$, is not met.

Solving the linear system (Eq. 4.39) for large N_m is computationally very expensive because the construction of the Hessian matrix $\mathbf{H}^k$ requires solving linearized forward and adjoint problems, which will be described later, N_m times. Thus, the linear system is solved by a conjugate gradient method, where the only Hessian-vector product is needed, not the Hessian matrix $\mathbf{H}^k$ itself. The CG iterations were preconditioned by the regularization operator

$$\mathbf{P} := \gamma \mathbf{K}^m + 0.1\gamma \mathbf{M}^m, \tag{4.42}$$

where $\mathbf{K}^m$ and $\mathbf{M}^m$ are the stiffness and mass matrices of the parameter space, respectively.

The inexact Newton method (Steihaug, 1983; Eisenstat and Walker, 1996) seeks the approximate solution to the linear system by finding the inexact Newton direction $\tilde{\mathbf{m}}^k$ such that

$$||\mathbf{H}^k \tilde{\mathbf{m}}^k + \mathbf{g}^k|| \leq \eta^k ||\mathbf{g}^k||, \tag{4.43}$$

where the left-hand side is the residual of the Newton system (Eq. 4.39), and η^k is called the forcing term, which determines how exactly the Newton system is solved. The forcing term affects the local convergence property of the Newton method and determined as $\eta^k = \min(0.5, \sqrt{\mathbf{g}^k/\mathbf{g}^0})$. The rationale for the η^k is that we do not have to solve the Newton system accurately when the current solution is far from the solution to the inverse problem. Also, we used the Steihaug termination criteria to prevent negative curvature,

$$(\tilde{\mathbf{m}}^k)^T \mathbf{H}^k \tilde{\mathbf{m}}^k \leq 0. \tag{4.44}$$

The Newton iterations are terminated when the norm of the gradient $||\mathbf{g}^k||$ becomes smaller than a threshold set to 10^{-7}. In the following sections, we describe how to obtain the gradient and the Hessian-vector product for both OTD and DTO approaches.

4.3.1.3 Optimize-then-Discretize approach

In this section, we explain the OTD approach to obtain the gradient and the Hessian-vector product of the objective functional (Eq. 4.35) via the formal Lagrangian method (Chavent, 2010). We begin by formulating the Lagrangian for the gradient derivation:

$$\begin{aligned}
\mathcal{L}^g(\psi, p, q, m) &:= \frac{1}{2}\int_0^T \int_\Omega (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})^2 \, dz dt + \frac{\gamma}{2}\int_0^T \left(\frac{dm(t)}{dt}\right)^2 dt \\
&+ \int_0^T \int_\Omega \frac{\partial \theta(\psi)}{\partial t} p + K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\frac{\partial p}{\partial z} \, dz dt \\
&- \int_0^T \int_{\Omega_F} mp \, ds dt + \int_\Omega (\psi(z,0) - \psi_0) q \, dz.
\end{aligned} \tag{4.45}$$

Here, the first term represents the data misfit, the second term represents the regularization, the third and forth terms represent the constraint by the forward problem, and the last term imposes the initial condition weakly via another test function

$q(z) \in \hat{\mathcal{V}}$. We refer the water potential $\psi(z,t)$ and the test function $p(x,t)$ to the state variable and the adjoint variable, respectively. To derive the gradient, we seek the stationary conditions of the Lagrangian $\mathcal{L}^g(\psi,p,q,m)$ with respect to its arguments. The variation of $\mathcal{L}^g$ with respect to p and q simply recovers the forward problem and the initial condition, respectively. By requiring the variation of $\mathcal{L}^g$ with respect to the state variable ψ in the direction $\hat{\psi} \in \hat{\mathcal{V}}$ vanish, we obtain the weak form of the adjoint PDE:

$$\begin{aligned}\mathcal{L}^g_\psi(\psi,p,q,m)(\hat{\psi}) &= \int_0^T \int_\Omega (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\mathcal{B}\theta'(\psi)\hat{\psi}\, dzdt \\ &+ \int_0^T \int_\Omega \frac{\partial \theta'(\psi)\hat{\psi}}{\partial t} p + \left(K'(\psi)\frac{\partial \psi}{\partial z}\hat{\psi} + K(\psi)\frac{\partial \hat{\psi}}{\partial z} + K'(\psi)\hat{\psi} \right) \frac{\partial p}{\partial z}\, dzdt \\ &+ \int_\Omega \hat{\psi}(z,0) q\, dz = 0. \end{aligned} \tag{4.46}$$

The strong form of the adjoint PDE can be recovered by removing $\hat{\psi}$ from the derivative terms by integration by parts:

$$\begin{aligned} &\int_0^T \int_\Omega \left(-\theta'(\psi)\frac{\partial p}{\partial t} + \left(K'(\psi)\frac{\partial \psi}{\partial z} + K'(\psi) \right) \frac{\partial p}{\partial z} - \frac{\partial}{\partial z}\left(K(\psi)\frac{\partial p}{\partial z} \right) \right) \hat{\psi} \\ &+ \int_0^T \int_\Omega \left(\mathcal{B}^*(\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\theta'(\psi) \right) \hat{\psi}\, dzdt \\ &+ \int_\Omega \theta'(\psi(z,T))\hat{\psi}(z,T)p(z,T)dz + \int_\Omega \hat{\psi}(z,0)(q - \theta'(\psi)p(z,0))dz \\ &+ \int_0^T K(\psi(t,0))\frac{\partial p(t,0)}{\partial z}\hat{\psi}(t,0)\, dt = 0 \end{aligned} \tag{4.47}$$

for all $\hat{\psi} \in \hat{\mathcal{V}}$. Here $\mathcal{B}^*$ is the adjoint of the observational operator. Because $\hat{\psi}$ is arbitrary, the third term reveals that $q = \theta'(\psi(z,0))p(z,0)$, and the remaining terms leads to the strong form of the adjoint PDE:

$$\begin{aligned} -\theta'(\psi)\frac{\partial p}{\partial t} + \left(K'(\psi)\frac{\partial \psi}{\partial z} + K'(\psi) \right) \frac{\partial p}{\partial z} &- \frac{\partial}{\partial z}\left(K(\psi)\frac{\partial p}{\partial z} \right) \\ &= -\mathcal{B}^*(\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\theta'(\psi), \end{aligned} \tag{4.48}$$

$$p(z,T) = 0, \tag{4.49}$$

$$p(z,t) = 0 \quad \text{on} \quad \partial\Omega_D, \tag{4.50}$$

$$K(\psi)\frac{\partial p}{\partial z} = 0 \quad \text{on} \quad \partial\Omega_N. \tag{4.51}$$

The adjoint PDE is linear in terms of the adjoint variable p and needs to be solved backward in time from the terminate condition $p(z,T) = 0$.

Now, we are ready to derive the gradient by taking the variation of the objective functional with respect to m in the direction of $\hat{m} \in \mathcal{M}$:

$$\mathcal{L}^g_m(\psi,p,q,m)(\hat{m}) = \gamma \int_0^T \frac{dm(t)}{dt}\frac{d\hat{m}}{dt}\, dt - \int_0^T \int_{\Omega_F} \hat{m} p\, dsdt. \tag{4.52}$$

Here, p satisfies the adjoint PDE, which depends on the solution to the forward PDE.

Next, we describe how to derive the Hessian-vector product through adjoint methods. We begin with formulating the Lagrangian functional for the Hessian-vector product:

$$\begin{aligned}
&\mathcal{L}^H(\psi, p, m, \tilde{\psi}, \tilde{p}, \tilde{m}) := \gamma \int_0^T \frac{dm(t)}{dt}\frac{d\tilde{m}}{dt}\,dt - \int_0^T \int_{\Omega_F} \tilde{m} p \,ds dt \\
+ &\int_0^T \int_\Omega \frac{\partial \theta(\psi)}{\partial t}\tilde{p} + K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\frac{\partial \tilde{p}}{\partial z}\,dz dt - \int_0^T \int_{\Omega_F} m\tilde{p}\,ds dt \\
+ &\int_\Omega (\psi(z,0) - \psi_0)\theta'(\psi(z,0))\tilde{p}(z,0)\,dz \\
+ &\int_0^T \int_\Omega (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\mathcal{B}\theta'(\psi)\tilde{\psi}\,dz dt \\
+ &\int_0^T \int_\Omega \frac{\partial \theta'(\psi)\tilde{\psi}}{\partial t} p + \left(K'(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi} + K(\psi)\frac{\partial \tilde{\psi}}{\partial z} + K'(\psi)\tilde{\psi}\right)\frac{\partial p}{\partial z}\,dz dt \\
+ &\int_\Omega \tilde{\psi}(z,0)\theta'(\psi(z,0))p(z,0)\,dz = 0.
\end{aligned} \tag{4.53}$$

Here the first and the second terms represent the gradient in the direction of $\hat{m} \in \mathcal{M}$, the forward PDE (the third and forth terms) is enforced by the Lagrange multiplier (incremental adjoint variable) $\tilde{p} \in \hat{\mathcal{V}}$ while the adjoint PDE (the sixth and seventh terms) by the Lagrange multiplier (incremental forward variable) $\tilde{\psi} \in \hat{\mathcal{V}}$, and we replaced q by $\theta'(\psi(z,0))\tilde{p}(z,0)$ for the initial condition of the forward PDE and by $\theta'(\psi)p(z,0)$ for the initial condition of the adjoint PDE, which appears in the last term of the weak form of the adjoint PDE.

To derive the Hessian-vector product, we require the variations of the Lagrangian with respect to its arguments vanish. By requiring the variations with respect to $\tilde{p}$, $\tilde{\psi}$, and $\tilde{m}$ vanish recovers the forward PDE, adjoint PDE, and the gradient, respectively. We require the variation with respect to the adjoint variable p in the direction of $\hat{p} \in \hat{\mathcal{V}}$ vanish:

$$\begin{aligned}
&\mathcal{L}_p^H(u, p, m, \tilde{u}, \tilde{p}, \tilde{m})(\hat{p}) = -\int_0^T \int_{\Omega_F} \tilde{m}\hat{p}\,ds dt \\
+ &\int_0^T \int_\Omega \frac{\partial \theta'(\psi)\tilde{\psi}}{\partial t}\hat{p} + \left(K'(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi} + K(\psi)\frac{\partial \tilde{\psi}}{\partial z} + K'(\psi)\tilde{\psi}\right)\frac{\partial \hat{p}}{\partial z}\,dz dt \\
+ &\int_\Omega \theta'(\psi)\hat{p}(z,0)\tilde{\psi}(z,0)\,dz = 0
\end{aligned} \tag{4.54}$$

for all $\hat{p}$. By removing the spatial derivative of $\hat{p}$ by integration by parts in space and utilizing the fact that $\hat{p}$ is arbitrary, we obtain the following strong form of the

incremental forward PDE:

$$\begin{aligned}
\frac{\partial \theta'(\psi)\tilde{\psi}}{\partial t} - \frac{\partial}{\partial z}\left(K'(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi} + K(\psi)\frac{\partial \tilde{\psi}}{\partial z} + K'(\psi)\tilde{\psi}\right) &= 0, && (4.55)\\
\tilde{\psi}(z,0) &= 0, && (4.56)\\
\tilde{\psi}(z,t) &= 0 \quad \text{on} \quad \partial\Omega_D, && (4.57)\\
K'(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi} + K(\psi)\frac{\partial \tilde{\psi}}{\partial z} + K'(\psi)\tilde{\psi} &= \tilde{m} \quad \text{on} \quad \partial\Omega_N. && (4.58)
\end{aligned}$$

The incremental forward PDE is linear given ψ and $\tilde{m}$. Next, we require the variation with respect to the state variable ψ in the direction of $\hat{\psi} \in \hat{\mathcal{V}}$ vanish:

$$\begin{aligned}
&\mathcal{L}^H_\psi(\psi, p, m, \tilde{\psi}, \tilde{p}, \tilde{m})(\hat{\psi})\\
=\ & \int_0^T \int_\Omega \frac{\partial \theta'(\psi)\hat{\psi}}{\partial t}\tilde{p} + \left(K'(\psi)\hat{\psi}\left(\frac{\partial \psi}{\partial z} + 1\right) + K(\psi)\frac{\partial \hat{\psi}}{\partial z}\right)\frac{\partial \tilde{p}}{\partial z}\, dzdt\\
+\ & \int_\Omega \hat{\psi}(z,0)\theta'(\psi(z,0))\tilde{p}(z,0) + (\psi(z,0) - \psi_0)\theta''(\psi(z,0))\hat{\psi}(z,0)\tilde{p}(z,0)\, dz\\
+\ & \int_0^T \int_\Omega \mathcal{B}\theta'(\psi)\hat{\psi}\mathcal{B}\theta'(\psi)\tilde{\psi} + (\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\mathcal{B}\theta''(\psi)\hat{\psi}\tilde{\psi}\, dzdt\\
+\ & \int_0^T \int_\Omega \frac{\partial \theta''(\psi)\hat{\psi}\tilde{\psi}}{\partial t} p\, dzdt\\
+\ & \int_0^T \int_\Omega \left(K''(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi}\hat{\psi} + K'(\psi)\frac{\partial \hat{\psi}}{\partial z}\tilde{\psi} + K'(\psi)\hat{\psi}\frac{\partial \tilde{\psi}}{\partial z} + K''(\psi)\hat{\psi}\tilde{\psi}\right)\frac{\partial p}{\partial z}\, dzdt\\
=\ & 0 \qquad (4.59)
\end{aligned}$$

for all $\hat{\psi}$. Here, we used $\tilde{\psi}(z,0) = 0$. By removing the spatial and temporal derivative of $\hat{\psi}$ by integration by parts in space and time, respectively, and utilizing the fact that $\hat{\psi}$ is arbitrary, we obtain the following strong form of the incremental adjoint PDE:

$$\begin{aligned}
-\theta'(\psi)\frac{\partial \tilde{p}}{\partial t} &+ K'(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\frac{\partial \tilde{p}}{\partial x} - \frac{\partial}{\partial z}\left(K(\psi)\frac{\partial \tilde{p}}{\partial z}\right)\\
&= -\left(K''(\psi)\frac{\partial \psi}{\partial z}\tilde{\psi} + K'(\psi)\frac{\partial \tilde{\psi}}{\partial z} + K''(\psi)\tilde{\psi}\right)\frac{\partial p}{\partial z} + \frac{\partial}{\partial z}\left(K'(\psi)\tilde{\psi}\frac{\partial p}{\partial z}\right)\\
&+ \theta''(\psi)\tilde{\psi}\frac{\partial p}{\partial t} - \mathcal{B}^*\mathcal{B}(\theta'(\psi))^2\tilde{\psi} - \mathcal{B}^*(\mathcal{B}\theta(\psi) - \theta_{\text{obs}})\theta''(\psi)\tilde{\psi}, && (4.60)\\
\tilde{p}(z,T) &= 0, && (4.61)\\
\tilde{p}(z,t) &= 0 \quad \text{on} \quad \partial\Omega_D, && (4.62)\\
K(\psi)\frac{\partial \tilde{p}}{\partial z} &= -K'(\psi)\tilde{\psi}\frac{\partial p}{\partial z} \quad \text{on} \quad \partial\Omega_N. && (4.63)
\end{aligned}$$

The incremental adjoint PDE is linear with respect to the incremental adjoint variable $\tilde{p}$.

Finally, we are ready to evaluate the Hessian action (Hessian-vector product after discretization). The variation with respect to m in the direction of $\hat{m} \in \mathcal{M}$ yields

$$\mathcal{L}_m^H(\psi, p, m, \tilde{\psi}, \tilde{p}, \tilde{m})(\hat{m}) = \gamma \int_0^T \frac{d\hat{m}}{dt}\frac{d\tilde{m}}{dt}\, dt - \int_0^T \int_{\Omega_F} \hat{m}\tilde{p}\, ds dt. \tag{4.64}$$

In the expression above, only the incremental adjoint $\tilde{p}$ appears, but it depends on all the other variables ψ, p, and $\tilde{\psi}$. Thus, the Hessian-vector product requires solving the forward PDE, the adjoint PDE, and the incremental forward and adjoint PDEs.

4.3.1.4 Discretize-then-Optimize approach

As in the OTD approach, we use the formal Lagrangian method (Chavent, 2010) to derive the gradient and the Hessian-vector product in the DTO approach, as well. We begin with deriving the gradient by formulating the Lagrangian:

$$L^g(\bar{\boldsymbol{\psi}}, \mathbf{m}, \bar{\mathbf{p}}) = J(\mathbf{m}) + \bar{\mathbf{p}}^T \bar{\mathbf{F}}, \tag{4.65}$$

where $\bar{\mathbf{p}} \in \mathbb{R}^{N_s(N_t+1)}$ is the adjoint variable enforcing the constraint by the forward problem $\bar{\mathbf{F}}$. As in the OTD approach, we require the derivatives of the Lagrangian with respect to the arguments vanish. The derivative in terms of the state variable $\bar{\boldsymbol{\psi}}$ is

$$\frac{\partial L^g}{\partial \bar{\boldsymbol{\psi}}} = \frac{\partial D}{\partial \bar{\boldsymbol{\psi}}} + \bar{\mathbf{p}}^T \frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}} = 0, \tag{4.66}$$

where $D = \frac{1}{2}\int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))\, dt$. The detail of $\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}$ and $\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}$ can be found in Sect. B.1.1 and B.1.2. By transposing both sides, we obtain the adjoint equation (also see Sect. B.1.3):

$$\left(\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}\right)^T \bar{\mathbf{p}} = -\left(\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}\right)^T. \tag{4.67}$$

Note that the left-hand side is available from the Newton solver for the forward problem. To derive the gradient, we require the derivative of the Lagrangian with respect to $\mathbf{m}$ vanish:

$$\frac{\partial L^g}{\partial \mathbf{m}} = \gamma \mathbf{m}^T \mathbf{K}^m + \bar{\mathbf{p}}^T \frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{m}} = \mathbf{0}, \tag{4.68}$$

where $\frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{m}}$ is derived in Sect. B.1.4. Because the gradient is the transpose of the derivative, the gradient $\mathbf{g}$ is

$$\mathbf{g} = \gamma \mathbf{K}^m \mathbf{m} + \left(\frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{m}}\right)^T \bar{\mathbf{p}}. \tag{4.69}$$

Next, we formulate the Lagrangian for the Hessian-vector product (i.e., minimizing the directional derivative in the direction of $\tilde{\mathbf{m}}$ given the forward and adjoint problems are satisfied):

$$L^H(\bar{\boldsymbol{\psi}}, \mathbf{m}, \bar{\mathbf{p}}; \bar{\tilde{\boldsymbol{\psi}}}, \tilde{\mathbf{m}}, \bar{\tilde{\mathbf{p}}}) = \tilde{\mathbf{m}}^T(\gamma \mathbf{K}^m \mathbf{m} + \mathbf{C}^T \bar{\mathbf{p}}) + \bar{\tilde{\mathbf{p}}}^T \bar{\mathbf{F}} + \bar{\tilde{\boldsymbol{\psi}}}^T (\mathbf{A}^T \bar{\mathbf{p}} + \mathbf{d}), \tag{4.70}$$

where $\bar{\tilde{\boldsymbol{\psi}}} \in \mathbb{R}^{N_s(N_t+1)}$ and $\bar{\tilde{\mathbf{p}}} \in \mathbb{R}^{N_s(N_t+1)}$ are incremental forward and adjoint variables, respectively, $\tilde{\mathbf{m}}$ is the vector to be applied to the Hessian, $\mathbf{C} := \frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{m}}$, $\mathbf{A} := \frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}$, and $\mathbf{d} := \left(\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}\right)^T$.

By requiring the derivative of the Lagrangian with respect to the adjoint variable vanish, we obtain

$$\frac{\partial L^H}{\partial \bar{\mathbf{p}}} = \tilde{\mathbf{m}}^T \mathbf{C}^T + \bar{\tilde{\boldsymbol{\psi}}}^T \mathbf{A}^T = \mathbf{0}. \tag{4.71}$$

Transposing the equation above results in the incremental forward problem (also see Sect. B.1.5):

$$\mathbf{A}\bar{\tilde{\boldsymbol{\psi}}} = -\mathbf{C}\tilde{\mathbf{m}}. \tag{4.72}$$

There are no new derivations required for the incremental forward problem because both $\mathbf{A}$ and $\mathbf{C}$ are available.

Next, we require the derivative of the Lagrangian with respect to the state variable vanish, we obtain

$$\frac{\partial L^H}{\partial \bar{\boldsymbol{\psi}}} = \tilde{\mathbf{m}}^T \frac{\partial \mathbf{C}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}} + \bar{\tilde{\mathbf{p}}}^T \mathbf{A} + \bar{\tilde{\boldsymbol{\psi}}}^T \frac{\partial \mathbf{A}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}} + \bar{\tilde{\boldsymbol{\psi}}}^T \frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}} = \mathbf{0}. \tag{4.73}$$

The detail of $\frac{\partial \mathbf{A}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}$ and $\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}$ are described in Sect. B.1.6 and B.1.7, respectively. By using the fact that $\frac{\partial \mathbf{C}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}} = \mathbf{0}$ and transposing the equation above, we obtain the incremental adjoint equation (also see Sect. B.1.8):

$$\mathbf{A}^T \bar{\tilde{\mathbf{p}}} = -\left(\left(\frac{\partial \mathbf{A}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)^T + \left(\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}\right)^T\right) \bar{\tilde{\boldsymbol{\psi}}}. \tag{4.74}$$

Note that the left hand side is the same as the for the adjoint problem.

Finally, the derivative of the Lagrangian with respect to $\mathbf{m}$ gives

$$\frac{\partial L^H}{\partial \mathbf{m}} = \gamma \tilde{\mathbf{m}}^T \mathbf{K}^m + \bar{\tilde{\mathbf{p}}}^T \mathbf{C}, \tag{4.75}$$

where we used the fact that $\frac{\partial \mathbf{C}^T \bar{\mathbf{p}}}{\partial \mathbf{m}} = \mathbf{0}$ and $\frac{\partial \mathbf{A}^T \bar{\mathbf{p}}}{\partial \mathbf{m}} = \mathbf{0}$. The transpose of the equation avobe is the gradient of the directional derivative in the direction of $\tilde{\mathbf{m}}$, which is the Hessian-vector product

$$\mathbf{H}\tilde{\mathbf{m}} = \gamma \mathbf{K}^m \tilde{\mathbf{m}} + \mathbf{C}^T \bar{\tilde{\mathbf{p}}}. \tag{4.76}$$

It is clear that we need the incremental adjoint variable $\bar{\tilde{\mathbf{p}}}$ to compute the Hessian-vector product, and the rest of the expressions are the same as the gradient (Eq. 4.69).

4.3.2 Physics-informed neural networks

Physics-informed neural networks (PINNs) use neural networks to approximate the solution to PDEs based on the universal approximation capability of NNs (Cybenko, 1989; Hornik, 1991). In the last few years, many studies have been conducted to understand how PINNs work for different types of PDEs. So far, it is known that the original PINNs framework proposed by Raissi et al. (2019) does not always work well for the forward modeling of PDEs (e.g., Fuks and Tchelepi (2020)), and modified PINN frameworks were proposed and tested (Wang et al., 2022; Patel et al., 2022). As for inverse modeling, PINNs have shown promising results in various fields Raissi et al. (2019); Jin et al. (2021); Bandai and Ghezzehei (2022). Our previous studies demonstrated that PINNs could be used to estimate surface water flux from soil moisture measurements (Bandai and Ghezzehei, 2021, 2022). In this study, we compare PINNs with adjoint methods to understand the capabilities and limitations of both methods.

Our implementation of PINNs is presented in Bandai and Ghezzehei (2022) with slight modifications to enable the comparison between PINNs and adjoint methods. In the PINN framework, we aim to approximate the solution to the RRE $\psi(z,t)$ by a feedforward neural network $\mathcal{N}(z,t)$. Here, we begin with the mathematical formulation of feedforward NNs with L hidden layers with layer-wise locally adaptive activation function (L-LAAFs) (Jagtap et al., 2020). NNs transform the input vector $\mathbf{x} \in \mathbb{R}^{n^x}$ into the output vector

$$\hat{\mathbf{y}} := \mathcal{N}(\mathbf{x}), \tag{4.77}$$

where the hat represents prediction in this section, while it means test functions in adjoint methods. The input vector $\mathbf{x}$ is transformed by the composition of affine transformation and non-linear activation functions in the following way:

$$\begin{aligned}
\mathbf{h}^{[1]} &:= \sigma(sa^{[1]}(\mathbf{W}^{[1]}\mathbf{x} + \mathbf{b}^{[1]})),\\
\mathbf{h}^{[2]} &:= \sigma(sa^{[2]}(\mathbf{W}^{[2]}\mathbf{h}^{[1]} + \mathbf{b}^{[2]})),\\
&\vdots\\
\mathbf{h}^{[L-1]} &:= \sigma(sa^{[L-1]}(\mathbf{W}^{[L-1]}\mathbf{h}^{[L-2]} + \mathbf{b}^{[L-1]})),\\
\mathbf{h}^{[L]} &:= \sigma(sa^{[L]}(\mathbf{W}^{[L]}\mathbf{h}^{[L-1]} + \mathbf{b}^{[L]})),
\end{aligned} \tag{4.78}$$

where $\mathbf{h}^{[k]} \in \mathbb{R}^{n^{[k]}}$ for $k = 1, ..., L$ is the vector corresponding to the kth hidden layer with $n^{[k]}$ units, $\mathbf{W}^{[k]}$ and $\mathbf{b}^{[k]}$ are weight matrices and bias vectors, respectively, for the kth hidden layer, $s \geq 0$ is a fixed scaling parameter set to 20, $a^{[k]}$ is a tuning parameter changing the shape of the element-wise activation function σ for the kth hidden layer. In this study, we used the hyperbolic tangent function (tanh) for the activation function. Then, the output vector $\hat{\mathbf{y}}$ is computed as

$$\hat{\mathbf{y}} := o(\mathbf{W}^{[L+1]}\mathbf{h}^{[L+1]} + \mathbf{b}^{[L+1]}), \tag{4.79}$$

where o is the output function, $\mathbf{W}^{[L+1]}$ and $\mathbf{b}^{[L+1]}$ are the weight matrix and bias vector for the output layer. We assemble all the weight matrices and bias vectors

as $\mathbf{W} := \{\mathbf{W}^{[1]}, ..., \mathbf{W}^{[L+1]}\}$ and $\mathbf{b} := \{\mathbf{b}^{[1]}, ..., \mathbf{b}^{[L+1]}\}$, and the slope parameters as $\mathbf{a} := \{a^{[1]}, ..., a^{[L]}\}$. Then, the NN parameters we need to estimate are summarized as $\Theta := \{\mathbf{W}, \mathbf{b}, \mathbf{a}\}$.

Because the water potential is negative for unsaturated soils, we used the identity function for the output function o and the following transformation on the output of NNs:

$$\psi(z,t) \approx \hat{\psi}(z,t) = -\exp(\mathcal{N}(t,z;\Theta)) + \beta, \tag{4.80}$$

where β is a fixed parameter that allows the water potential to be positive if we wish. In this study, we set $\beta = 1$. To construct PINNs, we define the residual of the PDE as

$$\hat{r}(z,t;\Theta) := \frac{\partial \hat{\theta}}{\partial t} - \frac{\partial}{\partial z}\left[\hat{K}\left(\frac{\partial \hat{\psi}}{\partial z} + 1\right)\right], \tag{4.81}$$

where $\hat{\theta}$ and $\hat{K}$ are computed from $\hat{\psi}$ with given WRCs and HCFs. To estimate the NN parameters Θ, we minimize the objective function

$$\mathcal{L}(\Theta) := \mathcal{L}_{\text{data}} + \gamma \mathcal{L}_r, \tag{4.82}$$

where γ is the regularization parameter, and data-misfit term $\mathcal{L}_{\text{data}}$ and regularization term $\mathcal{L}_r$ are defined as

$$\mathcal{L}_{\text{data}}(\Theta) := \sum_{i=1}^{N_{\text{data}}} [\hat{\theta}(z^i_{\text{data}}, t^i_{\text{data}}) - \theta^i_{\text{data}}]^2, \tag{4.83}$$

$$\mathcal{L}_r(\Theta) := \sum_{i=1}^{N_r} [\hat{r}(z^i_r, t^i_r)]^2, \tag{4.84}$$

where $\{\theta^i_{\text{data}}, z^i_{\text{data}}, t^i_{\text{data}}\}_{i=1}^{N_{\text{data}}}$ denotes the N_{data} data points, $\{z^i_r, t^i_r\}_{i=1}^{N_r}$ denotes the N_r residual points (also called collocation points) at which the residual of the PDE is evaluated. The residual points were randomly selected from the whole spatial and temporal domains. In the study, the initial condition and the Dirichlet boundary condition at the lower boundary were incorporated as data points. Thus, N_{obs} in the PINN framework here is not the same as N_{data} in the adjoint method. To evaluate the regularization term $\mathcal{L}_r$, we need to compute the partial derivatives in the residual. In the PINN framework, this is conducted by automatic differentiation (Baydin et al., 2018).

The objective function was minimized by the 10^5 iterations Adam optimizer (Kingma and Ba, 2014) followed by the L-BFGS-B optimizer (Byrd et al., 1995). The exponential decay of the learning rate was used for the Adam optimizer with the initial learning rate of 0.001, the decay rate of 0.9, and the decay step of 1000. The other parameters were set to their default values implemented in TensorFlow (Abadi et al., 2015). Only 128 of the randomly selected residual points are used for each iteration in the Adam optimizer, while all the data points are considered. The L-BFGS-B optimizer follows the Adam optimizer and is terminated once the loss function converges with prescribed thresholds. We used the L-BFGS-B optimizer implemented in

Scipy (Virtanen et al., 2020) with the following parameters: maxcor = 50, maxls = 50, maxiter = 50000, maxfun = 50000, ftol = 1.0×10^{-10}, gtol = 1.0×10^{-8}, and the default values for the other parameters.

After training the NNs, the surface water flux is evaluated by the Buckingham Darcy law (Eq. 4.12), where the gradient of the water potential is evaluated by automatic differentiation. This may not seem comparable to the implementation of adjoint methods, where the surface water flux is approximated by a finite element method. However, this implementation most exploits the flexibility of PINNs and is particularly advantageous for complex geometries. In the numerical examples, we discuss the consequences of the difference in the surface water flux evaluation between the methods.

4.3.3 Comparison between adjoint method and physics-informed neural networks

Before we discuss numerical examples, we discuss the comparison between the adjoint method and PINNs, which is summarized in Table 4.2. The adjoint method uses linear functions (or possibly higher-order elements) to approximate the solution to PDEs if the PDEs are discretized by finite element methods. The finite element coefficients are obtained by solving the system of non-linear equations resulting from the discretization of the PDEs. On the other hand, PINNs use NNs to approximate the solution, and the NN parameters are obtained by solving a minimization problem. The way to obtain the solution to PDEs for each method is directly related to whether the physics the PDEs describe is correctly imposed. As for the adjoint method, the physics is correctly embedded in the solution as long as the discretization of the PDEs is appropriate because the system of the non-linear equations is solved accurately. This is not always the case for PINNs because the minimization problem is high-dimensional, non-linear, and non-convex, and thus accurately solving the problem is very difficult, and there is no guarantee that the obtained solution satisfies the physical requirements (Patel et al., 2022). As we demonstrate in the numerical examples, PINN's solution is not necessarily correct if the amount of data is limited.

We need to deal with the noise in the data to prevent the noise from severely damaging the solution to the inverse problem. In the adjoint method, a regularization term is added to the objective function to penalize highly oscillatory components in the solution. Adding the regularization term plays a role in smoothing the solution and helping the Newton method to converge faster because it damps small eigenvalues of the Hessian. In contrast, there is no explicit strategy to deal with noise in the data in the PINN framework, while the PDE residual is added as a "regularization term" in the objective function (Eq. 4.82). By comparing it with the Lagrangian function for the adjoint method, we may interpret the regularization parameter γ in PINNs as the adjoint variable in the adjoint method. In this perspective, the regularization parameter γ should be a function of space and time, not a constant, and becomes smaller as the data-misfit term becomes smaller because the solution to

the adjoint PDE becomes small when the data-misfit term is small (see Eq. 4.48). The regularization term γ in the PINN framework has been studied, and several methods have been proposed and tested. It would be interesting to pursue the relationship with the adjoint variable in future studies.

Regarding the method to minimize the objective function, the adjoint method uses the Newton method with the gradient and the Hessian-vector product obtained by solving the adjoint PDE and the incremental forward and adjoint PDEs. On the other hand, PINNs use a stochastic gradient method or a Quasi-Newton method, where only the gradient computed by automatic differentiation is used. We are unaware of studies training PINNs with the Hessian information (not the Hessian approximation by gradient, such as the L-BFGS-B method), while the Newton method on NNs has been studied (Dauphin et al., 2014).

For time-dependent problems, the adjoint problem can have memory storage limitations because the Hessian-vector product requires the whole solutions for the forward and adjoint problems and the incremental forward and adjoint problems, which may be intractable for large-scale problems. In that case, we use a checkpointing method Ghattas and Willcox (2021) to trade additional computational burdens for memory storage. On the other hand, PINNs do not have such memory storage limitations for time-dependent problems. However, it is known that as the scale becomes larger, it is more challenging to train PINNs. To overcome this challenge, Wight and Zhao (2021) proposed and tested sequential training. We do not implement the checkingpoint method or the sequential training but discuss the limitations of the adjoint method and PINNs for large-scale time-dependent problems in a numerical example.

4.4 Numerical examples

In this section, we outline three numerical examples to assess the performance of the adjoint method and PINNs to estimate surface water flux from soil moisture measurements. In the three cases, we assumed that soil moisture dynamics is vertical and described by the RRE (Eq. 4.1) and the initial and boundary conditions, and the BC parameters for Sandy Loam A in Table 4.1 were used for the WRC and the HCF. For all the numerical experiments, we used synthetic measurements generated by solving the forward problem for each example. To account for measurement and model errors, we added Gaussian noise with a mean of zero and a standard deviation of 0.005 to the numerical solution. The number of measurement locations varies from five locations for Examples 1 and 2 to one for Example 3. The measurement data at the locations were extracted from the noisy synthetic data for the inverse modeling.

The performance of the methods were assessed by the accuracy of the reconstructed forward solution θ and the estimated inverse solution m by computing the

Table 4.2: Comparison between an adjoint method and physics-informed neural networks (PINNs).

	Adjoint method	PINNs
Basis functions	Linear functions	Neural networks (NNs)
Parameter for forward problem	Finite element coefficients	NN parameters
Physics	Imposed correctly	Not necessarily imposed accurately
Strategy for stability against noise in data	Regularization	Not explicit
Method to minimize objective function	Inexact Newton-Conjugate Gradient method	Stochastic gradient method and quasi-Newton method
Parameters	Finite element coefficients	NN parameters
Gradient Computation	Solving adjoint problem	Automatic differentiation
Hessian information	Yes	No, but possible with automatic differentiation
Limitation for time-dependent problems	Memory storage limitation	Difficulty in training NNs

L_2 relative error ϵ for each variable:

$$\epsilon^{\delta} := \left(\frac{\frac{1}{N}\sum_i^N (\hat{\delta}^i - \delta^i_{\text{true}})^2}{\frac{1}{N}\sum_i^N (\delta^i_{\text{true}})^2} \right)^{1/2} \tag{4.85}$$

for $\delta = \theta, m$, where $\hat{\delta}$ and δ_{true} are predicated and true values for each variable evaluated at all the node points.

The Adjoint method was using FEniCS (Logg et al., 2012), which is a finite element package implemented in C++ with a Python interface. To speed up the computation, we implemented the BC model in C++, and the rest was written in Python. We used the conjugate gradient method implemented in hiPPYlib (Villa et al., 2020) to solve the Newton system (Eq. 4.39). The Newton method used in the forward modeling does not necessarily converge well when the upper boundary condition imposes too strong evaporation from the soil. This happened at the first Newton iteration in the inverse modeling. Therefore, we stopped the forward modeling when we detected such behaviors from the number of Newton iterations for the forward modeling.

PINNs were implemented in Python using TensorFlow (Abadi et al., 2015). The weight parameters $\mathbf{W}$ were initialized by the Xavier initialization (Glorot and Bengio, 2010), while the bias parameters $\mathbf{b}$ and the slope parameters $\mathbf{a}$ were initialized to 0 and 0.05, respectively. Three different weight initializations were conducted to account for the effect of the weight initialization for each example. For all the numerical examples, we used NNs with five hidden layers with fifty units. Thus, the number of NN parameters was 10406. The computational time presented in the study was obtained with Intel(R) Core(TM) i7-9700K CPU 3.60GHz. In the next section, we describe the details and the purpose of each model problem.

4.4.1 Example 1

The first example is carried out to verify the implementation of the adjoint method (gradient and the Hessian-vector product evaluations) and provides a baseline for the performance of each method. The length of the soil Z was set to 100 cm, and the final time T was 10 hours. A uniform initial condition $\psi_0 = -5000$ cm for $z \in \Omega$ and a constant Dirichlet boundary condition $\psi_D = -5000$ cm were used. The surface water flux boundary condition m was set to $m = 2.0\sin(t\pi/T)$, as shown in Fig. 4.7. As for the finite element method, a constant spatial mesh and time-stepping were used, and N_s and N_t were set to 201 and 200, respectively. Thus, the number of finite element coefficients for the forward solution and the time-dependent boundary condition was 40401 and 201, respectively. The initial guess of the inexact Newton-CG method was $m_0 = 1.0\sin(t\pi/T)$. In PINNs, 10000 randomly selected residual points were used to train NNs. The volumetric water content measured at five locations $z = -10, -20, -30, -40, -50$ cm were used as measurement data.

4.4.2 Example 2

In this example, we investigated the ability of the two methods to recover a high-frequency component of the upper boundary condition. To achieve that, we set the surface water flux as $m = 2.0\sin(t\pi/T) + 0.2\sin(t\pi)$, as shown in Fig. 4.16. The other settings were the same as Example 1.

4.4.3 Example 3

In the last example, we considered soil moisture dynamics for a more extended period to study the robustness of the two methods. The final time T was set to 100 hours, and the length of the soil Z was set to 300 cm to prevent the water from reaching the bottom of the soil. The initial and the lower boundary condition were the same as Example 1. We set the surface water flux condition as Fig. 4.25, where three rainfall events and evaporation between the rainfalls were simulated. The rainfall rate for each event was set to 1 cm hours^{-1} for $0 < t < 10$, $2.0 \times \sin(t - 20)\pi/T$ cm hours^{-1} for $20 < t < 30$, $2.0\times\sin((t-50)\pi/T)+0.2\times\sin((t-50)\pi)$ cm hours^{-1} for $50 < t < 60$, and 1.0 cm hours^{-1} for $80 < t < 90$, the evaporation rate was 0.2 cm hours^{-1}.As for the finite element method, a constant spatial mesh and time-stepping were used, and N_s and N_t were set to 601 and 2000, respectively. Thus, the number of finite element coefficients was 1202601. In PINNs, 100000 randomly selected residual points were used to train NNs. The volumetric water content measured at $z = -5$ cm was used, which can be considered as soil moisture data from satellite remote sensing data.

4.5 Results and discussions

4.5.1 Example 1

First, the implementation of the gradient and the Hessian-vector product in the adjoint method was verified by comparing them with a finite difference method. The gradient and the Hessian-vector product were evaluated at $m = 1.0\sin(t\pi/T)$, and the regularization parameter γ was set to 10^{-4}. To compare them with a finite difference approximation, we computed the directional derivative (Gradient action) in the direction of a randomly selected vector for the gradient, and a similar approach was used for the Hessian action. Fig. 4.2 (**a**) demonstrates that the error between the gradient action derived from the adjoint method and a finite difference method converged to zero with a first order until numerical errors are not negligible. A similar trend was observed for the Hessian-vector product, as in Fig. 4.2 (**b**). These finite difference tests validated our implementation of the gradient and the Hessian-vector product via the adjoint method.

Fig. 4.3 (**a**) and (**b**) showed the best-fitted volumetric water content θ for the adjoint method and PINNs, respectively, with the true solution represented by the markers. The regularization parameter γ was set to 10^{-4} and 0.05 for the adjoint

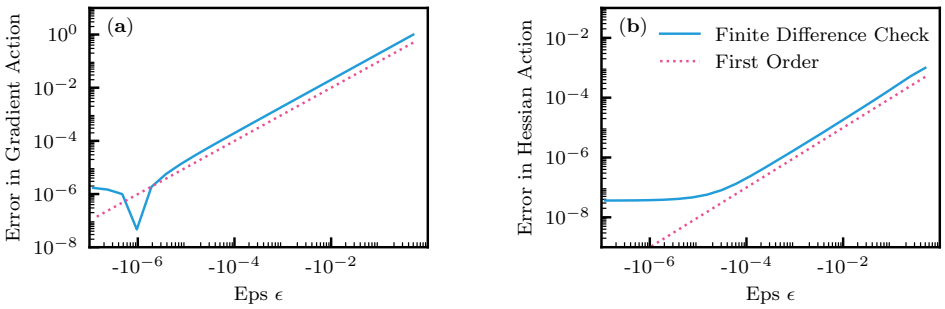

Figure 4.2: Example 1: The gradient (**a**) and the Hessian-vector product (**b**) were compared with finite difference approximations. The x-axis (Eps ϵ) represents the step size of the finite difference approximation.

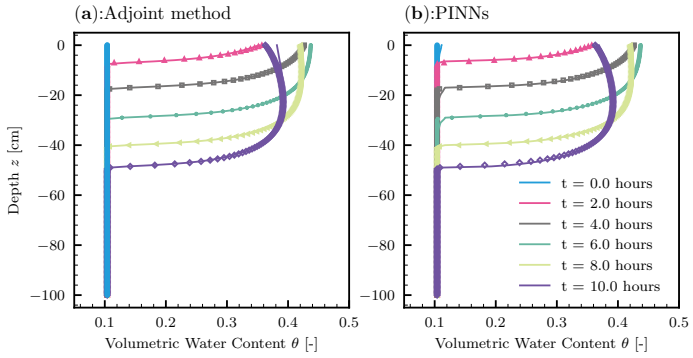

Figure 4.3: Example 1: The best-fitted solution to the forward problem (solid lines) against the true solution (markers) for the adjoint method (**a**) and PINNs (**b**) from data at five measurement locations $z = -10, -20, -30, -40, -50$ cm.

method and PINNs, respectively. The effect of the regularization parameter is discussed later. The relative L_2 error ϵ^{θ} was 0.0156 and 0.0164 for the adjoint method and PINNs, respectively. Fig. 4.4 and Fig. 4.5 demonstrate more detailed distributions of the error. For the adjoint method, larger errors were observed near the surface of the soil near the initial condition $t = 0$ because the estimation of the upper boundary condition was not accurate due to the fact that the water did not reach the first measurement location at $z = -10$ cm for a small t. In PINNs, larger errors were observed at the tip of wetting fronts, which is evident in Fig. 4.5 and Fig. 4.6. For Example 1, both methods were able to reconstruct the solution to the forward problem with accuracy because the number of measurement locations was sufficient.

The solution to the inverse problem m with the regularization parameter $\gamma = 1.0 \times 10^{-4}$ for the adjoint and $\gamma = 0.05$ for PINNs is shown in Fig. 4.7. The L_2 error

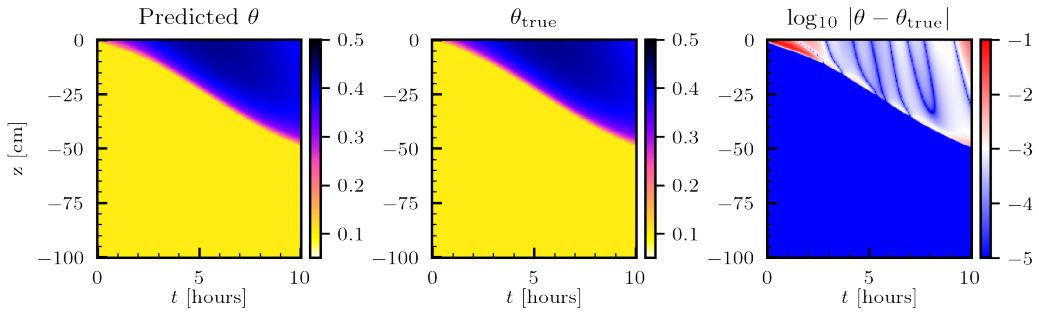

Figure 4.4: Example 1: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for the adjoint method. A small number 10^{-9} was added to the error for the visualization.

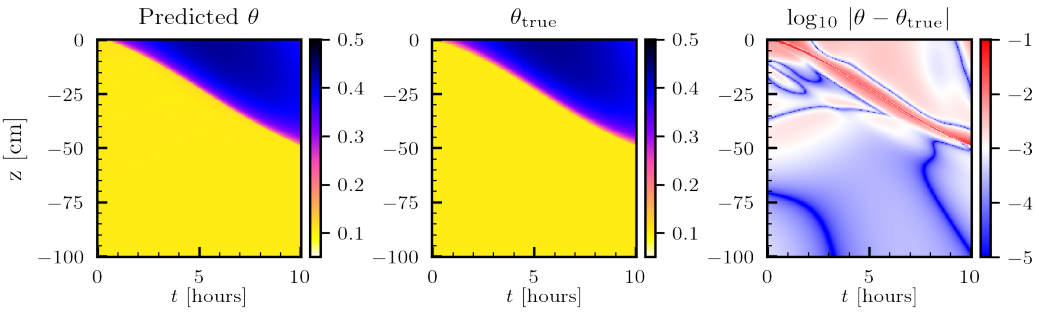

Figure 4.5: Example 1: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for PINNs.

ϵ^m was 0.065 for both methods. Both estimated surface water fluxes were satisfactory except for the boundaries. The adjoint method overestimated the surface water flux at both boundaries. This is because of the limited number of observation points and the boundary condition of the parameter space $\mathcal{M}$. On the other hand, PINNs' surface water flux was close to zero for both boundaries. This resulted from the estimated uniform distribution of the water potential near the surface at near $t = 0$ and $t = 10$. Nevertheless, there are errors for small t for PINNs as well due to the fact that water did not reach the first observational point $z = -10$ cm.

Next, we discuss the performance of the minimization algorithm for each method. Table 4.3 shows the result of the Newton-CG method for the adjoint method. At the initial Newton iteration, we needed many line searches because of the convergence issues of the forward solver when the surface water flux m imposed very strong evaporation. The total CG iterations were 84, and it took 390 seconds. As for PINNs, the evolution of the loss terms is shown in Fig. 4.8 (**a**) and (**b**) for the Adam and the L-BFGS-B optimizers, respectively. Note that the loss term for the residual $\mathcal{L}_r$ for the Adam optimizer is not identical to the one for the L-BFGS-B optimizer because the

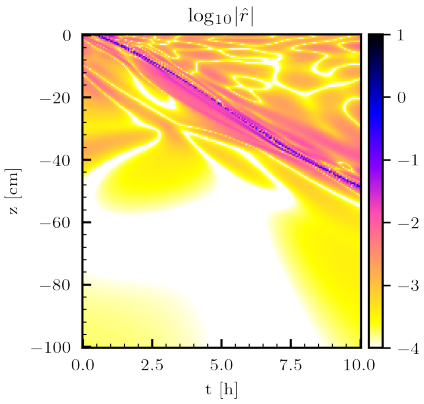

Figure 4.6: Example 1: The residual of the PDE on log-scale for PINNs.

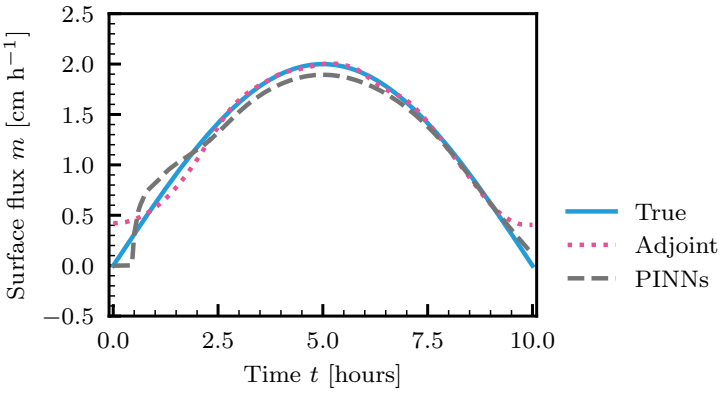

Figure 4.7: Example 1. The true (blue solid line) and the estimated surface water flux from data at five measurement locations $z = -10, -20, 30, -40, -50$ cm for the adjoint method (dotted pink) and PINNs (dotted grey).

former only used 128 residual points while the latter used the whole residual points. It took 828 and 663 seconds for each optimizer. For Example 1, the adjoint method was faster than PINNs.

Next, we investigated the effect of the regularization parameter γ for both methods by plotting a so-called L-curve (Hansen, 2000). The L-curve for the adjoint method is shown in Fig. 4.9 (**a**), where the square of the regularization term $\sqrt{\mathbf{m}^T\mathbf{K}^m\mathbf{m}}$ in Eq. 4.38 was plotted against the square of the data-misfit term $\int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))^T(\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))\,dt$ for different regularization parameter $\gamma \in \{10^{-6}, 5\times 10^{-6}, 10^{-5}, 5^{-5}, 10^{-4}, 5\times 10^{-4}, 10^{-3}, 5\times 10^{-3}, 10^{-2}, 5\times 10^{-2}\}$. A common strategy is to choose γ at which the curvature of the L-curve is maximum (Hansen, 2000). In this

Table 4.3: Example 1: The performance of the Newton-Conjugate gradient (CG) method.

Newton iteration	CG iterations	Line searches	Cost	Norm of the gradient
1	1	13	4.46×10^{-2}	2.08×10^{-1}
2	1	0	2.98×10^{-2}	1.11×10^{-1}
3	1	0	2.08×10^{-2}	7.40×10^{-2}
4	2	0	7.24×10^{-3}	3.86×10^{-2}
5	1	0	5.26×10^{-3}	4.25×10^{-2}
6	2	0	1.35×10^{-3}	1.99×10^{-2}
7	1	0	1.19×10^{-3}	1.10×10^{-2}
8	3	0	7.68×10^{-4}	3.20×10^{-3}
9	9	0	7.26×10^{-4}	7.28×10^{-4}
10	17	0	7.22×10^{-4}	2.87×10^{-4}
11	15	0	7.22×10^{-4}	6.02×10^{-6}
12	31	0	7.22×10^{-4}	4.62×10^{-8}

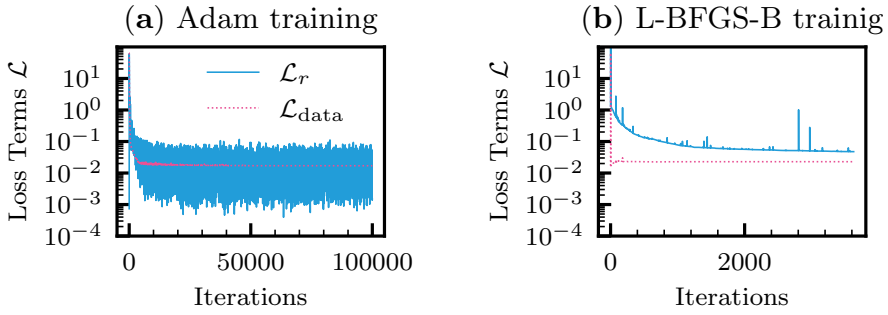

Figure 4.8: Example 1: The evoluion of the loss terms for the Adam optimizer (**a**) and the L-BFGS-B optimizer (**b**).

study, however, we chose the regularization parameter γ based on the accuracy of the estimated solution m because we observed the regularization parameter γ suggested by the L-curve gave too-smoothed solutions (see 4.10). For Example 1, we chose $\gamma = 1.0 \times 10^{-4}$ for the adjoint method, which is close to the point of the maximum curvature of the L-curve, while we see more deviations in the next two examples. As for PINNs, there are no available theories regarding how to appropriately choose the regularization parameter γ. For comparison, we show the L-curve for PINNs in Fig.4.9 (**b**), where the regularization term $\sqrt{\mathcal{L}_r}$ was plotted against the data-misfit term $\sqrt{\mathcal{L}_{\text{data}}}$ for different $\gamma \in \{10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 0.1, 0.5, 1, 5, 10\}$. Because we used three different initializations of NNs, there are three data points for each regularization parameter γ, which is shown by different colors (we observed similar results from different initializations). We observed a linear relationship on the log-log scale

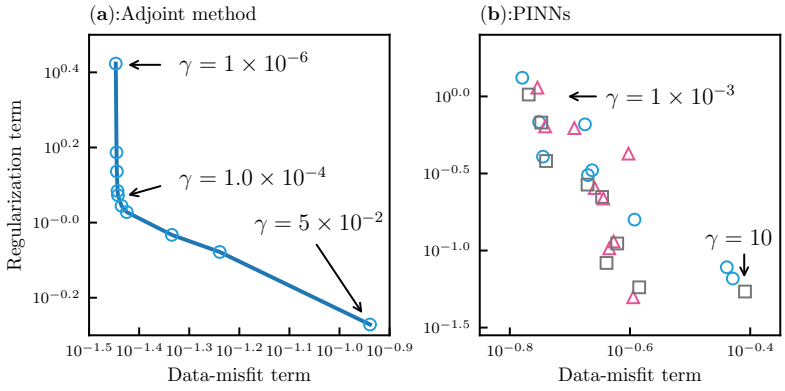

Figure 4.9: Example 1: The L-curve for the adjoint method (**a**) and physics-informed neural networks (PINNs) (**b**). The different colors in (**b**) represent results from different initialization of NNs.

with some outliers for high γ. In such high γ, we observed a too-smoothed solution m, which is similar to the too-smoothed solution obtained by the adjoint method. We selected $\gamma = 0.05$ based on the best-reconstructed inverse solution. Fig. 4.10 demonstrates the effect of the regularization parameter γ on the inverted solution m for the adjoint method. It is clear that the inverted solution was severely affected by the noise in the data when the regularization parameter γ was very low. On the other hand, if the regularization parameter γ was too high, the estimated surface flux deviated from the true one. Similar results were obtained for PINNs, as shown in Fig. 4.11. A large γ leaded to a wrong solution (e.g., $\gamma = 10$). However, the effect of the noise in the data appeared to be smaller than that of the adjoint method, regardless of no explicit regularization in the PINN framework. This may be related to NNs' property, the so-called spectral bias, where NNs tend to converge to a smoothed solution (Wang et al., 2022).

4.5.2 Example 2

In this example, we investigated whether both methods could recover a high-frequency component of the upper boundary condition. Fig. 4.12 showed the best-fitted θ for the adjoint method (**a**) and the PINNs (**b**), respectively. The regularization parameter γ was 10^{-5} and 0.01 for the adjoint method and PINNs, respectively. The relative L_2 error ϵ^θ was 0.00500 and 0.0281 for the adjoint method and PINNs, respectively. Fig. 4.13 and Fig. 4.14 demonstrate more detailed distributions of the error.

The estimated upper boundary condition m is shown for each method in Fig. 4.16. The L_2 error ϵ^m was 0.0497 and 0.0950 for the adjoint and PINNs, respectively, demonstrating that the adjoint method was better in terms of estimating the high-frequency component of the upper boundary condition than PINNs for this example.

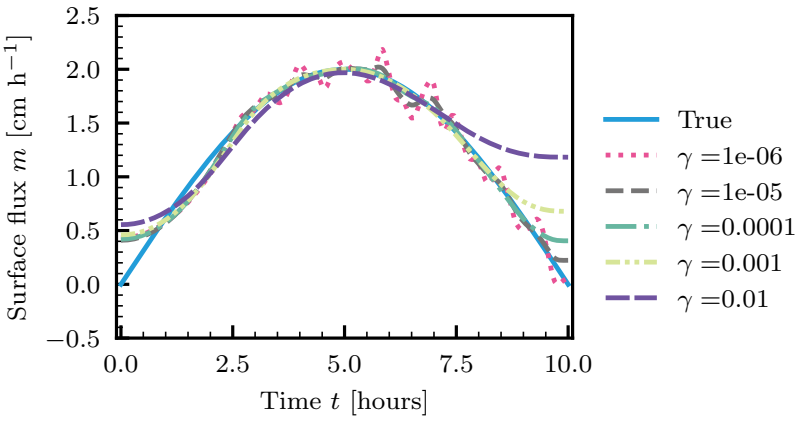

Figure 4.10: Example 1: The effect of the regularization parameter γ on the estimated solution m for the adjoint method.

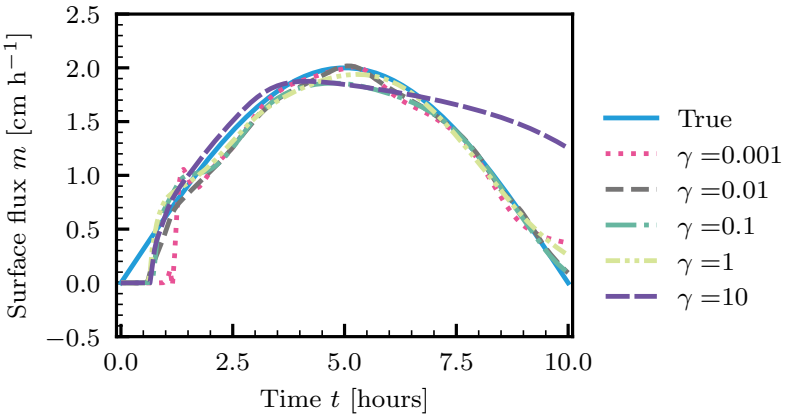

Figure 4.11: Example 1: The effect of the regularization parameter γ on the estimated solution m for PINNs.

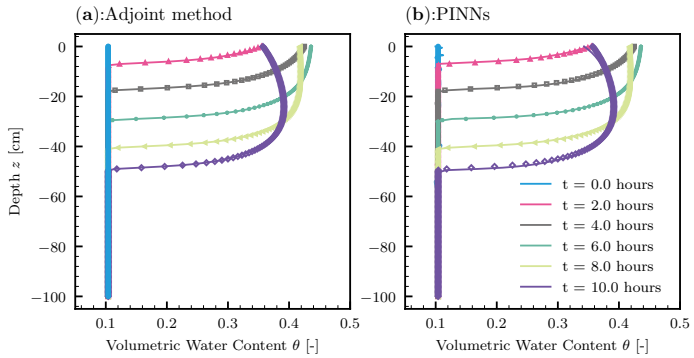

Figure 4.12: Example 2: The best-fitted solution to the forward problem (solid lines) against the true solution (markers) for the adjoint method (**a**) and PINNs (**b**) from data at five measurement locations $z = -10, -20, -30, -40, -50$ cm.

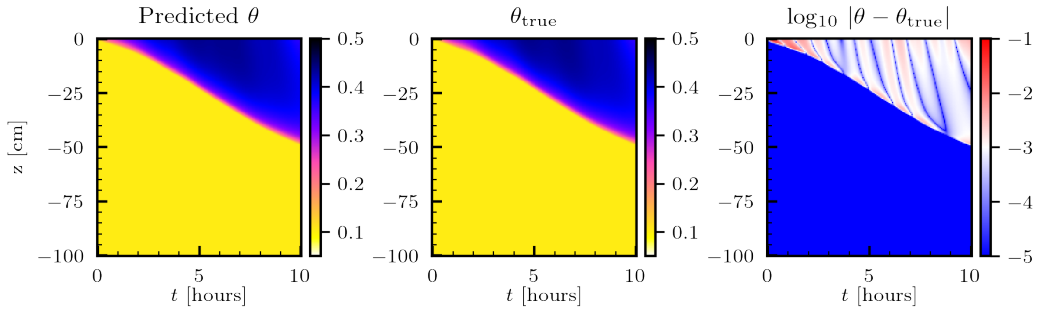

Figure 4.13: Example 2: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for the adjoint method. A small number 10^{-9} was added to the error for the visualization.

Next, we discuss the performance of the minimization algorithms. Table 4.4 shows the result for the Newton-CG method for the adjoint method. The total CG iteration was 206, and it took 779 seconds. Compared to Example 1, the number of CG iterations increased. This was because of the small eigenvalues of the Hessian due to the lower γ used in Example 2. As for PINNs, the evolution of the loss terms is shown in Fig. 4.17 (**a**) and (**b**) for the Adam and the L-BFGS-B optimizers, respectively. Regarding computational time, it took 898 and 1651 seconds for the Adam and the L-BFGS-B optimizers, respectively. Compared to Example 1, more iterations of the L-BFGS-B optimizer were needed for the convergence.

Fig. 4.18 (**a**) and (**b**) shows the L-curve for the adjoint method and PINNs, respectively. For the adjoint method, the regularization parameters were $\gamma \in \{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5^{-5}, 7.5 \times 10^{-5}, 10^{-4}, 2.5 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$,

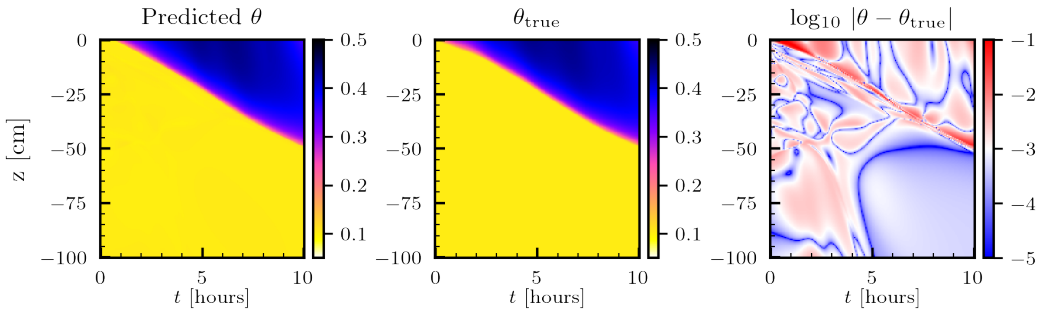

Figure 4.14: Example 2: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for PINNs.

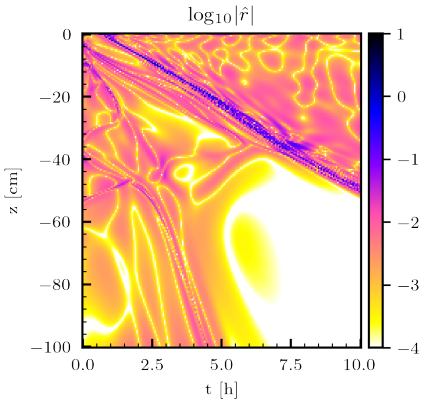

Figure 4.15: Example 2: The residual of the PDE on log-scale for PINNs.

while $\gamma \in \{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1, 5, 10\}$ were used for PINNs. For the adjoint method, the best result was obtained with $\gamma = 10^{-5}$, which is far from at which the curvature of the L-curve is maximum. For PINNs, $\gamma = 0.01$ gave the best result. Fig 4.19 shows the effect of the regularization parameter γ on the inverted solution m for the adjoint method. When the regularization parameter was $\gamma = 10^{-6}$, the inverted solution was severely affected by the noise in the data. On the other hand, when the regularization parameter γ was larger than 10^{-4}, the high-frequency component was not well recovered. Considering the fact that we could not estimate the best γ based on the L-curve (Fig. 4.18), it would be challenging to extract such a high-frequency component of the upper boundary condition in real applications. Similar results were obtained for PINNs, as shown in Fig. 4.20. As in the adjoint method, we could extract the high-frequency component of the upper boundary condition if the regularization parameter γ was appropriate.

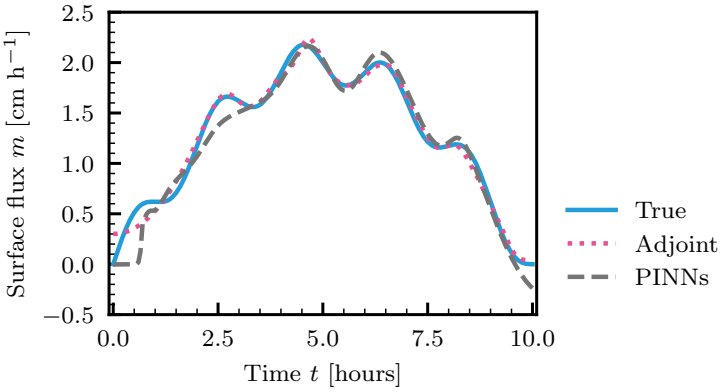

Figure 4.16: Example 2. The true (blue solid line) and the estimated surface water flux from data at five measurement locations $z = -10, -20, 30, -40, -50$ cm for the adjoint method (dotted pink) and PINNs (dotted grey).

Table 4.4: Example 2: The performance of Newton-Conjugate gradient (CG) method.

Newton iteration	CG iterations	Line searches	Cost	Norm of the gradient
1	1	16	1.25×10^{-1}	2.10×10^{-1}
2	1	0	9.51×10^{-2}	2.01×10^{-1}
3	1	0	2.55×10^{-2}	1.31×10^{-1}
4	1	0	2.03×10^{-2}	5.72×10^{-2}
5	2	0	5.07×10^{-3}	3.87×10^{-2}
6	1	0	3.64×10^{-3}	3.74×10^{-2}
7	2	0	1.13×10^{-3}	1.64×10^{-2}
8	3	0	7.21×10^{-4}	6.57×10^{-3}
9	11	0	5.98×10^{-4}	8.90×10^{-4}
10	9	0	5.94×10^{-4}	3.38×10^{-4}
11	35	0	5.92×10^{-4}	7.96×10^{-5}
12	23	0	5.92×10^{-4}	9.75×10^{-6}
13	48	0	5.92×10^{-4}	6.28×10^{-7}
14	68	0	5.92×10^{-4}	1.69×10^{-9}

4.5.3 Example 3

This example examines the performance of the two methods for longer soil moisture dynamics. Fig. 4.21 (**a**) and (**b**) show the best-fitted soil moisture θ for the adjoint method and PINNs, respectively. Fig. 4.22 and 4.23 show more detailed distribution of each method. The regularization parameter was $\gamma = t \times 10^{-5}$ and $\gamma = 0.1$ for the adjoint method and PINNs, respectively. The accuracy of the reconstructed forward solution ϵ^{θ} was 0.0489 and 0.131 for the adjoint method and PINNs, respectively. This numerical example clearly demonstrates the advantage of the adjoint method for a data-limited case. In this case, only the soil moisture data at $z = -5cm$ was available, and the PINNs could not reconstruct the solution to the forward problem

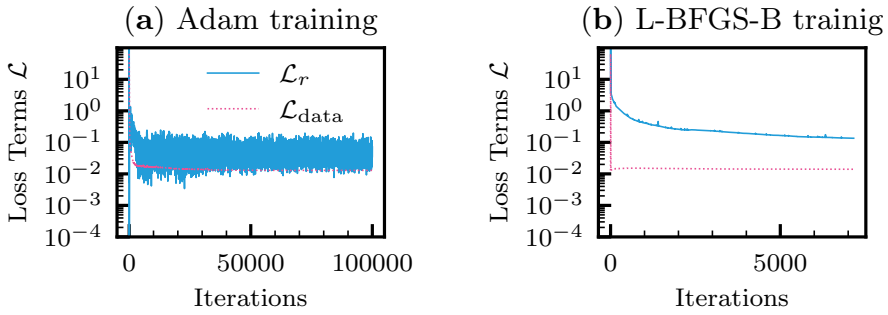

Figure 4.17: Example 2: The evoluion of the loss terms for the Adam optimizer (**a**) and the L-BFGS-B optimizer (**b**).

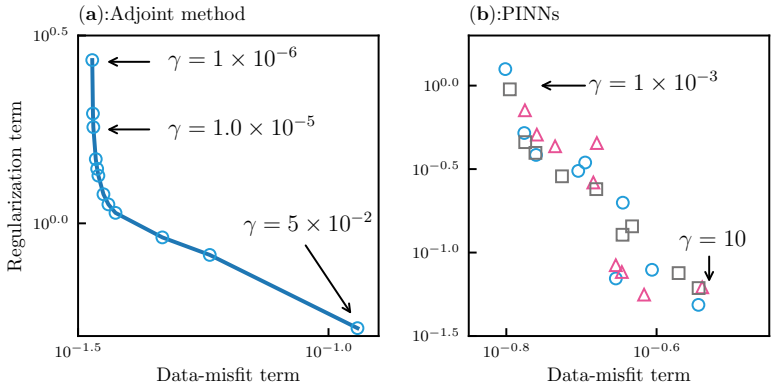

Figure 4.18: Example 2: The L-curve for the adjoint method (**a**) and physics-informed neural networks (PINNs) (**b**). The different colors in (**b**) represent results from different initialization of NNs.

accurately (see Fig. 4.21 (**b**)). However, note that the residual of the PDE was better minimized than those for the previous examples, as shown in Fig. 4.24, which was due to the larger number of residual points.

Nevertheless, the estimated surface flux m by PINNs was not particularly worse than that of the adjoint method (ϵ^m was 0.0797 and 0.111 for the adjoint method and PINNs, respectively), which is shown in Fig. 4.25.

Next, we discuss the performance of the minimization algorithms. Table 4.5 shows the result for the Newton-CG method for the adjoint method. The total CG iteration was 623, and it took 21533 seconds. This was because the upper boundary condition

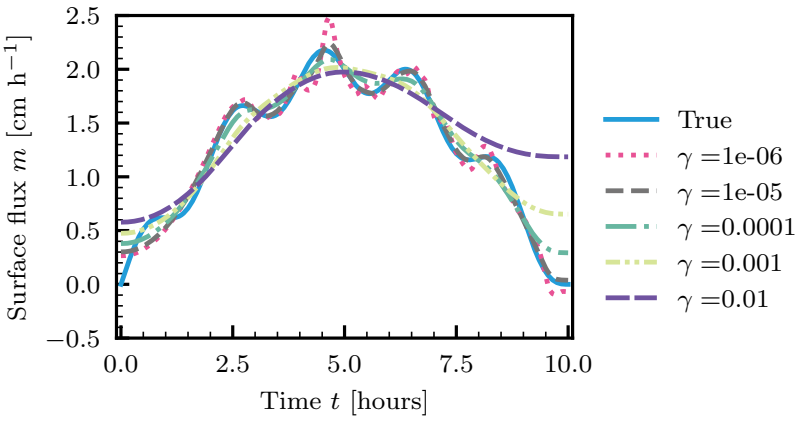

Figure 4.19: Example 2: The effect of the regularization parameter γ on the estimated solution m for the adjoint method.

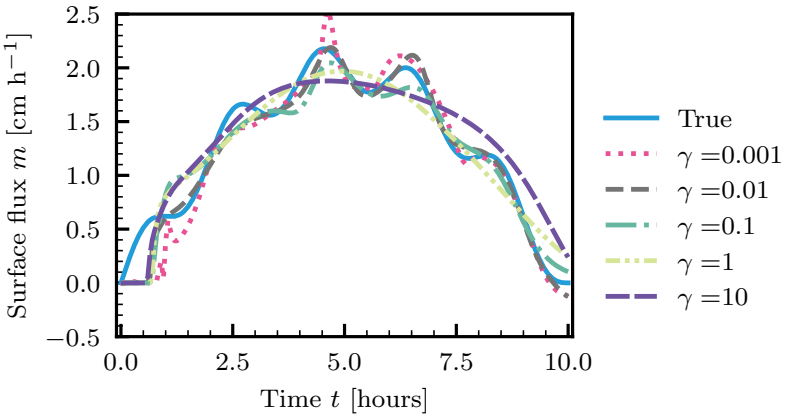

Figure 4.20: Example 2: The effect of the regularization parameter γ on the estimated solution m for PINNs.

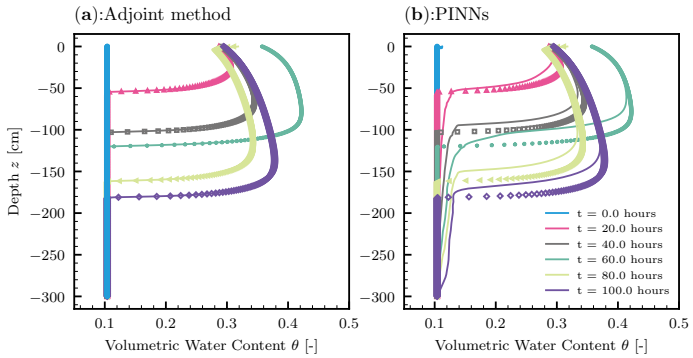

Figure 4.21: Example 3: The best-fitted solution to the forward problem (solid lines) against the true solution (markers) for the adjoint method (**a**) and PINNs (**b**) from data at five measurement locations $z = -5$ cm.

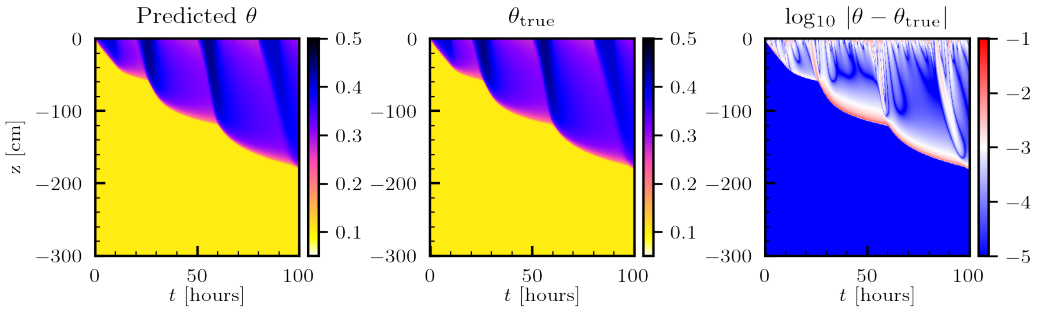

Figure 4.22: Example 3: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for the adjoint method. A small number 10^{-9} was added to the error for the visualization.

has more modes than previous examples. As for PINNs, the evolution of the loss terms is shown in Fig. 4.26 (**a**) and (**b**) for the Adam and the L-BFGS-B optimizers, respectively. Regarding computational time, it took 1820 and 23596 seconds for the Adam and the L-BFGS-B optimizers, respectively. Although both methods required much longer computational time, the adjoint method was still faster than PINNs. Nevertheless, both methods have plenty of room for improvement in the implementation. For example, an adaptive time-stepping method would drastically speed up the adjoint method.

Finally, we discuss the effect of the regularization parameter γ for each method. Fig. 4.27 (**a**) and (**b**) shows the L-curve for the adjoint method and PINNs, respectively. For the adjoint method, the regularization parameters were
$\gamma \in \{10^{-5}, 5^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$, while
$\gamma \in \{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1, 5, 10\}$ were used for PINNs. For the

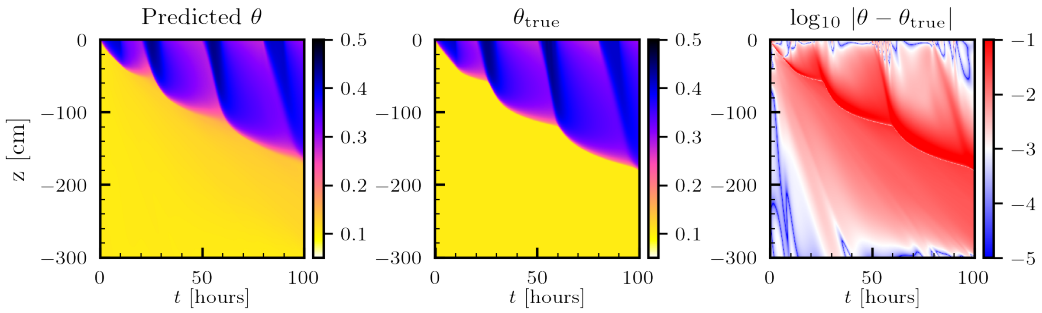

Figure 4.23: Example 3: The predicted (left), the true (center), and the error in log scale (right) volumetric water content θ for PINNs.

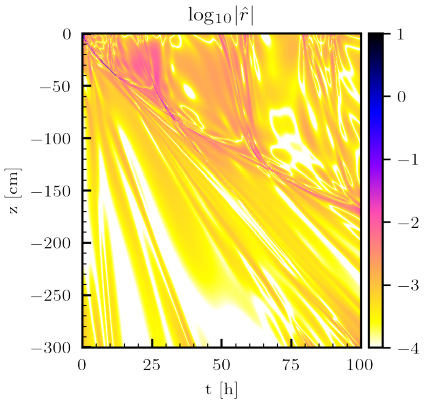

Figure 4.24: Example 3: The residual of the PDE on log-scale for PINNs.

adjoint method, the best result was obtained with $\gamma = t \times 10^{-5}$, which is far from at which the curvature of the L-curve is maximum. For PINNs, $\gamma = 0.1$ gave the best result. Fig 4.28 shows the effect of the regularization parameter γ on the inverted solution m for the adjoint method. When the regularization parameter was $\gamma = 10^{-5}$, the inverted solution was severely affected by the noise in the data, particularly during which the evaporation was simulated. On the other hand, when the regularization parameter γ was larger than 0.01, the adjoint method underestimated the surface water flux for the rainfall events. The result demonstrates the difficulty in estimating the surface water flux for evaporation periods, which are small in magnitude and prone to noise in the data. Similar results were obtained for PINNs, as shown in Fig. 4.29. Interestingly, compared to the adjoint method, the surface water flux estimate during the evaporation periods was less affected by the noise in the data for PINNs. Nevertheless, it is necessary to choose the right regularization parameter γ to recover the upper boundary condition well.

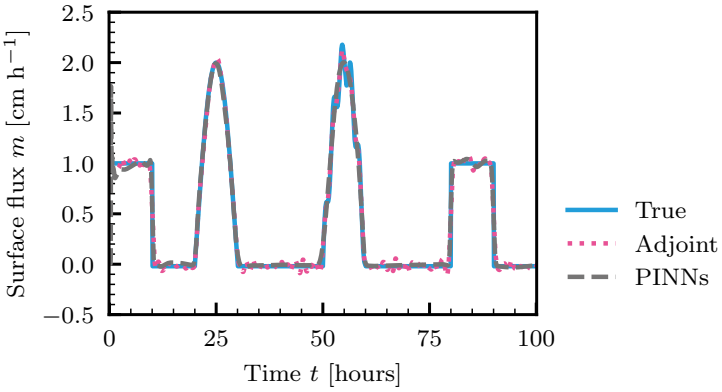

Figure 4.25: Example 3. The true (blue solid line) and the estimated surface water flux from data at five measurement locations $z = -5$ cm for the adjoint method (dotted pink) and PINNs (dotted grey).

Table 4.5: Example 3: The performance of Newton-Conjugate gradient (CG) method.

Newton iteration	CG iterations	Line searches	Cost	Norm of the gradient
1	2	0	2.59×10^{-1}	4.39×10^{-1}
2	2	0	2.15×10^{-1}	6.15×10^{-2}
3	1	11	9.38×10^{-2}	3.94×10^{-2}
4	2	2	5.41×10^{-2}	4.48×10^{-2}
5	5	6	3.41×10^{-2}	5.86×10^{-2}
6	4	1	1.65×10^{-2}	4.95×10^{-2}
7	4	0	7.81×10^{-3}	2.93×10^{-2}
8	6	0	3.09×10^{-3}	1.27×10^{-2}
9	16	0	1.62×10^{-3}	5.17×10^{-3}
10	57	0	1.38×10^{-3}	1.37×10^{-3}
11	107	0	1.37×10^{-3}	2.39×10^{-4}
12	172	0	1.37×10^{-3}	1.50×10^{-5}
13	245	0	1.37×10^{-3}	9.91×10^{-8}

4.6 Conclusions

Surface water flux is an essential variable for land surface modeling and water management. Estimating surface water flux from soil moisture measurements can be formulated as the inverse problem governed by the Richardson-Richards equation (RRE). This inverse problem is a large-scale numerical optimization problem and thus requires efficient techniques to overcome the computational challenge.

We applied two inverse methods to the inverse problem: physics-informed neural networks (PINNs) and an adjoint method. In PINN framework, the solution to the RRE is approximated by NNs, and NNs are trained by available data and the constraint by the PDE. On the other hand, the RRE is solved by a finite element method in the adjoint method, and the inverse problem is solved by minimizing

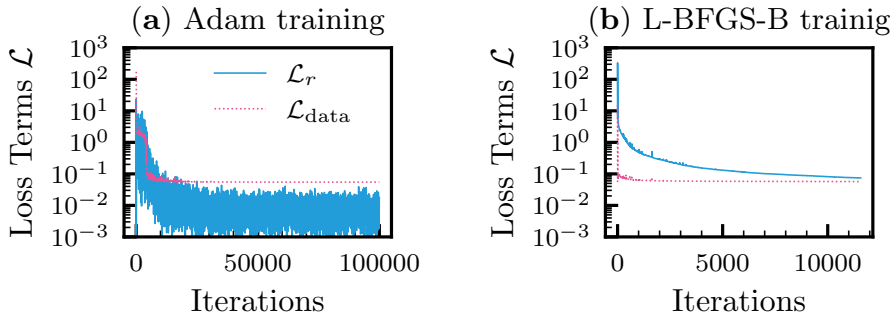

Figure 4.26: Example 3: The evoluion of the loss terms for the Adam optimizer (**a**) and the L-BFGS-B optimizer (**b**).

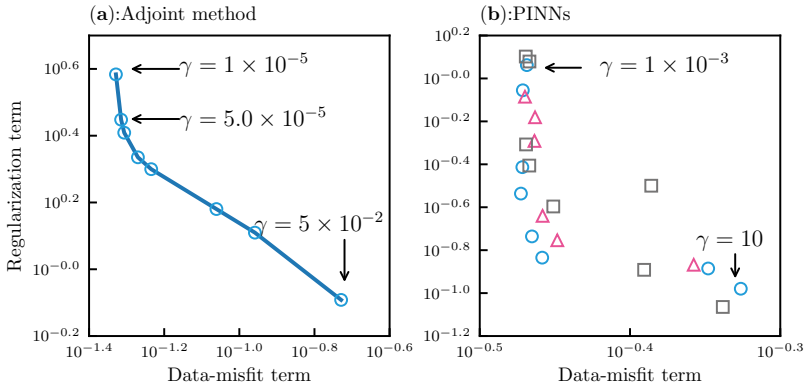

Figure 4.27: Example 3: The L-curve for the adjoint method (**a**) and physics-informed neural networks (PINNs) (**b**). The different colors in (**b**) represent results from different initialization of NNs.

an objective function via the (inexact) Newton method with the gradient and the Hessian information (Hessian-vector product) obtained by solving a so-called adjoint problem and the linearized forward and adjoint problems. Because the automatic differentiation used in training NNs is a special case of an adjoint method, it would be valuable to compare the two methods.

We investigated the performance, advantages, and limitations of the two methods through three numerical examples. The first numerical example demonstrated that both methods could estimate the upper boundary condition from soil moisture measurements at five locations. For this example, the adjoint method was faster than

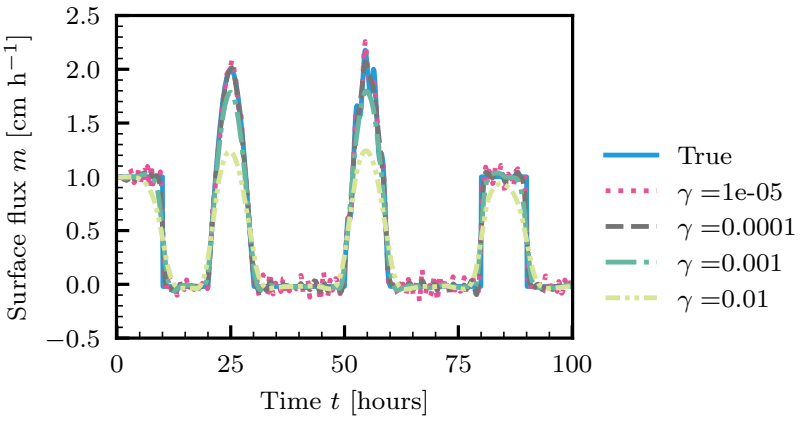

Figure 4.28: Example 3: The effect of the regularization parameter γ on the estimated solution m for the adjoint method.

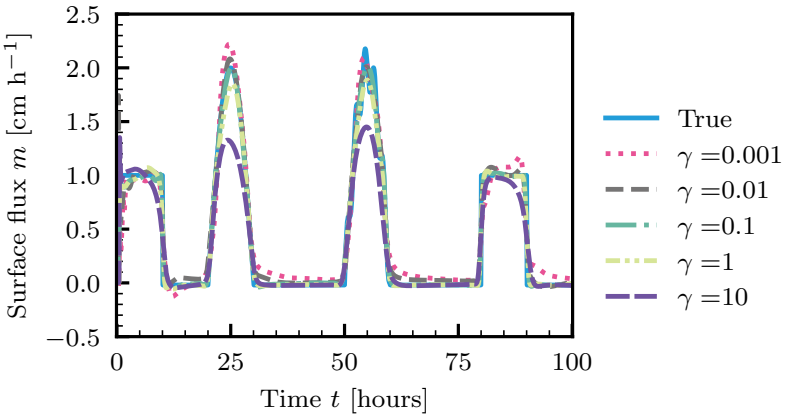

Figure 4.29: Example 3: The effect of the regularization parameter γ on the estimated solution m for PINNs.

PINNs. The effect of the regularization parameter γ for each method was investigated, and both behaved similarly for large γ but not for small γ, where the adjoint method was more affected by the noise in the data than PINNs.

The second numerical example investigated the ability of the two methods to extract a high-frequency component of the upper boundary condition. It was found that both methods could extract the high-frequency component if the regularization parameter γ was appropriately chosen. Compared to Example 1, it took longer computational times for both methods because the data used in this example were more informative than in the first example.

In the final example, the robustness of the two methods for a more realistic longer soil moisture dynamics (100 hours), where only soil moisture measurement at $z = -5$ cm was used for the inverse modeling. It was demonstrated that the adjoint method was superior to PINNs in the data-limited situation in terms of the accuracy of the reconstructed forward solution. The estimated surface water flux was satisfactory for both methods, while both methods required much longer computational time than previous examples. Although the adjoint method was faster than PINNs in our computational environment, the adjoint method needed about 6 hours for the inverse modeling.

This result was unsatisfactory considering our original expectation because we aimed to combine this algorithm with other algorithms to estimate the upper boundary condition and soil hydraulic properties simultaneously. The adjoint method could speed up if we use an adaptive time-stepping method. Although the study demonstrates the robustness of the adjoint method than PINNs, it is still noteworthy of the NN capability to approximate the solution to the PDEs. The number of NN parameters was much smaller than that of finite element coefficients used in the adjoint method. Also, the study illustrates the importance of the regularization parameter for both methods. Unfortunately, a common strategy used in the adjoint method (L-curve method) did not provide the best estimate of the regularization parameter. The regularization parameter for PINNs is still unclear though the relationship with the adjoint variable in the adjoint method should be explored. Finally, it would be highly desirable to implement the adjoint method with automatic differentiation to apply this method with more complicated but realistic soil hydraulic functions, including the effect of hysteresis.

4.7 List of abbreviations

BC: Brooks and Corey
DTO: Discretize-then-Optimize
FEMs: Finite element methods
HCF: Hydraulic conductivity function
L-LAAF: Layer-wise locally adaptive activation Function
NNs: Neural networks
OTD: Optimize-then-discretize

PDE: Partial differential equation
PINNs: Physics-informed neural networks
RRE: Richardson-Richards equation
WRC: Water retention curve

4.8 List of notation

superscript
$\hat{\cdot}$: test function in the adjoint method and prediction in PINNs except for used for $\hat{\Theta}$
$\tilde{\cdot}$: incremental variables
$\cdot^k$: kth Newton iteration
$\cdot^n$: nth time step

subscript
$\cdot_D$: Dirichlet boundary condition
$\cdot_F$: water flux boundary condition
$\cdot_h$: discretized space or function
$\cdot_{\mathrm{obs}}$: observation
$\cdot_0$: initial condition
$\cdot_{lb}$: lower boundary condition
$\cdot_{\mathrm{data}}$: data for PINNs training
$\cdot_r$: residual
$\cdot_{ub}$: upper boundary condition
$\cdot_U$: upper layer

alphabet
$\mathbf{a}$: the collection of trainable parameters for adaptive activation functions for a neural network
$a^{[k]}$: trainable parameter for the element-wise non-linear activation function
$\mathbf{A}$: derivative of the constraint by the forward problem
$\mathbf{b}$: the collection of bias vectors for a neural network
$\mathbf{b}^{[k]}$: bias vector for the kth hidden layer
$\mathbf{B}$: matrix from the observational operator
$\mathcal{B}$: observational operator
$\mathcal{B}^*$: adjoint of observational operator
c: parameter for line search
$\mathbf{C}$: derivative of the constraint by the forward problem by the parameter
$\mathbf{d}$: derivative of the data-misfit term by the state variable
D: data-misfit term
$\mathbf{F}$: system of non-linear equations
$\mathbf{F}'$: Jacobian matrix
$\mathbf{g}$: gradient vector

$\mathbf{h}^{[k]}$: vector for the kth hidden layer
H: Hessian matrix
H^1: Sobolev space
$\mathbf{J}$: discretized objective function
$\mathcal{J}$: objective functional
K: hydraulic conductivity [L T^{-1}]
$\mathbf{K}$: stiffness matrix
$\mathbf{K}^m$: stiffness matrix on parameter space
K_s: saturated hydraulic conductivity [L T^{-1}]
l: tortuosity parameter [-]
L: the number of hidden layers
L^g: Lagrangian for the gradient evaluation
L^H: Lagrangian for the Hessian-vector product evaluation
$\mathcal{L}$: loss function in PINNs
$\mathcal{L}^g$: Lagrangian functional for the gradient evaluation
$\mathcal{L}^H$: Lagrangian functional for the Hessian-vector product evaluation
$\mathcal{L}_i$: loss term for i constraints
m: Neumann flux boundary condition
$\mathbf{m}$: discretized flux boundary condition
$\tilde{\mathbf{m}}$: Newton direction
$\mathbf{M}$: mass matrix
$\mathcal{M}$: functional space for parameter space
$\mathbf{M}^L$: lumped mass matrix
$\mathcal{N}$: neural network functions
$n^{[k]}$: dimension of a vector corresponding to the kth hidden layer
n^x: dimension of input vector $\mathbf{x}$
n^y: dimension of output vector $\hat{\mathbf{y}}$
N_{data}: the number of data points
N_r: the number of residual points
N_s: the number of spatial nodes
N_t: the number of temporal steps
o: output functions
p: test function
$\mathbf{p}$: Discretized adjoint variable
$\mathbf{P}$: regularization operator
q: water flux in the vertical direction (positive downward) [L T^{-1}] or test function in the OTD approach
$\hat{r}$: the residual of the RRE
$\mathcal{R}$: regularization term
s: fixed scaling factor for adaptive activation functions
S_e: effective saturation [-]
t: time [T]
T: final time [T]
$\mathcal{V}$: trial space

$\hat{\mathcal{V}}$: test space
$\mathbf{W}$: the collection of the weight matrices for a neural network
$\mathbf{W}^{[k]}$: weight matrix for the kth hidden layer
$\mathbf{x}$: input vector
$\hat{\mathbf{y}}$: output vector
z: vertical coordinate (positive upward) [L]
Z: the vertical length of a soil [L]

Greek alphabet
α: step size of line search
β: fixed parameter for the output of neural networks
γ: regularization parameter
δ_{ij}: Kronecker symbol
Δt^n: time step at the time t^n [T]
ϵ^δ: L_2 relative error for variable δ
η: forcing term
θ: volumetric water content [L^3 L^{-3}]
Θ: neural network parameters
$\hat{\Theta}$: update of neural network parameters for each iteration of optimization algorithms
θ^n: collection of θ at all spatial nodes for the time t^n
θ_r: residual volumetric water content [L^3 L^{-3}]
θ_s: saturated volumetric water content [L^3 L^{-3}]
λ: pore-size distribution parameter [-]
σ: element-wise non-linear activation function
τ_a: absolute threshold for Newton iterations
τ_r: relative threshold for Newton iterations
ψ: water potential in soils [L]
ψ_0: initial condition of the water potential in soils [L]
ψ_c: air-entry value [L]
ψ_{lb}: water potential at the bottom boundary [L]
χ_i: linear function at node i for parameter space
ϕ_i: linear function at node i
Ω: spatial domain
$\partial\Omega$: spatial boundary

others
$:=$: equal by definition
∇: Nabla
$\bar{\cdot}$: assembly over all the time steps

References for Chapter 4

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., J., D., Devin, M., Ghemawat, S., I., G., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous distributed systems.

Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3.

Assouline, S. and Or, D. (2013). Conceptual and parametric representation of soil hydraulic properties: A review. *Vadose Zone Journal*, 12(4):1–20.

Bandai, T. and Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*, 57.

Bandai, T. and Ghezzehei, T. A. (2022). Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition. *Hydrology and Earth System Sciences*, 26(16):4469–4495.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43.

Brocca, L., Moramarco, T., Melone, F., and Wagner, W. (2013). A new method for rainfall estimation through soil moisture observations. *Geophysical Research Letters*, 40(5):853–858.

Brooks, R. H. and Corey, A. T. (1964). Hydraulic properties of porous media. *Hydrology Papers 3, Colorado State University*.

Buckingham, E. (1907). *Studies on the movement of soil moisture*, volume 38 of *Bulletin*. Washington, DC.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *Journal of Scientific Computing*, 16(5):1190–1208.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, 26(7):1483–1496.

Chavent, G. (2010). *Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications*. Springer.

Cosh, M. H., Caldwell, T. G., Baker, C. B., Bolten, J. D., Edwards, N., Goble, P., Hofman, H., Ochsner, T. E., Quiring, S., Schalk, C., Skumanich, M., Svoboda, M., and Woloszyn, M. E. (2021). Developing a strategy for the national coordinated soil moisture monitoring network. *Vadose Zone Journal*, 20:e20139.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals Systems*, 2:303–314.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, 4(January):2933–2941.

Dijkema, J., Koonce, J. E., Shillito, R. M., Ghezzehei, T. A., Berli, M., van der Ploeg, M. J., and van Genuchten, M. T. (2017). Water distribution in an arid zone soil: Numerical analysis of data from a large weighing lysimeter. *Vadose Zone Journal*.

Dorigo, W., Himmelbauer, I., Aberer, D., Schremmer, L., Petrakovic, I., Zappa, L., Blöschl, G., Bogena, H., Brocca, L., Calvet, J.-c., Camarero, J. J., and Capello, G. (2021). The International Soil Moisture Network : serving Earth system science for over a decade. *Hydrology and Earth System Sciences*, 25:5749–5804.

Eisenstat, S. C. and Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM Journal of Scientific Computing*, 17(1):16–32.

Fuks, O. and Tchelepi, H. A. (2020). Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1):19–37.

Ghattas, O. and Willcox, K. (2021). Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, Sardinia, Italy, May 13 - 15.

Groh, J., Stumpp, C., Lücke, A., Pütz, T., Vanderborght, J., and Vereecken, H. (2018). Inverse estimation of soil hydraulic and transport parameters of layered soils from water stable isotope and lysimeter data. *Vadose Zone Journal*.

Guermond, J.-L. and Pasquetti, R. (2013). A correction technique for the dispersive effects of mass lumping for transport problems. *Computer Methods in Applied Mechanics and Engineering*, 253:186–198.

Haber, E. and Ruthotto, L. (2018). Stable architectures for deep neural networks. *Inverse Problems*, 34:014004.

Hansen, P. C. (2000). The L-Curve and its use in the numerical treatment of inverse problems. *in Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering*, 4:119–142.

Hornik, K. (1991). Approximation capabilities of multilayer neural network. *Neural Networks*, 4(1991):251–257.

Isaac, T., Petra, N., Stadler, G., and Ghattas, O. (2015). Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *Journal of Computational Physics*, 296:348–368.

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136.

Jin, X., Cai, S., Li, H., and Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440.

Kelley, C. T. (2018). Numerical methods for nonlinear equations. *Acta Numerica*, 27:207–287.

Kingma, D. P. and Ba, J. B. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd international conference on learning representations*, San Diego, CA, USA, May 7 - 9.

Kosugi, K. (1996). Lognormal distribution model for unsaturated soil hydraulic properties. *Water Resources Research*, 32(9):2697–2703.

Logg, A., Mardal, K. A., and Wells, G. N. (2012). *Automated solution of differential equations by the finite element method*. Springer.

Mowlavi, S. and Nabi, S. (2022). Optimal control of PDEs using physics-informed neural networks. *arXiv*, pages 1–25.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, USA.

Patel, R. G., Manickam, I., Trask, N. A., Wood, M. A., Lee, M., Tomas, I., and Cyr, E. C. (2022). Thermodynamically consistent physics-informed neural networks for hyperbolic systems. *Journal of Computational Physics*, 449:110754.

Petra, N., Zhu, H., Stadler, G., Hughes, T. J., and Ghattas, O. (2012). An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model. *Journal of Glaciology*, 58(211):889–903.

Pollock, M. D., O'Donnell, G., Quinn, P., Dutton, M., Black, A., Wilkinson, M. E., Colli, M., Stagnaro, M., Lanza, L. G., Lewis, E., Kilsby, C. G., and O'Connell, P. E. (2018). Quantifying and mitigating wind-induced undercatch in rainfall measurements. *Water Resources Research*, 54(6):3863–3875.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Rawls, W. J., Brakensiek, C. L., and Saxton, K. E. (1982). Estimation of soil water properties. *Transactions - American Society of Agricultural Engineers*, 25(5):1316–1320.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333.

Richardson, L. F. (1922). *Weather prediction by numerical process*. Cambridge University Press, Cambridge, United Kingdom.

Sadeghi, M., Tuller, M., Warrick, A. W., Babaeian, E., Parajuli, K., Gohardoust, M. R., and Jones, S. B. (2019). An analytical model for estimation of land surface net water flux from near-surface soil moisture observations. *Journal of Hydrology*, 570:26–37.

Santos, J., Efendiev, Y., and Guarracino, L. (2006). Hydraulic conductivity estimation in partially saturated soils using the adjoint method. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):161–179.

Scudeler, C., Putti, M., and Paniconi, C. (2016). Mass-conservative reconstruction of Galerkin velocity fields for transport simulations. *Advances in Water Resources*, 94:470–485.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 23(8):26–28.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America*, 44:892–898.

Villa, U., Petra, N., and Ghattas, O. (2020). hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs. Part I: Deterministic inversion and linearized Bayesian inference. *ACM*, 2657(1):1–9.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Wang, S., Yu, X., and Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768.

Wight, C. L. and Zhao, J. (2021). Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. *Communications in Computational Physics*, 29(3):930–954.

Wilcox, L. C., Stadler, G., Bui-Thanh, T., and Ghattas, O. (2015). Discretely exact derivatives for hyperbolic PDE-constrained optimization problems discretized by the discontinuous Galerkin method. *Journal of Scientific Computing*, 63:138–162.

Willcox, K. E., Ghattas, O., and Heimbach, P. (2021). The imperative of physics-based modeling and inverse theory in computational science. *Nature Computational Science*, 1:166–168.

Chapter 5

Conclusion Remarks

In the dissertation, I explored various aspects of inverse modeling of soil moisture dynamics based on the Richardson-Richards equation (RRE)
(Richardson, 1922; Richards, 1931). As concluding remarks, I summarize the limitation of the current approaches and discuss future perspectives.

Chapter 2 showed that physics-informed neural networks (PINNs) (Raissi et al., 2019) were promising for inverse modeling of soil moisture dynamics. However, the current limitation of PINNs is that the application of PINNs to real soil moisture data is still challenging. Almost all the studies of testing PINNs are limited to simple synthetic data, and very few used PINNs with actual observations (Depina et al., 2021). It is challenging to train PINNs with actual soil moisture data because they are sparse and noisy. This results from using feedforward neural networks in PINNs, which provide tremendous flexibility but lack structure. Future studies might solve this issue by finding more problem-specific architecture of neural networks or transforming training data.

In Chapter 3, I demonstrated that the RRE with the Peters-Durner-Iden (PDI) model (Peters, 2013; Iden and Durner, 2014; Peters, 2014) could capture soil moisture dynamics more accurately than the one with the van Genuchten-Mualem (VGM) model (Mualem, 1976; van Genuchten, 1980) using upward infiltration experimental data. Nevertheless, the inverse modeling framework was only successful for medium-textured soils, and I faced difficulties with coarse-textured and clay-rich soils. As for coarse-textured, the experimental data violated the one-dimensional flow assumed in the RRE. Although this could be alleviated by a better design of experimental setup, it may be more appropriate to use two-dimensional RRE by taking advantage of two-dimensional soil moisture data provided by a shortwave infrared imaging camera. However, it would increase the computational burden significantly. For clay-rich soils, the heterogeneity of the soils violated the homogeneous assumption used in the RRE. Dealing with such heterogeneities in the inverse modeling framework is computationally very challenging, while Miller and Miller's similar media theory (Miller and Miller, 1956) might be promising. In this work, I used a gradient-free global optimization algorithm because it has been recognized that gradient-based algorithms are not suitable for inverse modeling to estimate soil hydraulic properties due to many bad

local optima. However, I believe this is due to the parameterization of soil hydraulic properties (e.g., the VGM and the PDI model), not the problem itself (i.e., the RRE and soil moisture data). Future studies should aim to find better parameterization of soil hydraulic properties that allow the use of gradient-based optimization algorithms, which are faster than global optimization algorithms. Promising approaches would be replacing such parametric models with more flexible functions, such as neural networks (Bandai and Ghezzehei, 2021) and finite elements.

In Chapter 4, I discussed the estimation of surface water flux from soil moisture measurements through an adjoint method (Ghattas and Willcox, 2021). The comparison between the adjoint method and PINNs clarified the similarities and differences between the methods, and numerical experiments demonstrated that the adjoint method is more robust than PINNs in terms of the reconstructed soil moisture profile for a data-limited case. One of the main limitations of the current adjoint method is that it does not optimize soil hydraulic properties. We can combine the adjoint method with another optimization algorithm for a small-scale problem to estimate surface water flux and soil hydraulic properties simultaneously. However, for large-scale problems, it would be computationally very demanding. Although we could re-implement the adjoint method for parameters to describe soil hydraulic properties, such implementation is time-consuming and error-prone. Using automatic differentiation would solve such issues but be slower than the adjoint method. Nevertheless, automatic differentiation (Baydin et al., 2018) would be necessary considering the complexity of soil hydraulic properties, including the effect of hysteresis (Assouline and Or, 2013). Therefore, future studies should combine adjoint methods with automatic differentiation. For example, at the initial stages of research, automatic differentiation enables us to implement inverse modeling much easier than adjoint methods. Once computational demand becomes an issue, we could replace automatic differentiation with adjoint methods. Such hybrid approaches would enable us to develop an inverse modeling framework that can analyze actual soil moisture data.

References for Chapter 5

Assouline, S. and Or, D. (2013). Conceptual and parametric representation of soil hydraulic properties: A review. *Vadose Zone Journal*, 12(4):1–20.

Bandai, T. and Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*, 57.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43.

Depina, I., Jain, S., Valsson, S. M., and Gotovac, H. (2021). Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*.

Ghattas, O. and Willcox, K. (2021). Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554.

Iden, S. C. and Durner, W. (2014). Comment on "Simple consistent models for water retention any hydraulic conductivity in the complete moisture range" by A. Peters. *Water Resources Research*, 50:7530–7534.

Miller, E. E. and Miller, R. D. (1956). Physical theory for capillary flow phenomena. *Journal of Applied Physics*, 27(4):324–332.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522.

Peters, A. (2013). Simple consistent models for water retention and hydraulic conductivity in the complete moisture range. *Water Resources Research*, 49:6765–6780.

Peters, A. (2014). Reply to comment by S. Iden and W. Durner on "Simple consistent for water retention and hydraulic conductivity in the complete moisture range". *Water Resources Research*, 50:7535–7539.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333.

Richardson, L. F. (1922). *Weather prediction by numerical process*. Cambridge University Press, Cambridge, United Kingdom.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America*, 44:892–898.

Appendix A

Supplementary materials for Chapter 2

A.1 Supplementary Text

A.1.1 Residual-based adaptive refinement

The original PINN framework proposed by Raissi et al. (2019) has difficulty in approximating the solution of PDEs that have steep gradients. To overcome this challenge, Lu et al. (2021) proposed the residual-based adaptive refinement (RAR) algorithm, which distributes collocation points during the training in the locations where the residual of PDEs is large.

In this study, as in Lu et al. (2021), after 10000 iterations of the Adam optimizer, the residual was evaluated at randomly sampled 10^6 locations from the whole spatial and temporal domain (see Fig. A.1 (**b**)). The collocation points were ordered according to the residual values, and the highest ten collocation points were added to the collocation points that were originally given. We iterated this procedure ten times before the L-BFGS-B algorithm was used to further minimize the loss function.

We tested the RAR algorithm for the forward modeling of the homogeneous soil (Sect. 3.1 in the main text), and the results are shown in Fig. A.1. The RAR algorithm appeared to improve the performance of PINNs for the problem, but the effects were minor. Therefore, we did not use the algorithm for further analysis.

A.1.2 Learning rate annealing

Wang et al. (2021) proposed the adaptive learning rate (ALR) algorithm, where the weight parameters in the loss function λ_i are updated in the following way:

$$\lambda_i^{n+1} = (1-\alpha)\lambda_i^n + \alpha\hat{\lambda_i}^n \quad \text{for} \quad i = m, ic, D, F, \tag{A.1}$$

where

$$\hat{\lambda_i}^n = \frac{\max_{\mathbf{W}^n}\{|\nabla_{\mathbf{W}^n}\mathcal{L}_r(\Theta^n)|\}}{\overline{|\nabla_{\mathbf{W}^n}\lambda_i^n\mathcal{L}_i(\Theta^n)|}}, \quad \text{for} \quad i = m, ic, D, F, \tag{A.2}$$

where the bar represents the mean of the values below the bar; Θ^n is the neural network parameters including the weight matrices $\mathbf{W}^n$ at nth iteration of the algorithm. In the study, α was set to 0.1, and the algorithm was used to update λ_i every 10 iterations of the the Adam algorithm to balance the relative importance of each loss term .

The ALR algorithm was tested for the forward modeling for the homogeneous soil case in Sect. 3.1 of the main text. The three weight parameters λ_{ic}, λ_{ub}, and λ_{lb} for the initial, upper boundary, and lower boundary condition, respectively, were initially set to ten, while they were updated using the ALR algorithm during the training (see Fig. A.2 (**b**)). Figure A.2 demonstrated that the effectiveness of the ALR algorithm was not clear compared to the L-LAAF algorithm. Figure A.2 (**c**) showed that the loss term for the residual $\mathcal{L}_r$ was not minimized as the L-LAAF algorithm (shown in Fig. 4 of the main text). Therefore, we only used the L-LAAF algorithm in the study for further analysis.

A.1.3 Finite difference method

A finite difference method was implemented on Matlab R2020b to solve the one-dimensional RRE to evaluate the performance of PINNs. To deal with the non-linear terms in the RRE, the modified Picard iteration was used (Celia et al., 1990). A constant spatial mesh size dz and time step dt were used. The internodal hydraulic conductivity K was computed from the geometric average of the adjacent nodes. The upper boundary condition given as a constant water flux was evaluated using a second-order one-sided finite difference approximation (LeVeque, 2007). Figure A.3 shows that the numerical error ϵ^θ decreased with decreasing dt (Fig. A.3 (**a**)) and dz (Fig. A.3 (**b**)).

A.1.4 Effects of weight parameters in loss function for heterogeneous case

In Sect. 3.2.4 in the main text, we investigated the effects of weight parameters λ_i in the loss function for the heterogeneous case. Here, the effects on the loss terms are shown. Figure A.4, A.5, and A.6 shows the effects on the loss terms for the upper layer, the lower layer, and the interface conditions, respectively.

A.1.5 Use of water potential measurements

It is straightforward to modify the PINN algorithm for water potential measurements. Note that we used the logarithm of the negative water potential measurements in the loss function. We used the water potential measurement data from the same HYDRUS-1D simulation used in the inverse modeling in the main text to estimate the surface water flux. Figure A.9 showed that the estimated surface flux was comparable to the case using volumetric water content.

A.2 Supplementary Figures

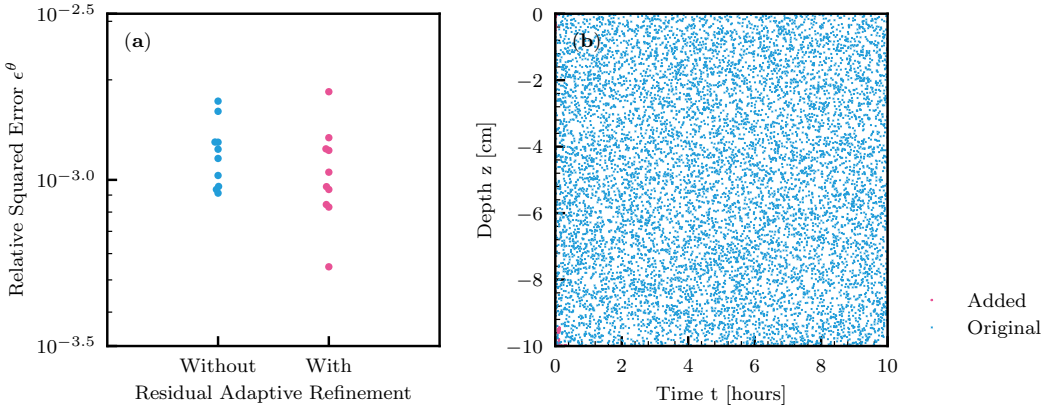

Figure A.1: (**a**): The effects of the residual-based adaptive refinement algorithm on the performance of PINNs for the forward problem for the homogeneous soil. (**b**): The distribution of the original and added collocation points for the same problem.

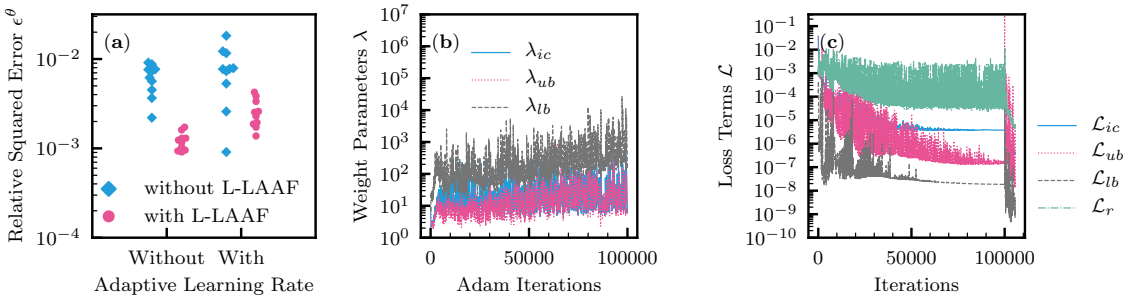

Figure A.2: The effects of the adaptive learning rate (ALR) algorithm for the forward problem of the homogeneous soil case. (**a**): The relative squared error ϵ^θ for PINNs with and without the ALR and L-LAAF algorithms. (**b**): The evolution of the weight parameters in the loss function during the Adam algorithm. (**c**): The evolution of the loss terms during the training.

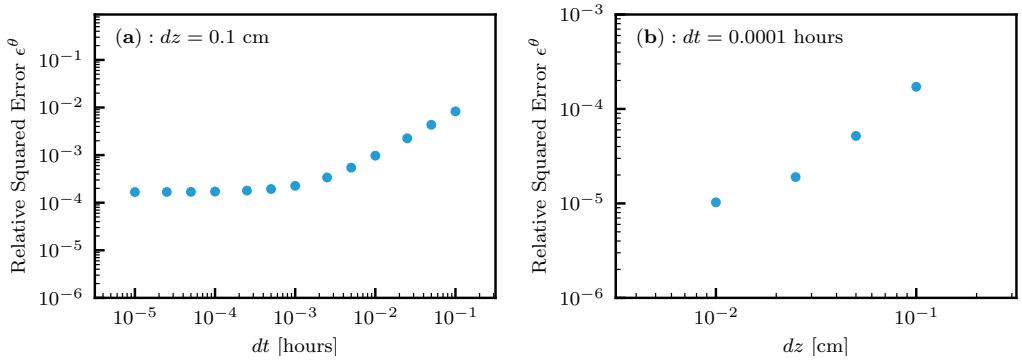

Figure A.3: (**a**): The relative squared error with respect to volumetric water content ϵ^θ for the finite difference solution with varying time steps dt. The spatial mesh size dz was fixed to 0.1 cm. (**b**): The relative squared error with respect to volumetric water content ϵ^θ for varying spatial mesh size dz. The time step dt was fixed to 0.0001 h.

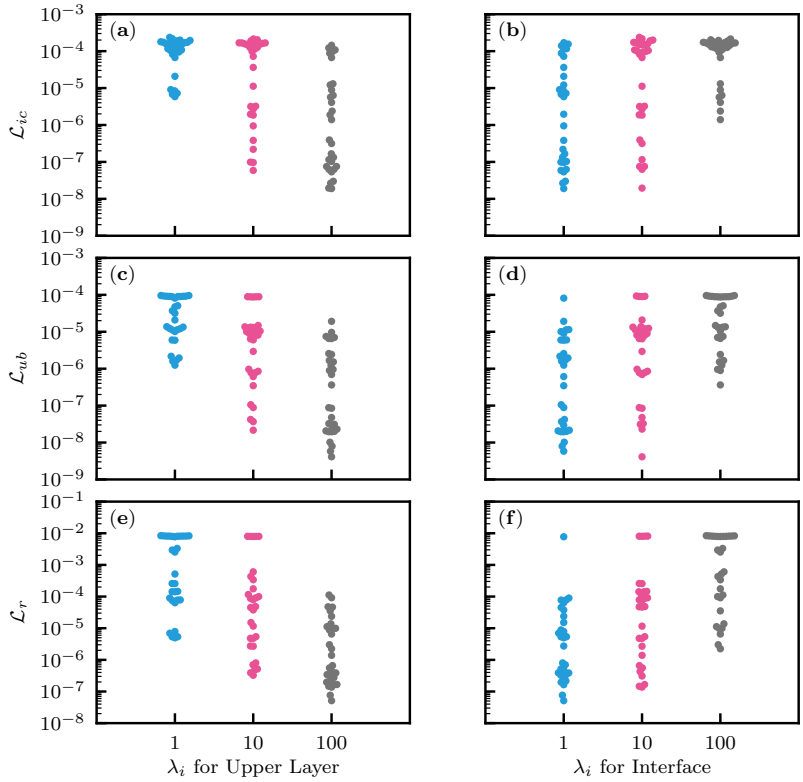

Figure A.4: Heterogeneous soil. The effects of weight parameters λ_i in the loss function on the loss terms corresponding to the upper layer. The left and right columns correspond to the effects of λ_i for the upper layer and interface conditions, respectively. (**a**) and (**b**): Loss term for the initial condition $\mathcal{L}_{ic}$. (**c**) and (**d**): Loss term for the upper boundary condition $\mathcal{L}_{ub}$. (**e**) and (**f**): Loss term for the residual of the PDE $\mathcal{L}_r$.

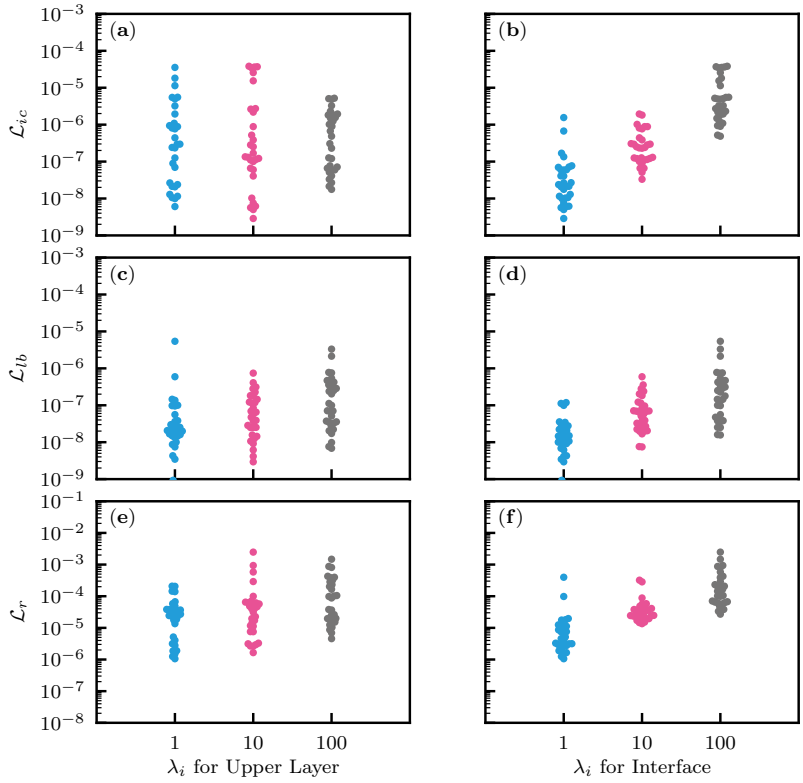

Figure A.5: Heterogeneous soil. The effects of weight parameters λ_i in the loss function on the loss terms corresponding to the lower layer. The left and right columns correspond to the effects of λ_i for the upper layer and interface conditions, respectively. (**a**) and (**b**): Loss term for the initial condition $\mathcal{L}_{ic}$. (**c**) and (**d**): Loss term for the lower boundary condition $\mathcal{L}_{ub}$. (**e**) and (**f**): Loss term for the residual of the PDE $\mathcal{L}_r$.

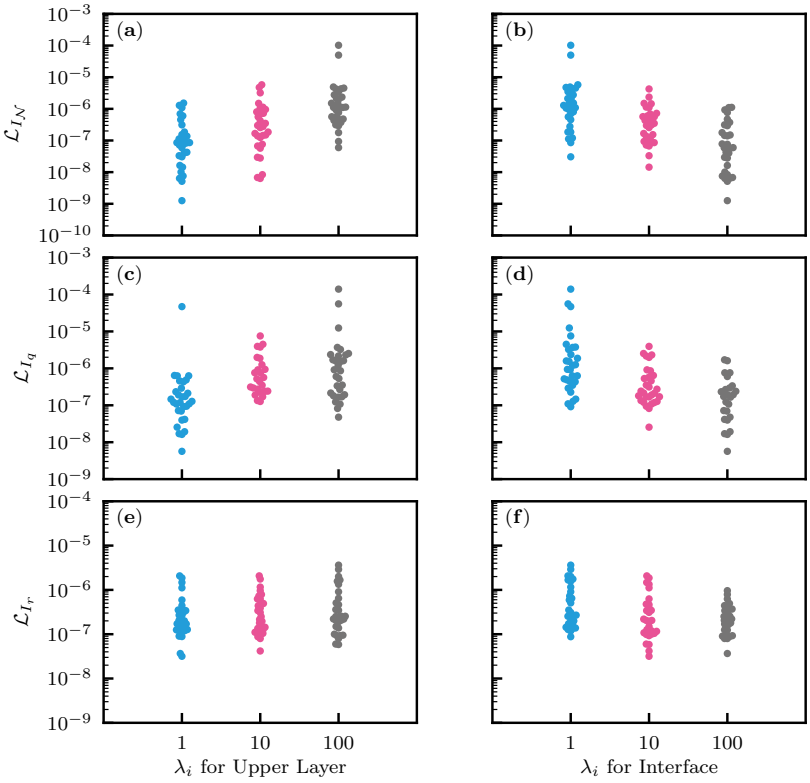

Figure A.6: Heterogeneous soil. The effects of weight parameters λ_i in the loss function on the loss terms corresponding to the interface conditions. The left and right columns correspond to the effects of λ_i for the upper layer and interface conditions, respectively. (**a**) and (**b**): Loss term for the continuity in the neural network output $\mathcal{L}_{I_\mathcal{N}}$. (**c**) and (**d**): Loss term for the continuity in the water flux $\mathcal{L}_{I_q}$. (**e**) and (**f**): Loss term for the continuity in the residual of the PDE $\mathcal{L}_{I_r}$.

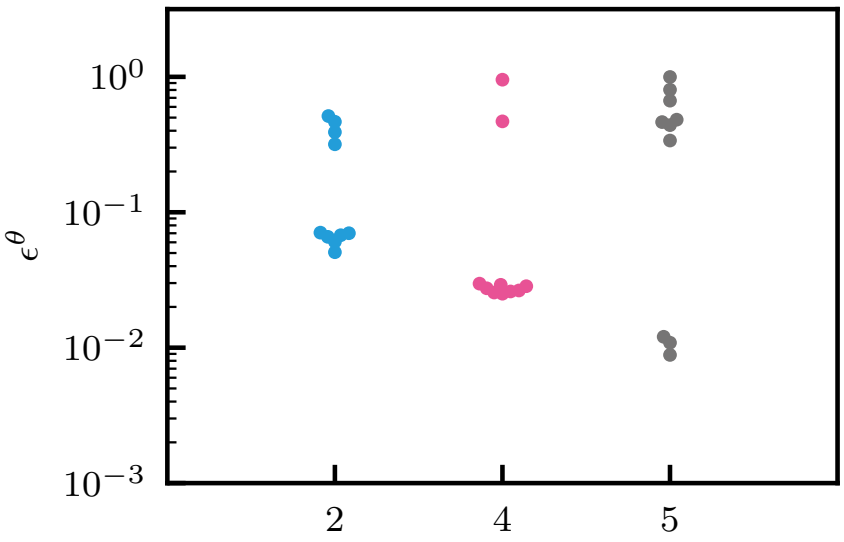

Figure A.7: The relative squared error in terms of volumetric water content ϵ^θ for different numbers of measurement locations.

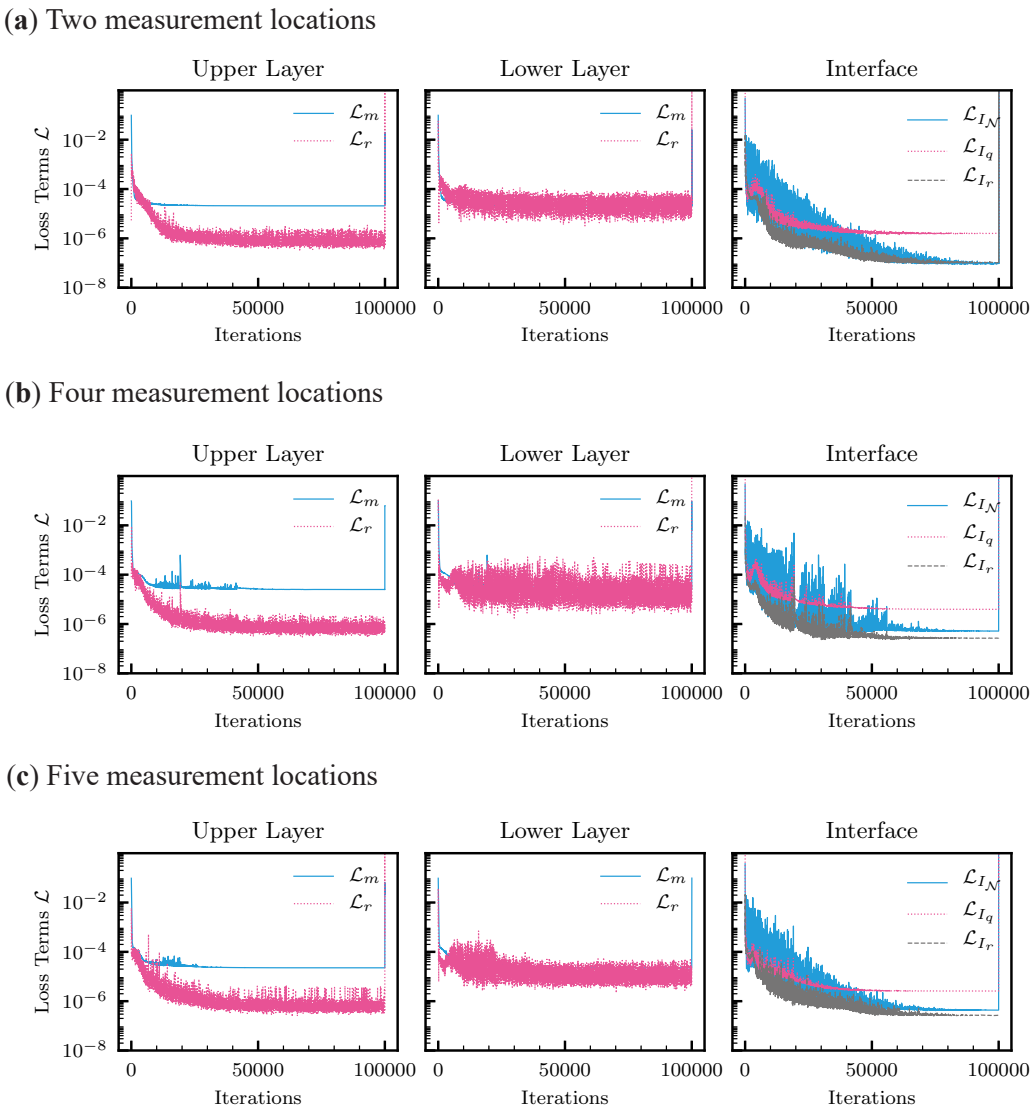

Figure A.8: Inverse modeling to estimate surface water flux from soil moisture measurements in a layered soil (upper layer: loam soil; lower layer: sandy loam soil). The evolution of loss terms for the upper layer (left column), lower layer (center column), and the interface conditions (right column) for different measurement locations z_m [cm]. (**a**): $z_m \in \{-5, -15\}$. (**b**): $z_m \in \{-3, -7, -13, -17\}$. (**c**): $z_m \in \{-1, -5, -9, -13, -17\}$.

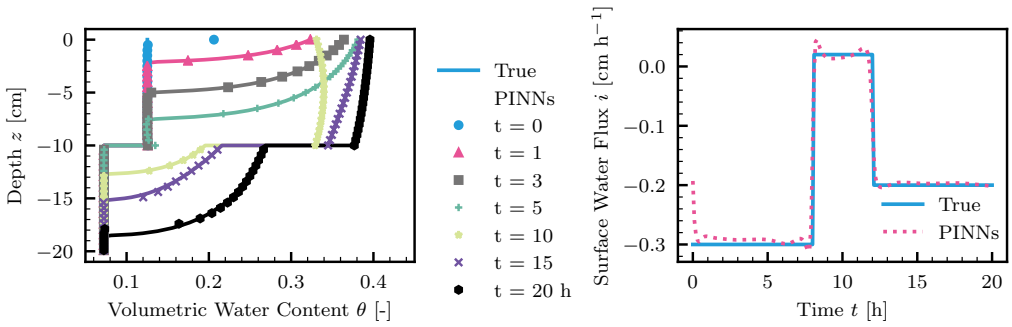

Figure A.9: Inverse modeling to estimate surface flux from five water potential measurements in a layered soil ($z \in -1, -5, -9, -13, -17$ cm). The left figure shows the comparison between the true and PINNs' volumetric water content. The right figure shows the true and estimated surface water flux.

References for Appendix A

Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res.*, 26(7):1483–1496.

LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations*. the Society for Industrial and Applied Mathematics.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Comput. Phys.*, 378:686–707.

Wang, S., Teng, Y., and Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.*, 43(5):A3055–A3081.

Appendix B

Supplementary materials for Chapter 4

B.1 Discretize-then-Optimize approach

In this section, we describe the detail of the DTO approach.

B.1.1 Derivative of forward problem in terms of state

In this section, we describe the derivative of the forward problem with respect to the state $\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}} \in \mathbb{R}^{N_s(N_t+1)\times N_s(N_t+1)}$. We begin with $\mathbf{F}^0$. The initial condition (Eq. 4.30) gives

$$\frac{\partial \mathbf{F}^0}{\partial \boldsymbol{\psi}^0} = \mathbf{I} \quad \text{and} \quad \frac{\partial \mathbf{F}^0}{\partial \boldsymbol{\psi}^n} = \mathbf{0} \quad \text{for} \quad 1 \leq n \leq N_t. \tag{B.1}$$

As for $\mathbf{F}^n$ for $n = 1, 2, ..., N_t$, Equation 4.25 gives

$$\left(\frac{\partial \mathbf{F}^n}{\partial \boldsymbol{\psi}^{n-1}}\right)_{ij} = \frac{\partial F_i^n}{\partial \psi_j^{n-1}} = -\mathbf{M}_{ij}^L \frac{\partial \theta_i^{n-1}}{\partial \psi_j^{n-1}}, \tag{B.2}$$

$$\left(\frac{\partial \mathbf{F}^n}{\partial \boldsymbol{\psi}^n}\right)_{ij} = \frac{\partial F_i^n}{\partial \psi_j^n} = \mathbf{M}_{ij}^L \frac{\partial \theta_i^n}{\partial \psi_j^n} +$$

$$\Delta t \left(K'(\psi_j^n) \int_\Omega \phi_j \frac{\partial \psi_h^n}{\partial z} \frac{\partial \phi_i}{\partial z}\, dz + \int_\Omega K_h \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z}\, dz + K'(\psi_j^n) \int_\Omega \phi_j \frac{\partial \phi_i}{\partial z}\, dz \right), \tag{B.3}$$

$$\left(\frac{\partial \mathbf{F}^n}{\partial \boldsymbol{\psi}^k}\right)_{ij} = \frac{\partial F_i^n}{\partial \psi_j^k} = 0 \quad \text{for} \quad k \neq n-1, n. \tag{B.4}$$

Here, we define a diagonal matrix $\boldsymbol{\Theta}^n$, whose ijth element is $\frac{\partial \theta_i^n}{\partial \psi_j^n}\delta_{ij}$ and a matrix $\mathbf{N}^n$, whose ijth element is

$$N_{ij}^n := K'(\psi_j^n) \int_\Omega \phi_j \frac{\partial \psi_h^n}{\partial z} \frac{\partial \phi_i}{\partial z}\, dz + \int_\Omega K_h \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z}\, dz + K'(\psi_j^n) \int_\Omega \phi_j \frac{\partial \phi_i}{\partial z}\, dz. \tag{B.5}$$

Note that we assume $\boldsymbol{\psi}^n$ is the exact solution to the non-linear problem (Equation 4.25). Then, the Jacobian matrix $\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}$ can be written as a block matrix:

$$\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{M}^L\boldsymbol{\Theta}^0 & \mathbf{M}^L\boldsymbol{\Theta}^1 + \Delta t\mathbf{N}^1 & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}^L\boldsymbol{\Theta}^1 & \mathbf{M}^L\boldsymbol{\Theta}^2 + \Delta t\mathbf{N}^2 & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & \ddots & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & -\mathbf{M}^L\boldsymbol{\Theta}^{N_t-2} & \mathbf{M}^L\boldsymbol{\Theta}^{N_t-1} + \Delta t\mathbf{N}^{N_t-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & -\mathbf{M}^L\boldsymbol{\Theta}^{N_t-1} & \mathbf{M}^L\boldsymbol{\Theta}^{N_t} + \Delta t\mathbf{N}^{N_t} \end{bmatrix}. \tag{B.6}$$

B.1.2 Derivative of data-misfit term with respect to state

This section derives $\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}$, where $D = \frac{1}{2}\int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))^T(\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))\,dt$. By the trapezoital rule, the integral can be approximated as

$$\begin{aligned} & \int_0^T (\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0_{\text{obs}})^T(\mathbf{B}\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{obs}}(t))\,dt \\ \approx\ & \frac{1}{2}\Delta t(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}})^T(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}}) + \Delta t(\mathbf{B}\boldsymbol{\theta}^1 - \boldsymbol{\theta}^1_{\text{obs}})^T(\mathbf{B}\boldsymbol{\theta}^1 - \boldsymbol{\theta}^1_{\text{obs}}) \\ +\ & ... \\ +\ & \Delta t(\mathbf{B}\boldsymbol{\theta}^{n-1} - \boldsymbol{\theta}^{n-1}_{\text{obs}})^T(\mathbf{B}\boldsymbol{\theta}^{n-1} - \boldsymbol{\theta}^{n-1}_{\text{obs}}) + \frac{1}{2}\Delta t(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}})^T(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}}), \end{aligned} \tag{B.7}$$

where $\boldsymbol{\theta}^n_{\text{obs}}$ for $n = 0, 1, ..., N_t$ is a vector consisting of the volumetric water content data collected at the measurement locations. Thus,

$$\frac{\partial D}{\partial \psi_i^n} = 0 \quad \text{for} \quad z_i \notin \{z^1_{\text{obs}}, .., z^{N_{\text{obs}}}_{\text{obs}}\} \quad \text{and} \quad n = 0, ..., N_t, \tag{B.8}$$

$$\frac{\partial D}{\partial \psi_i^n} = \frac{1}{2}\Delta t(\theta_i^n - \theta^n_{\text{obs, i}})\frac{d\theta}{d\psi}(\psi_i^n) \quad \text{for} \quad z_i \in \{z^1_{\text{obs}}, .., z^{N_{\text{obs}}}_{\text{obs}}\} \quad \text{and} \quad n = 0, N_t, \tag{B.9}$$

$$\frac{\partial D}{\partial \psi_i^n} = \Delta t(\theta_i^n - \theta^n_{\text{obs, i}})\frac{d\theta}{d\psi}(\psi_i^n) \quad \text{for} \quad z_i \in \{z^1_{\text{obs}}, .., z^{N_{\text{obs}}}_{\text{obs}}\} \quad \text{and} \quad n = 1, 2, ..., N_t - 1, \tag{B.10}$$

where z_i is the spatial coordinate corresponding to ith node for $i = 1, ..., N_s$, $\theta^n_{\text{obs, i}}$ is the volumetric water content measurement at $z = z_i$. Thus,

$$\left(\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}\right)^T = \Delta t \begin{bmatrix} \frac{1}{2}\boldsymbol{\Theta}^0\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}}) \\ \boldsymbol{\Theta}^1\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^1 - \boldsymbol{\theta}^1_{\text{obs}}) \\ ... \\ \boldsymbol{\Theta}^{N_t-1}\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^{N_t-1} - \boldsymbol{\theta}^{N_t-1}_{\text{obs}}) \\ \frac{1}{2}\boldsymbol{\Theta}^{N_t}\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^{N_t} - \boldsymbol{\theta}^{N_t}_{\text{obs}}) \end{bmatrix}. \tag{B.11}$$

B.1.3 Adjoint problem

The adjoint problem is

$$\left(\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}\right)^T \bar{\mathbf{p}} = -\left(\frac{\partial D}{\partial \bar{\boldsymbol{\psi}}}\right)^T. \tag{B.12}$$

From Eq. B.6 and B.11, the adjoint equations are

$$\mathbf{p}^0 = -\frac{1}{2}\Delta t\boldsymbol{\Theta}^0\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}}) + \mathbf{M}^L\boldsymbol{\Theta}^0\mathbf{p}^1, \quad \text{(B.13)}$$

$$(\mathbf{M}^L\boldsymbol{\Theta}^n + \Delta t(\mathbf{N}^n)^T)\mathbf{p}^n = -\Delta t\boldsymbol{\Theta}^n\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}}) + \mathbf{M}^L\boldsymbol{\Theta}^n\mathbf{p}^{n+1}$$

$$\text{for} \quad n = N_t - 1, ..., 1, \quad \text{(B.14)}$$

$$(\mathbf{M}^L\boldsymbol{\Theta}^{N_t} + \Delta t(\mathbf{N}^{N_t})^T)\mathbf{p}^{N_t} = -\frac{1}{2}\Delta t\boldsymbol{\Theta}^{N_t}\mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^{N_t} - \boldsymbol{\theta}^{N_t}_{\text{obs}}), \quad \text{(B.15)}$$

which are solved backward in time (i.e., from $\mathbf{p}^{N_t}$). By comparing the adjoint equations and the adjoint PDE (Eq. B.1.3), we can recognize that the terminal condition for the OTD approach is $p(z,T) = 0$, while the adjoint variable for the last time step $\mathbf{p}^{N_t}$ is not zero if there is a difference between the estimated and measured volumetric water content at the measurement locations.

B.1.4 Derivative of forward problem in terms of parameter

Next, the derivative of the forward problem with respect to the parameter $\frac{\partial\bar{\mathbf{F}}}{\partial\mathbf{m}} \in \mathbb{R}^{N_s(N_t+1)\times(N_t+1)}$ is described. The initial condition (Equation 4.30) gives

$$\frac{\partial\mathbf{F}^0}{\partial\mathbf{m}} = \mathbf{0}. \quad \text{(B.16)}$$

As for $\mathbf{F}^n$ for $n = 1, 2, ..., N_t$, Equation 4.25 gives

$$\left(\frac{\partial\mathbf{F}^n}{\partial m_n}\right)_i = -\Delta t\int_{\partial\Omega_F}\phi_i\,ds, \quad \text{(B.17)}$$

$$\left(\frac{\partial\mathbf{F}^n}{\partial m_k}\right)_i = 0 \quad \text{for} \quad k \neq n. \quad \text{(B.18)}$$

We define a vector $\mathbf{b}$, where its ith element b_i is

$$b_i = \int_{\partial\Omega_F}\phi_i\,ds. \quad \text{(B.19)}$$

Then, the Jacobian matrix $\frac{\partial\bar{\mathbf{F}}}{\partial\mathbf{m}}$ can be written as a block matrix:

$$\frac{\partial\bar{\mathbf{F}}}{\partial\mathbf{m}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\Delta t\mathbf{b} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\Delta t\mathbf{b} & ... & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & \ddots & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & -\Delta t\mathbf{b} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} & -\Delta t\mathbf{b} \end{bmatrix}. \quad \text{(B.20)}$$

B.1.5 Incremental forward problem

The incremental forward problem is

$$\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\boldsymbol{\psi}}}\bar{\tilde{\boldsymbol{\psi}}} = -\frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{m}}\tilde{\mathbf{m}}, \tag{B.21}$$

and from Eq. B.6 and B.20, the incremental forward equations are

$$\tilde{\boldsymbol{\psi}}^0 = \mathbf{0}, \tag{B.22}$$

$$(\mathbf{M}^L\boldsymbol{\Theta}^n + \Delta t^n \mathbf{N}^n)\tilde{\boldsymbol{\psi}}^n = \Delta t \tilde{m}_n \mathbf{b} + \mathbf{M}^L\boldsymbol{\Theta}^{n-1}\tilde{\boldsymbol{\psi}}^{n-1} \quad \text{for} \quad n = 1, ..., N_t, \tag{B.23}$$

where $\tilde{m}_i$ is the ith element of the vector $\tilde{\mathbf{m}}$.

B.1.6 Derivative of adjoint problem

$\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}$ is the derivative of the left-hand side of the adjoint problem (Eq. 4.67). Because the adjoint equation for each time step only depends on the state variable on the same time step, the matrix $\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}$ is a diagonal block matrix, whose nth ($n = 0, 1, ..., N_t$) main diagonal element is denoted by $\left(\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)_n$, and then

$$\left(\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)_0 = -\mathbf{M}^L\underline{\boldsymbol{\Theta}}^0\mathbf{p}^1, \tag{B.24}$$

$$\left(\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)_n = \mathbf{M}^L\underline{\boldsymbol{\Theta}}^n(\mathbf{p}^n - \mathbf{p}^{n+1}) + \Delta t\mathbf{X}^n \quad \text{for} \quad n = N_t - 1, ..., 1, \tag{B.25}$$

$$\left(\frac{\partial \mathbf{A}^T\bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)_{N_t} = \mathbf{M}^L\underline{\boldsymbol{\Theta}}^{N_t}\mathbf{p}^{N_t} + \Delta t\mathbf{X}^n \quad \text{for} \quad n = N_t, \tag{B.26}$$

where $\underline{\boldsymbol{\Theta}}^n$ is a diagonal matrix, whose ith main diagonal element is $\frac{d^2\theta}{d\psi^2}(\psi_i^n)$, $\mathbf{X}^n := \frac{\partial(\mathbf{N}^n)^T\mathbf{p}^n}{\partial \boldsymbol{\psi}^n}$. The ijth element of the matrix $\mathbf{X}^n$ is

$$\begin{aligned}
&(\mathbf{X})_{ij} = \frac{\partial((\mathbf{N}^n)^T\mathbf{p}^n)_i}{\partial \psi_j^n} = \frac{\partial \sum_k((\mathbf{N}^n)^T)_{ik}p_k^n}{\partial \psi_j^n} \\
= &\frac{\partial \sum_k(\mathbf{N}^n)_{ki}p_k^n}{\partial \psi_j^n} \\
= &\frac{\partial\left(\sum_k p_k^n K'(\psi_i^n)\int_\Omega \phi_i \frac{\partial \psi_h^n}{\partial z}\frac{\partial \phi_k}{\partial z}\,dz + p_k^n\int_\Omega K_h\frac{\partial \phi_i}{\partial z}\frac{\partial \phi_k}{\partial z}\,dz + p_k^n K'(\psi_i^n)\int_\Omega \phi_i\frac{\partial \phi_k}{\partial z}\,dz\right)}{\partial \psi_j^n} \\
= &\frac{\partial\left(K'(\psi_i^n)\int_\Omega \phi_i \frac{\partial \psi_h^n}{\partial z}\frac{\partial p_h^n}{\partial z}\,dz + \int_\Omega K_h\frac{\partial \phi_i}{\partial z}\frac{\partial p_h^n}{\partial z}\,dz + K'(\psi_i^n)\int_\Omega \phi_i\frac{\partial p_h^n}{\partial z}\,dz\right)}{\partial \psi_j^n} \\
= &K''(\psi_i^n)\delta_{ij}\int_\Omega \phi_i\frac{\partial \psi_h^n}{\partial z}\frac{\partial p_h^n}{\partial z}\,dz + K'(\psi_i^n)\int_\Omega \phi_i\frac{\partial \phi_j}{\partial z}\frac{\partial p_h^n}{\partial z}\,dz \\
+ &K'(\psi_j^n)\int_\Omega \phi_j\frac{\partial \phi_i}{\partial z}\frac{\partial p_h^n}{\partial z}\,dz + K''(\psi_i^n)\delta_{ij}\int_\Omega \phi_i\frac{\partial p_h^n}{\partial z}\,dz.
\end{aligned} \tag{B.27}$$

B.1.7 Second derivative of data-misfit term with respect to state

We need to compute the second derivative of data-misfit term with respect to state $\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}$, which is a block diagonal matrix, whose nth main diagonal element ($n = 0, 1, ..., N_t$) is denoted by $\left(\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}\right)_n$, and then

$$\left(\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}\right)_n = \frac{1}{2}\Delta t \left(\underline{\boldsymbol{\Theta}}^n \mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}}) + \boldsymbol{\Theta}^n \mathbf{B}^T \mathbf{B} \boldsymbol{\Theta}^n\right)$$
$$\text{for} \quad n = 0, N_t, \tag{B.28}$$
$$\left(\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}\right)_n = \Delta t \left(\underline{\boldsymbol{\Theta}}^n \mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}}) + \boldsymbol{\Theta}^n \mathbf{B}^T \mathbf{B} \boldsymbol{\Theta}^n\right)$$
$$\text{for} \quad n = 1, ..., N_t - 1. \tag{B.29}$$

B.1.8 Incremental adjoint problem

The incremental adjoint problem is

$$\mathbf{A}^T \bar{\tilde{\mathbf{p}}} = -\left(\left(\frac{\partial \mathbf{A}^T \bar{\mathbf{p}}}{\partial \bar{\boldsymbol{\psi}}}\right)^T + \left(\frac{\partial \mathbf{d}}{\partial \bar{\boldsymbol{\psi}}}\right)^T\right) \bar{\tilde{\boldsymbol{\psi}}}, \tag{B.30}$$

and from Eq. B.24 and B.28 the incremental adjoint equations are

$$\tilde{\mathbf{p}}^0 = -\left(-\mathbf{M}^L \underline{\boldsymbol{\Theta}}^0 \mathbf{p}^1 + \frac{1}{2}\Delta t \left(\underline{\boldsymbol{\Theta}}^0 \mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}}) + \boldsymbol{\Theta}^0 \mathbf{B}^T \mathbf{B} \boldsymbol{\Theta}^0\right)\right) \tilde{\boldsymbol{\psi}}^{\mathbf{0}} + \mathbf{M}^L \boldsymbol{\Theta}^0 \tilde{\mathbf{p}}^1, \tag{B.31}$$
$$(\mathbf{M}^L \boldsymbol{\Theta}^n + \Delta t (\mathbf{N}^n)^T) \tilde{\mathbf{p}}^n =$$
$$-\left(\mathbf{M}^L \underline{\boldsymbol{\Theta}}^n (\mathbf{p}^n - \mathbf{p}^{n+1}) + \Delta t \, (\mathbf{X}^n)^T + \Delta t \left(\underline{\boldsymbol{\Theta}}^n \mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^n - \boldsymbol{\theta}^n_{\text{obs}}) + \boldsymbol{\Theta}^n \mathbf{B}^T \mathbf{B} \boldsymbol{\Theta}^n\right)\right) \tilde{\boldsymbol{\psi}}^{\boldsymbol{n}}$$
$$+\mathbf{M}^L \boldsymbol{\Theta}^n \tilde{\mathbf{p}}^{n+1} \quad \text{for} \quad n = 1, ..., N_t - 1, \tag{B.32}$$
$$(\mathbf{M}^L \boldsymbol{\Theta}^{N_t} + \Delta t (\mathbf{N}^{N_t})^T) \tilde{\mathbf{p}}^{N_t} =$$
$$-\left(\mathbf{M}^L \underline{\boldsymbol{\Theta}}^{N_t} \mathbf{p}^{N_t} + \Delta t \left(\mathbf{X}^{N_t}\right)^T + \frac{1}{2}\Delta t \left(\underline{\boldsymbol{\Theta}}^n \mathbf{B}^T(\mathbf{B}\boldsymbol{\theta}^0 - \boldsymbol{\theta}^0_{\text{obs}}) + \boldsymbol{\Theta}^n \mathbf{B}^T \mathbf{B} \boldsymbol{\Theta}^n\right)\right) \tilde{\boldsymbol{\psi}}^{\boldsymbol{N_t}}, \tag{B.33}$$

which are solved backward in time (i.e., from the last row). By comparing the incremental adjoint equations and the adjoint PDE (Eq. 4.74), we can recognize that the terminal condition for the OTD approach is $\tilde{p}(z, T) = 0$, while the incremental adjoint variable for the last time step $\tilde{\mathbf{p}}^{N_t}$ is not zero.