

# A Novel Data-driven Numerical Method for Hydrological Modeling of Water Infiltration in Porous Media

Zeyuan Song, Zheyu Jiang  
 School of Chemical Engineering  
 Oklahoma State University  
 Stillwater, OK 74074  
 {taekwon.song,zjiang}@okstate.edu

## Abstract

Root-zone soil moisture monitoring is essential for sensor-based smart irrigation and agricultural drought prevention. Modeling the spatiotemporal water flow dynamics in porous media such as soil is typically achieved by solving an agro-hydrological model, the most important of which being the Richards equation. In this paper, we present a novel data-driven solution algorithm named the DRW (Data-driven global Random Walk) algorithm, which holistically integrates adaptive linearization scheme, neural networks, and global random walk in a finite volume discretization framework. We discuss the need and benefits of introducing these components to achieve synergistic improvements in solution accuracy and numerical stability. We show that the DRW algorithm can accurately solve  $n$ -dimensional Richards equation with guaranteed convergence under reasonable assumptions. Through examples, we also demonstrate that the DRW algorithm can better preserve the underlying physics and mass conservation of the Richards equation compared to state-of-the-art solution algorithms and commercial solver.

*Keywords:* Soil moisture; sustainable agriculture; hydrological modeling; global random walk; neural network

## 1 Introduction

With increasing demand for food and the resulting food-energy-water nexus challenges from the growing population, there is an increasing interest among chemical engineering researchers, especially those in process systems engineering (PSE), to design the next-generation food and agricultural systems that are sustainable, resource-efficient, and resilient. In particular, the development of new PSE tools that provide “sustainable engineering solutions for food and water” is prominently featured by the National Academies as one of the “key research priorities and new directions in chemical engineering” [1].

Along this line, sensor-based digital agriculture for sustainable and efficient use of water by leveraging real-time soil monitoring is essential for improving agricultural production and crop productivity, providing basis for precision irrigation and agriculture, preventing leaching of agrochemicals and soil nutrients into groundwater, and predicting agricultural droughts [2]. Recent studies reveal that adjusting irrigation activities based on root-zone soil moisture information can reduce irrigation water consumption by 40-60% [3] and increase farmer’s revenue by 20-60% [4]. Modeling the spatiotemporal behavior of root zone soil moisture from precipitation and surface soil moisture data is typically achieved by solving an agro-hydrological model that describes water movement through unsaturated soils. Nowadays, most existing agro-hydrological models are based

on the Richards equation [5], which captures irrigation, precipitation, evapotranspiration, runoff, and drainage dynamics in soil:

$$\begin{aligned}\partial_t \theta(\psi) + \nabla \cdot \mathbf{q} &= -S(\psi), \\ \mathbf{q} &= -K(\theta(\psi)) \nabla(\psi + z),\end{aligned}\tag{1}$$

where  $\psi$  stands for pressure head (in, e.g., m),  $\mathbf{q}$  represents the water flux (in, e.g.,  $\text{m}^3/\text{m}^2 \cdot \text{s}$ ),  $S$  is the sink term associated with root water uptake (in, e.g.,  $\text{s}^{-1}$ ),  $\theta$  denotes the soil moisture content (in, e.g.,  $\text{m}^3/\text{m}^3$ ),  $K$  is unsaturated hydraulic water conductivity (in, e.g.,  $\text{m}/\text{s}$ ),  $t \in [0, T]$  denotes the time (in, e.g., s), and  $z$  corresponds to the vertical depth (in, e.g., m). The Richards equation is a nonlinear convection-diffusion equation [6], in which the convection term is due to gravity and the diffusive term comes from Darcy's law [7]. For unsaturated flow, both  $\theta$  and  $K$  are highly nonlinear functions of pressure head  $\psi$  and soil properties, making Equation (1) challenging to solve. Specifically,  $\theta(\psi)$  and  $K(\psi)$  (or  $K(\theta)$ , depending on the model) are commonly referred to as the water retention curve (WRC) and hydraulic conductivity function (HCF), respectively. Several empirical models have been developed for WRC and HCF for major soil types, among which some of the widely adopted ones are summarized in Table 1.

Model	HCF ( $K(\psi)$ or $K(\theta)$ )	WRC ( $\theta(\psi)$ )
Haverkamp [8]	$K_s \frac{A}{A +  \psi ^\gamma}$	$\theta_r + \frac{\alpha(\theta_s - \theta_r)}{\alpha +  \psi ^\beta}$
Mualem-van Genuchten [9, 10]	$K_s \sqrt{\frac{\theta - \theta_r}{\theta_s - \theta_r}} \left\{ 1 - \left[ 1 - \left( \frac{\theta - \theta_r}{\theta_s - \theta_r} \right)^{\frac{l}{l-1}} \right]^{\frac{l-1}{l}} \right\}^2$	$\theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha \psi )^n]^{\frac{n-1}{n}}}$
Gardner [11]	$K_s e^{\alpha\psi}$	$\theta_r + (\theta_s - \theta_r) e^{\alpha\psi}$

Table 1: Some of the widely used HCF and WRC models. In these models,  $A$ ,  $\gamma$ ,  $\alpha$ ,  $\beta$ ,  $n$ ,  $\theta_s$ , and  $\theta_r$  are soil-specific parameters.

Due to the high nonlinearity of WRC and HCF, analytical solutions to the Richards equation do not exist in general [12]. Thus, the Richards equation is typically solved by numerically, which almost always requires some form of discretization. Consider the discretized version of Equation (1), whose control volume  $V \subset \mathbb{R}^d$  ( $d = 1, 2, 3$ ) is discretized into  $N$  small cells  $V_1, \dots, V_N$ . Using implicit Euler method on the time domain with a time step size of  $\Delta t$ , the discretized Richards equation at time step  $m = 0, 1, \dots, \lceil \frac{T}{\Delta t} \rceil - 1$  can be expressed as:

$$\begin{cases} \theta(\psi_i^{m+1}) - \theta(\psi_i^m) - \Delta t \nabla \cdot \left[ K(\theta(\psi_i^{m+1})) \nabla(\psi_i^{m+1} + z) \right] + S(\psi_i^{m+1}) = 0, \\ \text{Dirichlet boundary condition: } \psi_j(\cdot) = 0 \text{ for all } V_j \subset \partial V, \\ \text{Initial condition: } \psi(0, \cdot) = \psi_0(\cdot), \end{cases}\tag{2}$$

where  $\psi_i^m$  is the pressure head in cell  $V_i$  and time step  $m$ , and  $\psi_0(\cdot)$  denotes the initial condition at  $t = 0$ .

The performance of a numerical partial differential equation (PDE) solver depends theoretically on the well-posedness of the PDE [13], which is an essential property that certifies the accuracy and reliability of numerical solutions to the PDE. A PDE is said to be well-posed if its weak solution exists, is unique, and depends continuously on the problem's initial conditions [13, 14]. The weak solution of Equation (2) is formally defined as:

**Definition 1.1.** Given  $\psi_i^m \in H_0^1(V)$ , if for any  $v \in H_0^1(V)$  and  $v(T, \cdot) = 0$ ,

$$\langle (\theta(\psi^{m+1}) - \theta(\psi^m)), v \rangle_{V_i} + \Delta t \langle K(\theta(\psi^{m+1})) \nabla(\psi^{m+1} + z), \nabla v \rangle_{V_i} + \langle S(\psi^{m+1}), v \rangle_{V_i} = 0\tag{3}$$

holds, then  $\psi_i^{m+1}$  is a weak solution of the discretized Richards equation.

Note that in Definition 1.1, an inner product  $\langle \cdot, \cdot \rangle_{V_i} : L^2([0, T], H_0^1(V)) \rightarrow L^2([0, T], H_0^1(V))$  is defined as  $\langle f, g \rangle_{V_i} := \int_{V_i} fg \, dV$ . When  $V_i$  is sufficiently small,  $\int_{V_i} fg \, dV = (fg)_i \text{vol}(V_i)$ , in which  $(fg)_i$  denotes the value of  $fg$  evaluated at cell  $V_i$ , and  $\|f\|^2 := \langle f, f \rangle_{V_i} = (f^2)_i \text{vol}(V_i)$ . In particular, we point out that Equation (3) is originated from Equation (2) as:

$$\begin{aligned} \langle \nabla \cdot [K(\Theta(\psi^{m+1})) \nabla (\psi^{m+1} + z)], v \rangle_{V_i} &= \int_{V_i} \nabla \cdot [K(\Theta(\psi^{m+1})) \nabla (\psi^{m+1} + z)] v \, dV \\ &= \int_{\partial V_i} K(\Theta(\psi^{m+1})) \nabla (\psi^{m+1} + z) v \, dS - \int_{V_i} K(\Theta(\psi^{m+1})) \nabla (\psi^{m+1} + z) \nabla v \, dV \\ &= -\langle K(\Theta(\psi^{m+1})) \nabla (\psi^{m+1} + z), \nabla v \rangle_{V_i}, \end{aligned}$$

in which we use integration by parts for higher dimensions and the fact that the surface integral over  $\partial V_i$  is 0. We remark that, in the context of the Richards equation, the existence and uniqueness of its weak solution have been rigorously established and carefully studied [15, 16, 17], laying the theoretical foundation for developing an efficient solution algorithm to solve the discretized Richards equation numerically.

## 2 Literature Review

Among existing solution algorithms for the Richards equation, methods based on finite difference and finite element discretizations [18, 19] have been studied and implemented the most [20]. However, these methods often face challenges when handling large-scale problems and suffer from instability issues such as oscillations [21]. Recently, Ireson et al. [22] used the method of lines to convert the 1-D Richards equation into an ordinary differential equation (ODE), which was then solved by finite difference method. Despite these advancements, finite difference- and finite element-based methods generally require high mesh resolution to satisfy the local equilibrium condition [23, 24, 25]. Furthermore, they tend to fail to preserve global mass balance [26] and other important underlying physical relations connecting soil moisture, pressure head, and water flux [27, 28], which further deteriorates solution accuracy.

Meanwhile, finite volume discretization method (FVM) has demonstrated promising potential in achieving high solution accuracy and preserving the mass conservation when solving the Richards equation [29]. For example, Lai and Ogden [30] obtained a family of mass-conservative finite volume predictor-corrector solutions for the 1-D Richards equation. A second-order accurate monotone FVM was also proposed by Misiats and Lipnikov [16] for solving the 1-D Richards equation. Several other FVM-based numerical methods have also been developed to solve 1-D and 2-D Richards equations [31, 32, 33]. However, like finite difference- and finite element-based methods, conventional FVM typically converts the discretized Richards equation into a large, stiff matrix equation, which can be challenging to solve.

Thus, instead of following the standard practice of converting the discretized equations into a matrix equation, methods such as the linearization scheme attempt to solve the discretized Richards equations iteratively. A few notable linearization scheme implementations include Bergamaschi and Putti [34] and Pop et al. [35]. In standard linearization schemes, a fixed linearization parameter  $L$  is used for all time steps and discretized cells. However, choosing an appropriate static value of  $L$  is not straightforward, as soil moisture content and pressure head exhibit strong spatiotemporal patterns. To address this drawback, Mitra and Pop [36] modified the standard linearization scheme for solving 1-D nonlinear diffusion equations by allowing  $L$  to be adaptive with respect to space

and time. More recently, Albuja and Avila [37] modified the Newton-type scheme and proposed a new linearization scheme for solving the 1-D Richards equation with guaranteed global convergence. Nevertheless, some of the existing adaptive linearization schemes have been reported to be suffered from numerical oscillations [38]. And so far, no adaptive linearization scheme has been established for solving the 3-D Richards equation.

On the other hand, as a promising approach to enhance the accuracy and stability of numerical algorithms, stochastic methods such as the random walk model have been introduced and incorporated in discretization-based methods. In the random walk model, particles, in this case water molecules, can move to their neighboring discretized cells following certain probabilities. Furthermore, random walks are memoryless, meaning that the current movements of particles are independent of their past movements. These properties makes the random walk model attractive for solving various transport problems [39], including the Richards equation (e.g., [40]). Having said that, in conventional random walk model, only one particle can move at a time. To overcome this, the global random walk (GRW) model was recently proposed [41, 42], allowing all particles to move and/or stay at the same time. The GRW model has been successfully implemented to model groundwater transport [41, 43], diffusion process [44, 43], and 1-D and 2-D Richards equation [42]. Furthermore, the GRW model has been recently coupled with the linearization scheme for the first time [42], despite that a static constant  $L$  was adopted. Nevertheless, a key assumption commonly used in state-of-the-art GRW models for the Richards equation is that the pressure head in each discretized cell at any time step is proportional to the number of particles in the same cell and time step, which is yet to be validated. Furthermore, successful implementation of GRW model to the 3-D Richards equation has not be explored in the literature.

Leveraging these new advancements, we recently proposed a novel solution algorithm, which we named as the data-driven global random walk (DRW) algorithm, for 1-D and 2-D Richards equations [27, 28]. Our DRW algorithm outperforms state-of-the-art Richards equation solvers for in terms of solution accuracy and the ability to capture underlying physics. In particular, we find that the use of neural networks to explicitly characterize the relationship between pressure head and the number of water molecules in the DRW framework significantly enhances the solution accuracy. In this article, we further improve the numerical accuracy and computational efficiency of our DRW algorithmic framework by introducing several innovative approaches. Specifically, we will formally present our DRW algorithm for solving any  $d$ -dimensional Richards equation, provide rigorous theoretical justification of the convergence behavior of our DRW algorithm, and conduct systematic case studies and in-depth analyses involving 1- through 3-D problems. Some of the new contributions to the previously established DRW framework are highlighted as follows:

- We generalize our prior works [28, 27] for any  $d$ -dimensional ( $d = 1, 2, 3$ ) Richards equation, and demonstrate that the our new DRW algorithm can be versatily adopted to modeling different realistic scenarios (e.g., layered soil, actual precipitation).
- We introduce a “coarse-to-fine” approach to enhance the solution accuracy of our DRW algorithm without requiring a large amount of high-accuracy, fine-mesh training data. We demonstrate that this coarse-to-fine approach maintains a good balance between computational efficiency and solution accuracy.
- We show that, by synergistically integrating our novel adaptive linearization scheme, global random walk, and neural network, our new DRW algorithmic framework significantly enhances the performance of FVM-based solver in preserving the underlying physical relationships and mass conservation associated with the Richards equation.

We organize the subsequent sections of the paper as follows. In Section 3, we derive the FVM-based adaptive linearization scheme formulation and prove its global convergence. Then, in Section 4, we incorporate the adaptive linearization scheme formulation in the DRW algorithm. To compare the performance of our DRW algorithm with state-of-the-art solvers, we conduct case studies and discuss key findings in Section 5. Finally, we summarize the results and discuss future directions in Section 7.

### 3 Adaptive Linearization Scheme of Discretized Richards Equation

In this section, we will formally introduce the adaptive linearization scheme formulation of the Richards equation discretized by FVM. We will also derive necessary conditions for the linearization parameter to ensure convergence. Furthermore, we will analyze the convergence behavior of the resulting sequence of solutions  $\{\psi_i^{m+1,s}\}_s$ , where  $s$  is the iteration count.

#### 3.1 Adaptive linearization scheme for the Richards Equation

We present the key steps involved in the discretization of the Richards equation using FVM, followed by introducing adaptive linearization scheme to solve the discretized equation iteratively. First, integrating both sides of Equation (1) over  $V$  gives:

$$\int_V [\partial_t \theta(\psi) + S(\psi)] dV = \int_V \nabla \cdot [K(\theta) \nabla(\psi + z)] dV. \quad (4)$$

Then, we apply the divergence theorem to Equation (4), which converts the volume integral on the RHS into a surface integral:

$$[\partial_t \theta(\psi) + S(\psi)] \text{vol}(V) = \oint_{S_V} K(\theta) \nabla(\psi + z) \cdot \mathbf{n} dS_V, \quad (5)$$

where  $\text{vol}(V)$  is the volume of  $V$ ,  $S_V$  is the surface of  $V$  and  $\mathbf{n}$  is the outward pointing unit normal to the boundary  $\partial V$ . The common surface shared by cell  $V_i$  and cell  $V_j$  is denoted as  $\omega_{i,j}$ . With this, we can rewrite the operator  $K(\cdot) \nabla(\cdot)$  and the outward pointing unit normal vector  $\mathbf{n}$  on  $\omega_{i,j}$  as  $[K(\cdot) \nabla(\cdot)]_{\omega_{i,j}}$  and  $\mathbf{n}_{\omega_{i,j}}$ , respectively. With this, the spatial discretization of Equation (5) is expressed as:

$$\partial_t \theta_i \text{vol}(V_i) + S(\psi_i) \text{vol}(V_i) = \sum_{j \in \mathcal{N}_i} [K(\theta) \nabla(\psi + z)]_{\omega_{i,j}} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} \quad \forall i = 1, \dots, N, \quad (6)$$

where  $\partial_t \theta_i$  refers to the time derivative  $\partial_t \theta(\psi_i)$  in cell  $V_i$ ,  $\mathcal{N}_i$  denotes the index set of all the neighboring cells sharing a common surface with  $V_i$ , and  $A_{\omega_{i,j}}$  is the area of surface  $\omega_{i,j}$ .

In standard linearization scheme, for every cell  $V_i$  and time step  $m + 1$ , one would add the term  $L(\psi_i^{m+1,s+1} - \psi_i^{m+1,s})$  to either side of Equation (6), so that the Richards equation can be solved in an iterative manner and the pressure head solution  $\psi_i^m$  is the fixed-point solution of this iterative procedure. In standard linearization schemes, in which  $L$  is a static constant, a trial-and-error procedure is typically required to obtain an appropriate  $L$  value that avoids convergence issues. Not only is this search procedure tedious to implement, the solutions obtained are also less accurate most of the time as we will show in Section 5.1. Thus, inspired by previous works [36, 37], we proposed a novel adaptive linearization scheme that replaces the static  $L$  with  $L_i^{m+1,s}$ , which

adjusts itself for each specific discretized cell, time step, and iteration count. We then introduce the term  $L_i^{m+1,s}(\psi_i^{m+1,s+1} - \psi_i^{m+1,s})$  to the LHS of Equation (6), which leads to:

$$\begin{aligned} \psi_i^{m+1,s+1} = & \psi_i^{m+1,s} + \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} [K(\theta) \nabla(\psi + z)]_{\omega_{i,j}}^{m+1,s} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} \\ & - \frac{1}{L_i^{m+1,s}} \left[ \partial_t \theta_i^{m+1,s} + S(\psi_i^{m+1,s}) \right] \text{vol}(V_i), \end{aligned} \quad (7)$$

By discretizing  $\partial_t \theta_i^{m+1,s}$  using implicit Euler scheme as  $\frac{\theta(\psi_i^{m+1,s}) - \theta(\psi_i^m)}{\Delta t}$ , we can obtain the adaptive linearization scheme of the Richards equation that is discretized by FVM:

$$\begin{aligned} \psi_i^{m+1,s+1} = & \psi_i^{m+1,s} + \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \frac{(\psi + z)_j^{m+1,s} - (\psi + z)_i^{m+1,s}}{\text{dist}(V_j, V_i)} \mathbf{e} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} \\ & - \frac{1}{L_i^{m+1,s}} \left[ \frac{\theta(\psi_i^{m+1,s}) - \theta(\psi_i^m)}{\Delta t} + S(\psi_i^{m+1,s}) \right] \text{vol}(V_i), \end{aligned} \quad (8)$$

where  $\mathbf{e} = (1, 1, 1)$  for the standard 3-D Cartesian coordinate system, and  $\text{dist}(\cdot, \cdot)$  represents the Euclidean distance function.

### 3.2 Choice of Adaptive Linearization Parameter

In adaptive linearization scheme, one needs to automatically select an appropriate linearization parameter. We observe that  $L_i^{m+1,s}$  needs to be sufficiently large because otherwise, the RHS of Equation (8) could approach infinity, which affects the convergence of adaptive linearization scheme. In addition, the choice of  $L_i^{m+1,s}$  can impact the accuracy of solutions. Considering these aspects, we propose a new way of selecting appropriate linearization parameter. First, to prevent  $L_i^{m+1,s}$  from being too close to 0, we impose a user-specified global lower bound  $L_0$ :

$$L_i^{m+1,s} \geq L_0.$$

In addition, we must ensure that  $\frac{\|\psi_i^{m+1,s+1} - \psi_i^{m+1,s}\|}{\|\psi_i^{m+1,s}\|}$  is no greater than a prespecified tolerance  $\rho$ . In other words, we have:

$$\frac{\|\psi_i^{m+1,s+1} - \psi_i^{m+1,s}\|}{\|\psi_i^{m+1,s}\|} = \frac{\frac{1}{L_i^{m+1,s}} \|g_i^{m+1,s}\|}{\|\psi_i^{m+1,s}\|} \leq \rho \quad \forall s = 1, \dots, S,$$

where  $S$  is the user-specified total number of iterations for convergence,  $\rho$  should be no less than the overall tolerance of convergence  $\epsilon$  (to be discussed in Section 3.3), and:

$$\begin{aligned} g_i^{m+1,s} = & \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \frac{(\psi + z)_j^{m+1,s} - (\psi + z)_i^{m+1,s}}{\text{dist}(V_j, V_i)} \mathbf{e} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} \\ & - \frac{\theta(\psi_i^{m+1,s}) - \theta(\psi_i^m)}{\Delta t} \text{vol}(V_i) - S(\psi_i^{m+1,s}) \text{vol}(V_i). \end{aligned}$$

This implies that:

$$L_i^{m+1,s} \geq \frac{(1 + \rho) \|g_i^{m+1,s}\|}{\rho \|\psi_i^{m+1,s}\|} \quad \forall s = 1, \dots, S, \quad (9)$$

whose RHS can be explicitly determined from the results of the previous iteration. In actual implementation, we choose  $L_i^{m+1,s}$  based on:

$$L_i^{m+1,s} = \max \left\{ L_0, \frac{(1 + \rho) \|g_i^{m+1,s}\|}{\rho \|\psi_i^{m+1,s}\|} \right\} \quad \forall s = 1, \dots, S. \quad (10)$$

While Equation (10) gives an automated way to choose the linearization parameter, in the meantime, we need to monitor the sensitivity of solutions obtained by adaptive linearization scheme and make sure that the solutions do not change drastically with respect to small perturbations. To do this, following Zarba [45] and Celia and Zarba [46], we explicitly write down Equation (8) for all discretized cells in the form of a matrix equation:

$$\mathbf{A} \mathbf{x}^{m+1,s} = \mathbf{b}, \quad (11)$$

where the  $i$ th element of vector  $\mathbf{x}^{m+1,s}$  is  $x_i^{m+1,s} = \psi_i^{m+1,s+1} - \psi_i^{m+1,s}$ , which corresponds to cell  $V_i$ . To evaluate the choice of  $L_i^{m+1,s}$ , we calculate the condition number of  $\mathbf{A}$  based on the chosen  $L_i^{m+1,s}$ . If the condition number is larger than a user-specified threshold, we will update  $L_0$  in Equation (10) so that the condition number drops below the threshold. For 1-D problems, Zarba [45] showed that  $\mathbf{A}$  is a  $N \times N$  asymmetric tridiagonal matrix. In this case, the condition number of  $\mathbf{A}$  can be determined by calculating its eigenvalues. On the other hand, for 2-D and 3-D problems,  $\mathbf{A}$  is a rectangular matrix, so that singular value decomposition will be used to determine its condition number.

### 3.3 Convergence Analysis of Adaptive linearization scheme

Here, we study the convergence behavior of our adaptive linearization scheme. From Bergamaschi and Putti [34], as  $L_i^{m+1,s}$  approaches  $\dot{\theta}(\psi_i^{m+1,s})$ , our adaptive linearization scheme essentially becomes the Newton's scheme, which exhibits quadratic convergence. In general, the convergence of our adaptive linearization scheme formulation of Equation (8) is characterized in Theorem 3.1. To show this, the idea is to leverage Definition 1.1 and find  $\psi_i^{m+1,s+1} \in H_0^1(V)$  given  $\psi_i^m, \psi_i^{m+1,s} \in H_0^1(V)$  such that:

$$\begin{aligned} \langle \theta(\psi^{m+1,s+1}) - \theta(\psi^m), v \rangle_{V_i} + \Delta t L_i^{m+1,s} (\psi_i^{m+1,s+1} - \psi_i^{m+1,s}) v + \langle S(\psi^{m+1,s+1}), v \rangle_{V_i} \\ = -\Delta t \langle K(\theta(\psi^{m+1})) \nabla(\psi^{m+1,s+1} + z), \nabla v \rangle_{V_i} \end{aligned} \quad (12)$$

holds for any  $v \in H_0^1(V)$ . We remark that, unlike previous proofs [36, 37] are based on several restrictive assumptions, some of which may not be valid in reality, our proof of convergence follows a distinct approach that is intuitive and flexible, as it does not involve any additional assumptions.

**Theorem 3.1.**  $\{\psi_i^{m+1,s}\}_s$  converges to  $\psi_i^{m+1} \in H_0^1(V)$  for  $m = 0, 1, \dots, \lceil \frac{T}{\Delta t} \rceil - 1$  and  $i = 1, \dots, N$ .

*Proof.* First, we state two general observations for water infiltration in soil:

Observation 1: There exists a scaling factor  $0 < \gamma_{s,0} < \infty$  such that  $\|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| < \gamma_{s,0} \|\psi_i^{m+1,s} - \psi_i^{m+1}\|$ . In other words,  $\|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| < +\infty$ .

Observation 2:  $\dot{\theta}(\psi) = \frac{d\theta}{d\psi}|_{\psi_i^{m+1,s}} > 0$ , which is valid in most WRC models (see Table 1). Similarly,  $\dot{S}(\psi) = \frac{dS}{d\theta} \frac{d\theta}{d\psi}|_{\psi_i^{m+1,s}} \geq 0$  in the region between the start and optimal root water extraction [47, 48].

First, we subtract Equation (3) from Equation (12) and obtain:

$$\begin{aligned} & \langle \theta(\psi^{m+1,s+1}) - \theta(\psi^{m+1}), v \rangle_{V_i} + \Delta t L_i^{m+1,s} (\psi_i^{m+1,s+1} - \psi_i^{m+1,s}) v \\ & + \langle S(\psi^{m+1,s+1}) - S(\psi^{m+1}), v \rangle_{V_i} = -\Delta t \langle K(\theta(\psi^{m+1})) \nabla(\psi^{m+1,s+1} - \psi^{m+1}), \nabla v \rangle_{V_i}. \end{aligned} \quad (13)$$

Since  $\psi_i^{m+1,s+1} - \psi_i^{m+1,s} = (\psi_i^{m+1,s+1} - \psi_i^{m+1}) - (\psi_i^{m+1,s} - \psi_i^{m+1})$ , Equation (13) can be rewritten as:

$$\begin{aligned} \Delta t L_i^{m+1,s} (\psi_i^{m+1,s} - \psi_i^{m+1}) v & = \langle \theta(\psi^{m+1,s+1}) - \theta(\psi^{m+1}), v \rangle_{V_i} + \langle S(\psi^{m+1,s+1}) - S(\psi^{m+1}), v \rangle_{V_i} \\ \Delta t L_i^{m+1,s} (\psi_i^{m+1,s+1} - \psi_i^{m+1}) v & + \Delta t \langle K(\theta(\psi^{m+1})) \nabla(\psi^{m+1,s+1} - \psi^{m+1}), \nabla v \rangle_{V_i}. \end{aligned} \quad (14)$$

Let  $v = \psi_i^{m+1,s+1} - \psi_i^{m+1}$ , then from Observation 1, the LHS of Equation (14) can be bounded by:

$$\begin{aligned} \Delta t L_i^{m+1,s} (\psi_i^{m+1,s} - \psi_i^{m+1}) v & = \Delta t L_i^{m+1,s} (\psi_i^{m+1,s} - \psi_i^{m+1}) (\psi_i^{m+1,s+1} - \psi_i^{m+1}) \\ & < \frac{\gamma_{s,0} \Delta t L_i^{m+1,s}}{\text{vol}(V_i)} (\psi_i^{m+1,s} - \psi_i^{m+1})^2 \text{vol}(V_i) \\ & = \frac{\gamma_{s,0} \Delta t L_i^{m+1,s}}{\text{vol}(V_i)} \|\psi_i^{m+1,s} - \psi_i^{m+1}\|^2. \end{aligned} \quad (15)$$

Similarly, for the third term on the RHS of Equation (14), we have:

$$\begin{aligned} \Delta t L_i^{m+1,s} (\psi_i^{m+1,s+1} - \psi_i^{m+1}) v & = \Delta t L_i^{m+1,s} (\psi_i^{m+1,s+1} - \psi_i^{m+1})^2 \\ & = \frac{\Delta t L_i^{m+1,s}}{\text{vol}(V_i)} \|\psi_i^{m+1,s+1} - \psi_i^{m+1}\|^2. \end{aligned} \quad (16)$$

By the mean value theorem, the first and second terms on the RHS of Equation (14) can be written as:

$$\begin{aligned} \langle \theta(\psi^{m+1,s+1}) - \theta(\psi^{m+1}), v \rangle_{V_i} & = \dot{\theta}(\xi_\theta) \langle \psi^{m+1,s+1} - \psi^{m+1}, v \rangle_{V_i} = \dot{\theta}(\xi_\theta) \|\psi_i^{m+1,s+1} - \psi_i^{m+1}\|^2; \\ \langle S(\psi^{m+1,s+1}) - S(\psi^{m+1}), v \rangle_{V_i} & = \dot{S}(\xi_S) \|\psi_i^{m+1,s+1} - \psi_i^{m+1}\|^2; \end{aligned} \quad (17)$$

for  $\xi_\theta, \xi_S \in (\psi_i^{m+1,s+1}, \psi_i^{m+1})$ .

Lastly, for the last term on the RHS of Equation (14), we have:

$$\Delta t \langle K(\theta(\psi^{m+1})) \nabla(\psi^{m+1,s+1} - \psi^{m+1}), \nabla v \rangle_{V_i} = \Delta t K(\theta(\psi_i^{m+1})) \|\nabla \psi_i^{m+1,s+1} - \nabla \psi_i^{m+1}\|^2. \quad (18)$$

Combining Equations (15) through (18) leads to:

$$\begin{aligned} \|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| & < \sqrt{\frac{\gamma_{s,0} L_i^{m+1,s} \Delta t}{L_i^{m+1,s} \Delta t + (\dot{\theta}(\xi_\theta) + \dot{S}(\xi_S)) \text{vol}(V_i)}} \|\psi_i^{m+1,s} - \psi_i^{m+1}\| \\ & < \sqrt{\gamma_{s,0}} \|\psi_i^{m+1,s} - \psi_i^{m+1}\|. \end{aligned} \quad (19)$$

Let  $\gamma_{s,1} := \sqrt{\gamma_{s,0}}$  and replace  $\gamma_{s,0}$  in Observation 1 by  $\gamma_{s,1}$ . One can repeat the derivations above to obtain  $\|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| < \sqrt{\gamma_{s,1}} \|\psi_i^{m+1,s} - \psi_i^{m+1}\|$ . We can further define  $\gamma_{s,2} = \sqrt{\gamma_{s,1}}$ , and so on, until we obtain a scaling factor  $\gamma_s < 1$  such that  $\|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| < \gamma_s \|\psi_i^{m+1,s} - \psi_i^{m+1}\|$ .

With this, one can show that for a given tolerance  $\epsilon > 0$ , there exists  $S \in \mathbb{N}^+$  such that:

$$\begin{aligned} \|\psi_i^{m+1,s+1} - \psi_i^{m+1}\| & < \gamma_s \|\psi_i^{m+1,s} - \psi_i^{m+1}\| < \gamma_s \gamma_{s-1} \|\psi_i^{m+1,s-1} - \psi_i^{m+1}\| < \dots \\ & < \prod_{k=1}^s \gamma_k \|\psi_i^{m+1,1} - \psi_i^{m+1}\| < \left( \max_{1 \leq k \leq s} \gamma_k \right)^s \|\psi_i^{m+1,1} - \psi_i^{m+1}\| < \epsilon \quad \forall s \geq S, \end{aligned}$$

which is consistent with the convergence criterion. This completes the proof.  $\square$



## 4 Data-driven Global Random Walk (DRW) Algorithm

Once the adaptive linearization scheme for the Richards equation is established, we integrate it with our DRW algorithm to enhance the solution accuracy and mass conservation performance. In this section, we discuss what DRW is and how it is implemented to solve the adaptive linearization scheme formulation of the Richards equation discretized by FVM. In Section 5, we illustrate the need and benefits of our DRW algorithm through case studies.

### 4.1 Formulation and Convergence of DRW Algorithm

To derive the DRW algorithm for the adaptively linearized Richards equation after it is discretized, let  $n_i^{m,s}$  be the number of water molecules in cell  $V_i$  at iteration step  $s$  and time step  $m$ , and  $\delta n_{i,j}^{m,s}$  be the number of particles moving from the cell  $V_i$  to the neighboring cell  $j \in \mathcal{N}_i$ . Clearly, we have:

$$n_i^{m+1,s} = \delta n_{i,i}^{m+1,s} + \sum_{j \in \mathcal{N}_i} \delta n_{j,i}^{m+1,s}. \quad (20)$$

For instance, for the 1-D case, we can write  $n_i^{m+1,s} = \delta n_{i,i}^{m+1,s} + \delta n_{i+1,i}^{m+1,s} + \delta n_{i-1,i}^{m+1,s}$  for every  $i = 1, \dots, N$ . Thus, as long as the relationship between pressure head  $\psi_i^{m,s}$  and the number of particles  $n_i^{m,s}$  is established, Equation (20) can be incorporated into our adaptive linearization scheme formulation of Equation (8) to solve the Richards equation iteratively. As pointed out earlier, existing global random walk (GRW) formulations (e.g., [42]) are based on the key assumption that  $\psi_i^{m,s}$  is proportional to  $n_i^{m,s}$  for any iteration  $s$  and time step  $m$ . Although this assumption is shown to be valid for diffusion equations [44], the Richards equation is a convection-diffusion equation. Thus, it remains theoretically unclear whether this assumption can be extended to the Richards equation. In fact, we will show in Section 5 that, for the Richards equation, the actual relationship between  $\psi_i^{m,s}$  and  $n_i^{m,s}$  is not only nonlinear, but may also be nonsmooth or non-explicit.

Therefore, in this work, we propose a data-driven approach to accurately approximate the nonlinear map and inverse map between  $\psi_i^{m,s}$  and  $n_i^{m,s}$  and integrate this approach with the GRW framework. The resulting data-driven global random walk (DRW) algorithm uses two neural networks to learn the two relationships. A neural network is capable of approximating any function provided that it contains enough neurons [49, 50]. Depending on the problem settings, the desired choices of optimal optimizer, number of hidden layers, and activation functions can vary. Based on our extensive research, we find that a simple three-layer neural network with 256 neurons in each layer achieves the best performance for most 1-D through 3-D problems compared to other more complex neural network architectures (e.g., LSTM). Also, we find that stochastic gradient decent (SGD) optimizer often outperforms others (e.g., Adam or RMSProp). Overall, one neural network, denoted as  $\hat{f}_{\text{NN}}$ , learns the map  $f : n_i^{m,s} \rightarrow \psi_i^{m,s}$ , whereas the other neural network, denoted as  $\hat{f}_{\text{NN}}^{-1}$  learns the inverse map  $f^{-1} : \psi_i^{m,s} \rightarrow n_i^{m,s}$ .

When neural network training is complete, the following DRW formulation of the adaptive linearization scheme of the Richards equation incorporates  $\hat{f}_{\text{NN}}^{-1}$  in Equation (8):

$$n_i^{m+1,s+1} = n_i^{m+1,s} + \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \mathbf{e} \cdot \mathbf{n}_{\omega_{i,j}} \frac{n_j^{m+1,s} - n_i^{m+1,s}}{\text{dist}(V_j, V_i)} A_{\omega_{i,j}} + \hat{f}_{\text{NN}}^{-1}(J), \quad (21)$$

where  $J = \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \mathbf{e} \cdot \mathbf{n}_{\omega_{i,j}} \frac{z_j^{m+1,s} - z_i^{m+1,s}}{\text{dist}(V_j, V_i)} A_{\omega_{i,j}} - \frac{1}{L_i^{m+1,s}} \left( \frac{\theta_i^{m+1,s} - \theta_i^m}{\Delta t} + S(\psi_i^{m+1,s}) \right) \text{vol}(V_i)$ .

To solve Equation (21), we will adopt a similar strategy as in Equation (10) to automatically select

the appropriate linearization parameter  $L_i^{m+1,s}$ . Note that in actual implementation, we typically choose  $L_i^{m+1,s}$  at its lower bound.

Similarly, one can extend Theorem 3.1 and derive a similar convergence property for the sequence  $\{n_i^{m+1,s}\}_s$ :

**Theorem 4.1.**  $\{n_i^{m+1,s}\}_s$  converges to  $n_i^{m+1} \in H_0^1(V)$  for  $m = 0, 1, \dots, \lceil \frac{T}{\Delta t} \rceil - 1$  and  $i = 1, \dots, N$ .

Once the converged solutions  $n_i^{m+1}$  are obtained, they will be converted to  $\psi_i^{m+1}$  using  $\hat{f}_{\text{NN}}$ .

## 4.2 Implementation of DRW Algorithm

Two steps, namely neural network training and solution process, are involved in the DRW algorithm to solve the discretized Richards equation. As discussed above, in neural network training, two neural networks, namely  $\hat{f}_{\text{NN}}$  and  $\hat{f}_{\text{NN}}^{-1}$ , are trained to approximate  $f$  and  $f^{-1}$ , respectively. First, for each cell  $V_i$  and time step  $m$ , we obtain  $n_i^{m,S}$  from off-the-shelf or in-house developed GRW solvers (e.g., [42]), which adopt standard linearization scheme using a static linearization parameter. Here,  $S$  is the user-specified total iteration number. The corresponding  $\psi_i^{m,S}$  solutions are obtained from the adaptive linearization scheme of Equation (8). The resulting set of solution pairs,  $\mathcal{S} = \{(\psi_i^{m,S}, n_i^{m,S})\}_{i,m}$ , form the original set of ‘‘reference solutions’’.

Next, we apply data augmentation to  $\mathcal{S}$  to significantly increase the size of the training set. This is achieved by introducing Gaussian noise  $Z_q \sim \mathcal{N}(0, \sigma_q^2)$  with different variances  $\sigma_1^2, \dots, \sigma_Q^2$  to each and every reference solution in  $\mathcal{S}$ . In other words,  $\psi_{i,q}^{m,S} := \psi_i^{m,S} + Z_q$  and  $n_{i,q}^{m,S} := n_i^{m,S} + Z_q$ . After data augmentation, the resulting expanded set of reference solutions  $\{(\psi_{i,q}^{m,S}, n_{i,q}^{m,S})\}_{i,m,q}$ , denoted as  $\mathcal{S}'$ , will be used for neural network training. In Section 5.1, we will show that introducing Gaussian noise can greatly reduce the biases of reference solutions and enhance generalization performance [51], thereby significantly improving the accuracy of numerical solutions.

Once offline training (only one time) is complete, the initial and boundary conditions of the Richards equation will be mapped to the number of particles using  $\hat{f}_{\text{NN}}^{-1}$ . Then, starting from  $m = 0$  (initial condition), we can leverage  $\hat{f}_{\text{NN}}^{-1}$  and solve Equation (21) is solved iteratively to obtain a sequence  $\{n_i^{m+1,s}\}_s$  for each new time step  $m + 1$  and cell  $V_i$ . To monitor the convergence of  $\{n_i^{m+1,s}\}_s$ , we define the relative error  $\text{RE}_s$  as:

$$\text{RE}_s := \left| \frac{n_i^{m+1,s+1} - n_i^{m+1,s}}{n_i^{m+1,s+1}} \right|. \quad (22)$$

Once  $\text{RE}_s$  is below a user-specified tolerance  $\text{tol}$  (typically in the order of  $10^{-6}$ ), we declare convergence of  $\{n_i^{m+1,s}\}_s$  to  $n_i^{m+1}$ . With this, we can determine the converged  $\psi_i^{m+1}$  using  $\hat{f}_{\text{NN}}$ , followed by obtaining other physical quantities such as soil moisture content  $\theta_i^{m+1}$  and  $\mathbf{q}_i^{m+1}$  from the WRC and HCF models (Table 1) and Equation (1).

Furthermore, it is worth mentioning that, when neural network training for a specific problem setting (e.g., boundary condition and initial condition) is complete, the trained neural networks can be saved as a pretrained model. As we encounter a new problem setting, the pretrained model can be quickly trained with a small number of epochs (typically no more than 100) before it can be deployed to solve the new problem.

## 5 Case Studies

We have now completed the formulation of our DRW-based solution algorithm for the Richards equation. In this section, we systematically validate our DRW numerical framework on selected

1-D through 3-D benchmark problems modified from the literature [19, 52, 53, 54, 55]. Specifically, we extensively study the 1-D benchmark problem of Celia et al. [19] to demonstrate the need and benefits of adaptive linearization scheme, data-driven global random walk, and data augmentation. Also, using this problem as a benchmark, we demonstrate the accuracy of our solution algorithm with respect to state-of-the-art solvers. In the 1-D layered soil case study proposed by Berardi et al. [54], we show that our solution algorithm is capable of handling discontinuities in soil properties and modeling the infiltration process through the interface of two different soils. In the 2-D case study adopted from Gąsiorowski and Kolerski [52], we show that our DRW algorithm can better satisfy mass balance embedded in the Richards equation. In the 3-D case study adopted from Tracy [53] in which an analytical solution to the Richards equation exists, we show that our DRW algorithm produces much more accurate solutions compared to GRW solvers. Finally, we study a 3-D problem adopted from Orouskhani et al. [55] featuring an actual center-pivot system and validate the accuracy and robustness of our DRW algorithm in modeling real-world precipitation and irrigation scenarios for a long period of time.

## 5.1 A 1-D Benchmark Problem

Here, we study the 1-D benchmark problem over a 40 cm deep soil presented by Celia et al. [19]. The HCF and WRC adopt the model of Haverkamp et al. [8] (see Table 1). And the parameters are listed in Table 5.1. The initial condition is given by  $\psi(z, 0) = -61.5$  cm, whereas the two boundary conditions are  $\psi(40 \text{ cm}, t) = -20.7$  cm,  $\psi(0, t) = -61.5$  cm, respectively [8]. This benchmark problem ignores the sink term.

Soil-specific Parameters	Values	Units
Saturated hydraulic conductivity, $K_s$	0.00944	cm/s
Saturated soil moisture content, $\theta_s$	0.287	–
Residual soil moisture content, $\theta_r$	0.075	–
$\alpha$ in Haverkamp’s model	$1.611 \times 10^6$	cm
$A$ in Haverkamp’s model	$1.175 \times 10^6$	cm
$\beta$ in Haverkamp’s model	3.96	–
$\gamma$ in Haverkamp’s model	4.74	–
Total time, $T$	360	s

Table 2: soil-specific parameters and their values used in the 1-D case study of Celia et al. [19] based on the empirical model developed by Haverkamp et al. [8].

Through this 1-D illustrative example, we will highlight the benefits of (a) adopting an adaptive linearization scheme as opposed to standard linearization scheme, (b) implementing the DRW algorithm as opposed to GRW algorithms, and (c) integrating the adaptive linearization scheme with DRW in a holistic numerical framework.

### 5.1.1 The Need for Adaptive Linearization Scheme

To illustrate how adaptive linearization scheme improves convergence and accuracy of conventional linearization schemes, we compare the pressure head solution profiles at  $t = T = 360$  seconds obtained by different linearization parameters after (a)  $S = 500$  iterations and (b)  $S = 10,000$  iterations. We adopt a spatial grid containing 101 mesh points ( $\Delta z = 0.4$  cm) and a temporal grid of  $\Delta t = 1$  second. As shown in Figure 1, when using standard linearization scheme, the choice of

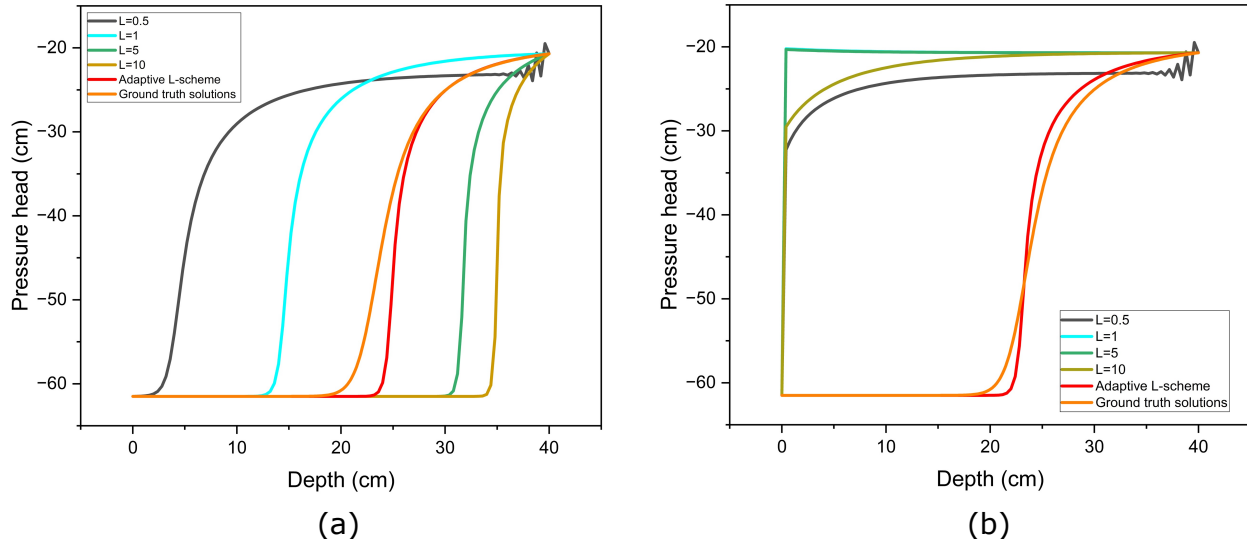


Figure 1: Comparison of pressure head solution profiles after (a)  $S = 500$  and (b)  $S = 10000$  iterations for the 1-D benchmark problem [19] using standard and our adaptive linearization schemes (Equation (8)). The solutions obtained from Celia et al. [19] based on very fine space and time steps are marked as the “ground truth solutions”.

linearization parameter and the total number of iterations can impact the solution accuracy and algorithm stability significantly. For example, when the linearization parameter is too small (e.g.,  $L = 0.5$  for this problem), the stability of the linearization scheme can be severely affected (as illustrated by the zigzag pressure head profile towards  $z = 40$  cm). Another key observation is that, increasing the total number of iterations sometimes deteriorates solution accuracy of standard linearization schemes. These observations pose a practical challenge to use standard linearization scheme for solving the Richards equation, especially when the ground truth solutions are absent, as identifying the optimal linearization parameter and total number of iterations that would yield accurate solutions will not be possible without referring to ground truth solutions.

Unlike standard linearization schemes, by implementing the adaptive linearization scheme of Equation (8) (note that DRW algorithm is not yet introduced), we observe that the pressure head solutions are close to the ground truth solutions even when only a limited number of iterations ( $S = 500$ ) is used. Furthermore, as the number of iterations increases, the solution accuracy actually improves. This makes adaptive linearization scheme a robust and reliable numerical scheme that produce solutions that are close to true solutions in one shot. Also, it is worth noting that our adaptive linearization scheme successfully bypasses the singularity issue as  $L_i^{m+1,s}$  approaches to 0 and correctly calculates the pressure head solutions for  $z \in [0, 20$  cm] where  $\theta(\psi)$  becomes small.

### 5.1.2 The Need for Data-driven Global Random Walk

To generate the reference solutions, we consider a coarse spatial discretization containing 40 cells (i.e., grid size  $\Delta z = 1$  cm) and solve for  $T = 40$  seconds. Therefore, we obtain 1,640 pressure head solutions (41 mesh points  $\times$  40 time steps) using the finite difference method developed by Celia et al. [19]. On the other hand, the number of particles solutions are obtained from the GRW solver developed by Suciú et al. [42] for different static linearization parameter values ranging from 0.5 to 15. To obtain the initial and boundary conditions in terms of the number of particles, we multiply these conditions, originally represented in terms of pressure head, by a factor of  $10^{10}$  particles per cm of pressure head.

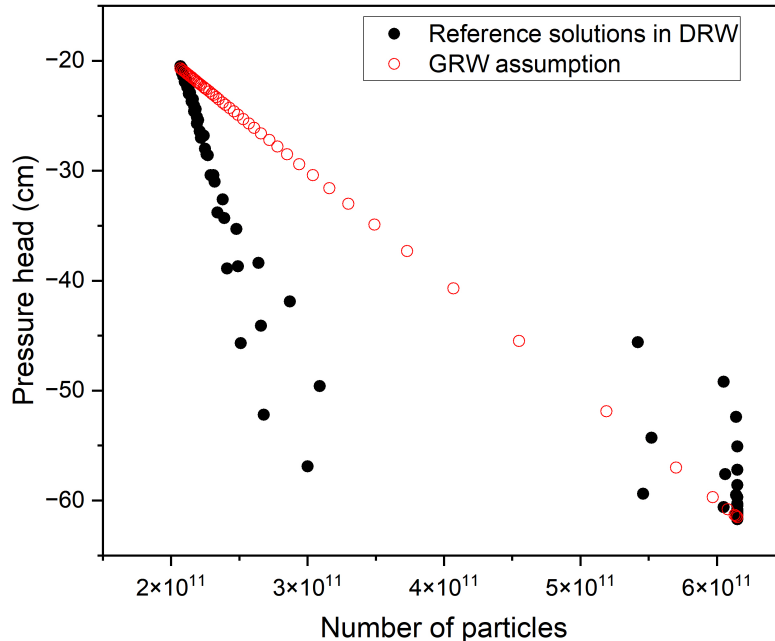


Figure 2: The relationships between  $\psi_i^{m,S}$  and the number of particles  $n_i^{m,S}$  at  $S = 20,000$  iterations to be used by our DRW algorithm (black filled circles) and the GRW solver (red empty circles), respectively.

As discussed earlier, a key underlying assumption underneath existing GRW solvers is that  $\psi_i^{m,S}$  is proportional to  $n_i^{m,S}$ . To validate this assumption, in Figure 2, we plot 1,640 original reference solutions prior to performing data augmentation. It is clear from Figure 2 that, while the magnitude of pressure head shows roughly an increasing trend with respect to the number of particles present in a cell, the exact relationship between the number of particles and pressure head is neither smooth nor explicit. Interestingly, we do observe close-to-linear trends in specific regions of pressure head values (i.e., between  $-20$  and  $-30$  cm, and between  $-50$  and  $-60$  cm). By examining Figure 1, one can see that these two regions correspond to the two ends of the spatial domain. Indeed, towards both ends,  $\frac{\partial \psi}{\partial z}$  is close to 0, indicating that the advection term in Equation (1),  $\nabla \cdot (K \nabla z) = \frac{\partial K}{\partial z} = \frac{\partial K}{\partial \psi} \frac{\partial \psi}{\partial z}$ , vanishes. Thus, the Richards equation essentially becomes the diffusion equation, in which the proportionality assumption between pressure head and the number of particles is shown to be valid [44]. However, for the rest of the region where pressure head changes rapidly with respect to depth, this assumption is no longer valid, calling for the use of data-driven methods, such as neural network approximation used in our DRW algorithm, to effectively learn the mapping and inverse maps between pressure head and the number of particles.

### 5.1.3 Improving DRW Algorithm Performance via Data Augmentation

To implement our DRW algorithm, we use two fully-connected neural networks, each containing 3 hidden layers and 256 neurons in each layer, to learn  $f$  and  $f^{-1}$ . We use the leaky ReLU activation function [56] and stochastic gradient descent (SGD) optimizer with a learning rate of 0.001. While the original dataset used for neural network training contains 1,640 reference solutions, we also perform data augmentation to increase dataset size. To do this, we use the GRW solver [42]

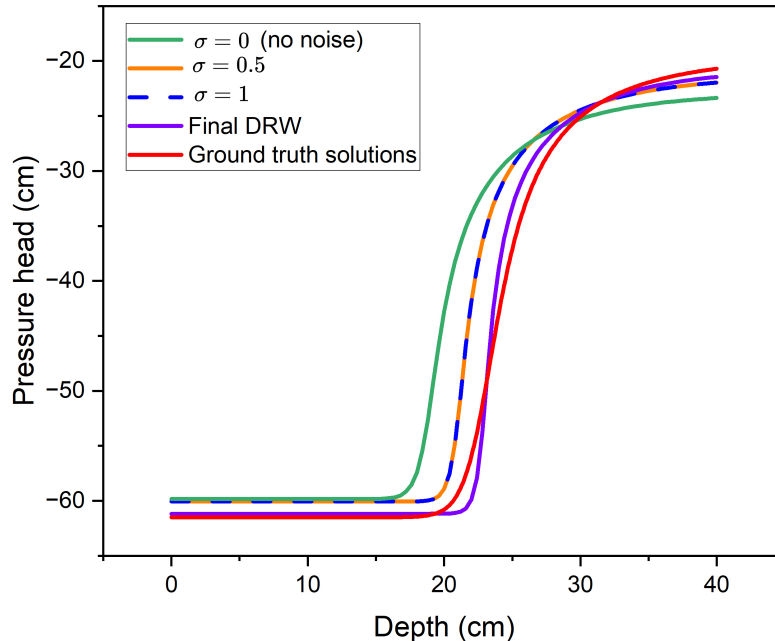


Figure 3: Pressure head profiles synthesized using different numerical methods and different training data. Specifically, curves named  $\sigma = 0, 0.5, 1$  are calculated from our DRW algorithm, in which only 1,640 reference solutions (with Gaussian noise added for  $\sigma = 0.1, 0.5, 1$ ) are used for neural network training. The purple curve is the final DRW pressure head profile obtained by considering the full 17,104 data points for neural network training through data augmentation.

to obtain the number of particles solutions. We then make multiple copies of the pressure head solutions and append each copy to the number of particles solutions obtained from the GRW solver [42]. Finally, we add zero-mean Gaussian noises with standard deviation varying from 0.1 to 0.5 to these augmented reference solutions. Overall, this leads to a total of 17,104 sets of reference solutions for neural network training and validation. Note that, as previously discussed, the original and augmented reference solutions are generated using a coarse grid ( $\Delta z = 1$  cm). Thus, they can be obtained relatively efficiently. On the other hand, in the solution step, we will use a more refined grid containing 101 mesh points ( $\Delta z = 0.4$  cm).

To understand the impact of data augmentation to the solution quality of our DRW algorithm, we examine several cases as illustrated in Figure 3. First, we observe that, compared to directly using the original reference solutions for neural network training, simply introducing Gaussian noise to the reference solutions can significantly improve the solution accuracy of our DRW algorithm. Note that these augmented reference solutions come from a small set of original reference solutions generated from a coarse grid. This “coarse-to-fine” approach can therefore enhance the solution accuracy of our DRW algorithm without requiring a large amount of high-accuracy, fine-mesh training data. Furthermore, when augmented reference solutions are used for training, only 100 additional epochs are needed to retrain the neural networks that are already trained using the original reference solutions. Second, we notice that there is almost no difference in final pressure head solution profile when Gaussian noises of different magnitudes are directly added to the original reference solutions without augmenting them together. Third, increasing the size of training data (from 1,640 to 17,104) via data augmentation of original reference solutions is an

effective way to improve the solution accuracy of our DRW algorithm, as the pressure head profile matches very well with the ground truth solution.

#### 5.1.4 Incorporating Adaptive Linearization Scheme in GRW Framework

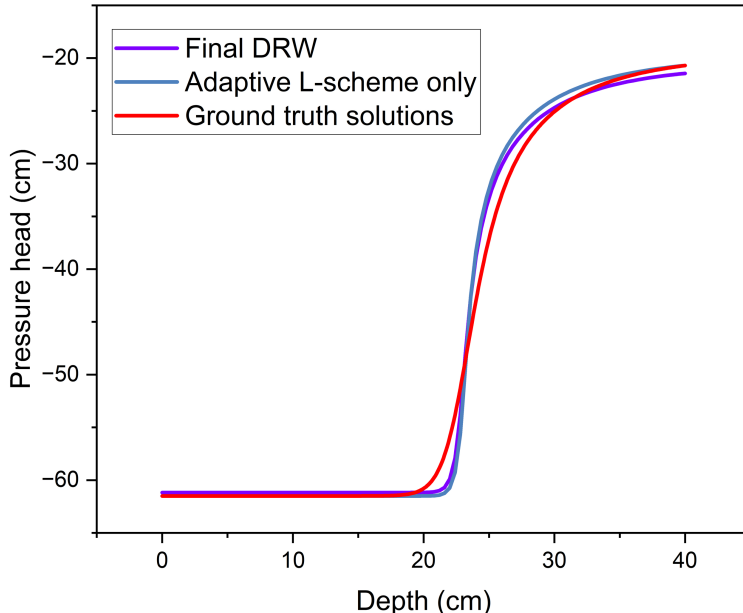


Figure 4: Comparison of pressure head solution profiles produced from adaptive linearization scheme itself (Equation (8)) and from the integrated DRW framework coupled with adaptive linearization scheme (Equation (21)).

From Figure 4, it is clear that incorporating adaptive linearization scheme in the DRW framework synergistically improves the overall solution accuracy of the Richards equation, especially in the region where pressure head changes rapidly with respect to depth (i.e., between  $z = 20$  to  $30$  cm). On the other hand, we observe slight discrepancy in pressure head solution close to  $z = 40$  cm when comparing our DRW algorithm with ground-truth solutions, whereas the solution produced by adaptive linearization scheme alone matches perfectly with ground-truth solution at  $z = 40$  cm, which corresponds to one of the boundary conditions. Similar discrepancies have been observed in Figure 3 as well. We believe that this is due to the fact that  $\hat{f}_{\text{NN}}$  and  $\hat{f}_{\text{NN}}^{-1}$  can only approximate the nonlinear maps of  $f$  and  $f^{-1}$ , respectively, and the resulting induced error causes discrepancies in pressure head solutions even at the boundaries. To overcome this limitation, one way is to increase the size of the augmented reference solutions for neural network training. This is supported by the observations in Figure 3, in which the pressure head solutions get closer to the ground truth solutions at  $z = 40$  cm as data augmentation takes place. Another approach is to switch from DRW (Equation (21)) to adaptive linearization scheme only (Equation (8)) when solving for the boundary conditions. We leave this research for future work.

#### 5.1.5 Comparison with Benchmark Solvers

We compare our DRW solver with benchmark algorithms based on computational performance and solution accuracy under two scenarios. In Scenario 1, we set the error tolerance  $\text{tol}$  to be  $3.2 \times 10^{-5}$ , whereas in Scenario 2, we set the total number of iterations  $S = 500$ . In terms of computational

efficiency, we use two metrics. The first metric is the relative error RE defined in Equation (22), which measures speed of convergence. The second metric is the condition number of matrix  $\mathbf{A}$  defined in Equation (11), which measures the sensitivity of linearization scheme subject to small perturbations. These results are summarized in Tables 3 through 6.

Algorithm	Average number of iterations needed (Average final RE reached)	
	Standard linearization scheme	Adaptive linearization scheme
GRW [42]	736 ( $3.1999 \times 10^{-5}$ )	1 ( $9.2873 \times 10^{-6}$ )
DRW	2 ( $3.1975 \times 10^{-5}$ )	1 ( $8.5403 \times 10^{-6}$ )

Table 3: Comparison of average number of iterations across all discretized cells and time steps needed to reach the specified tol (Scenario 1) for GRW and DRW algorithms that implement standard or adaptive linearization scheme. In standard linearization scheme, we use the optimal static linearization parameter of 3.5 identified by trial-and-error process.

Algorithm	Average final RE reached	
	Standard linearization scheme	Adaptive linearization scheme
GRW [42]	$4.1130 \times 10^{-5}$	$4.8824 \times 10^{-7}$
DRW	$3.9287 \times 10^{-5}$	$5.2006 \times 10^{-7}$

Table 4: Comparison of average RE after 500 iterations (Scenario 2) across all discretized cells and time steps for GRW and DRW algorithms that implement standard or adaptive linearization scheme. We also use the optimal static linearization parameter of 3.5 when implementing the standard linearization scheme.

Algorithm	Average condition number of $\mathbf{A}$ obtained from [45] (Scenario 1)	
	Standard linearization scheme	Adaptive linearization scheme
GRW [42]	1.7666	1.0064
DRW	1.7472	1.0074

Table 5: Comparison of average condition number under Scenario 1 across all time steps (as Equation (11) already considers all discretized cells) for GRW and DRW algorithms that implement standard or adaptive linearization scheme.

Algorithm	Average condition number of $\mathbf{A}$ obtained from [45] (Scenario 2)	
	Standard linearization scheme	Adaptive linearization scheme
GRW [42]	1.7206	1.0064
DRW	1.7137	1.0074

Table 6: Comparison of average condition number under Scenario 2 across all time steps for GRW and DRW algorithms that implement standard or adaptive linearization scheme.

From Tables 3 and 4, it is clear that, compared to using standard linearization scheme, implementing adaptive linearization scheme can greatly accelerate convergence for both in GRW and DRW



solvers. From Tables 5 and Table 6, we see that implementing adaptive linearization scheme also significantly improves the stability of GRW and DRW based solvers, as matrix  $\mathbf{A}$  is well-conditioned. Both observations suggest that adaptive linearization scheme outperforms standard linearization in enhancing convergence behavior of both GRW and DRW algorithms and is a powerful iterative procedure to solve the Richards equation.

In terms of accuracy, we also consider two metrics. The first metric is the discrepancy against ground truth solutions of Celia et al. [19]. The comparison results are illustrated in Figure 5. And the second metric is the solver’s performance in preserving the mass (moisture) balance, which is quantified by the mass balance measure MB defined in [19]:

$$\text{MB} = \frac{\text{total additional mass in the domain}}{\text{total water flux into the domain}}. \quad (23)$$

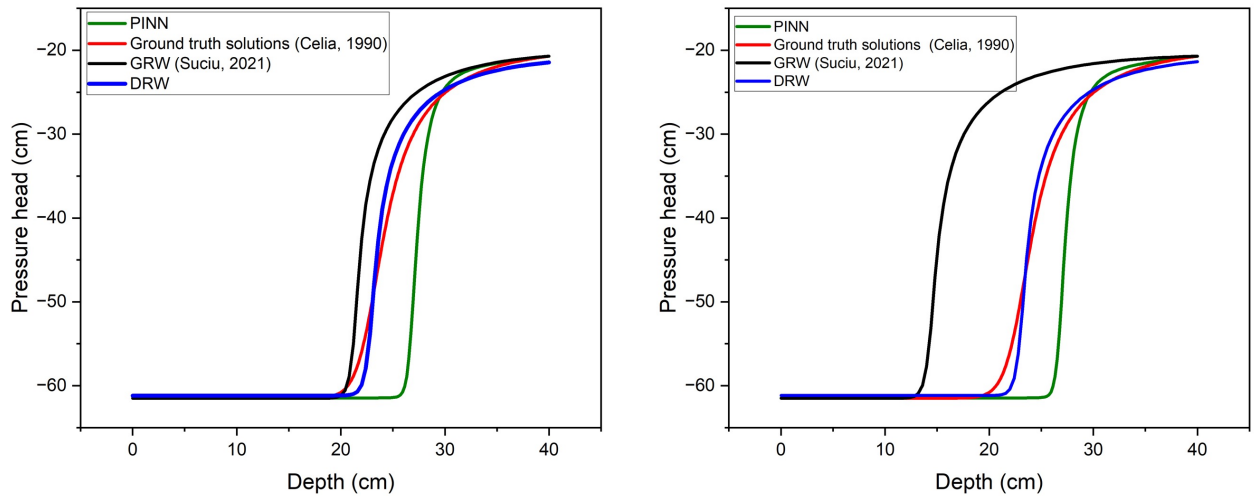


Figure 5: Pressure head profiles at  $t = T = 360$  sec obtained by different algorithms under (left) Scenario 1, and (right) Scenario 2. Our DRW algorithm incorporates adaptive linearization scheme, while the GRW algorithm adopts the standard linearization scheme (as implemented in Suciu et al. [42]). Note that physics-informed neural network (PINN) method is not an iterative method, thus the solution profile is the same under both scenarios.

In Figure 5, we compare the pressure head profiles obtained from our DRW algorithm (which implements adaptive linearization scheme), the GRW algorithm, and a state-of-the-art physics-informed neural network (PINN) method based on the work of Bandai and Ghezzehei [57], against the ground truth solution [19]. Clearly, as more iterations are used, both GRW and DRW solutions move closer to ground truth solutions. However, in both scenarios, compared with the DRW solutions, PINN and GRW solutions are further apart from ground truth solutions.

In our earlier work [28], we illustrated the DRW algorithm’s ability to accurately capture underlying physics of water flow dynamics in soil, including the relationships among pressure head, soil moisture, and water flux. Extending these observations, we compare the mass conservation performance among various algorithms.

As summarized in Table 7, among all numerical methods studied, in both Scenarios 1 and 2, our DRW algorithm achieves the highest MB values while using the coarsest time step, which is more computationally efficient than using finer time steps. Similarly, Table 8 shows that, compared to other numerical methods, our DRW algorithm achieves the highest MB values when fixing  $\Delta t$  to be the same. In addition, given more iterations, the MB value in all iterative methods will improve.

Method used	Scenario	$\Delta t$ (sec)	MB
GRW algorithm [42]	1	16.79	96.46%
DRW algorithm	1	18.68	<b>98.92%</b>
GRW algorithm	2	16.87	62.12%
DRW algorithm	2	18.20	<b>99.28%</b>
Celia et al. [19]	N/A	10	95.00%

Table 7: MB results of different numerical methods. Note that here,  $\Delta t$  is determined for each method by the heuristic formula of Equation 5 in [42].

Method used	Scenario	MB ( $\Delta t = 10$ sec)
GRW algorithm [42]	1	98.00%
DRW algorithm	1	<b>99.47%</b>
GRW algorithm	2	80.37%
DRW algorithm	2	<b>99.85%</b>
Celia et al. [19]	N/A	95.00%

Table 8: MB results of different numerical methods, in which a common  $\Delta t = 10$  sec is used for all numerical methods.

This is also consistent with how the accuracy of the pressure head profiles improves with increasing iterations when comparing Figure 5a with 5b.

## 5.2 A 1-D Layered Soil Benchmark Problem

To investigate how robust our DRW algorithm is in handling realistic problems, we study the classic Hills’ problem [58] that involves the 1-D water infiltration into two layers of very dry soil, each having a depth of 30 cm. The top layer (layer 1) corresponds to Berino loamy fine sand and the bottom layer (layer 2) corresponds to Gledale clay loam. The WRC and HCF follow the Mualem-van Genuchten model. The soil-specific parameters are extracted from [58] and are listed in Table 9. This benchmark problem also ignores the sink term.

As pointed out by Berardi et al. [54], the dry condition is the most challenging physical case to model from a numerical point of view. The presence of discontinuous interface across the two soil layers presents yet another complication to this problem. And we simulate the problem up to a total time of  $T = 7.5$  minutes. For neural network training, we generate a total of 30,500 reference solutions by the GRW solver [42] using the optimal static linearization parameter of 5.

Soil	$\theta_r$	$\theta_s$	$\alpha$	$n$	$K_s$
Berino loamy fine sand	0.029	0.366	0.028	2.239	541.0
Gledale clay loam	0.106	0.469	0.010	1.395	13.10

Table 9: Soil-specific parameters and constants used in the layered soil problem of Hills et al. [58].

Figure 6 illustrates the soil moisture profile at three different times obtained using our DRW algorithm, the GRW solver, and the Transversal Method of Lines (TMOL) [54] which is considered the current state-of-the-art algorithm for this problem. All three approaches adopt the same discretized temporal ( $\Delta t = 1$  second) and spatial steps ( $\Delta z = 1$  cm). We set the  $RE_s = 10^{-5}$  as the

common stopping criteria for every cell and time step. From Figure 6, we observe that our DRW algorithm is capable of successfully simulating this challenging problem with discontinuities in soil properties at the interface. The soil moisture solutions obtained by our DRW algorithm are also consistent with existing solvers. In fact, compared to the GRW solver, the solutions produced by our DRW algorithm are closer to the state-of-the-art TMOL solutions.

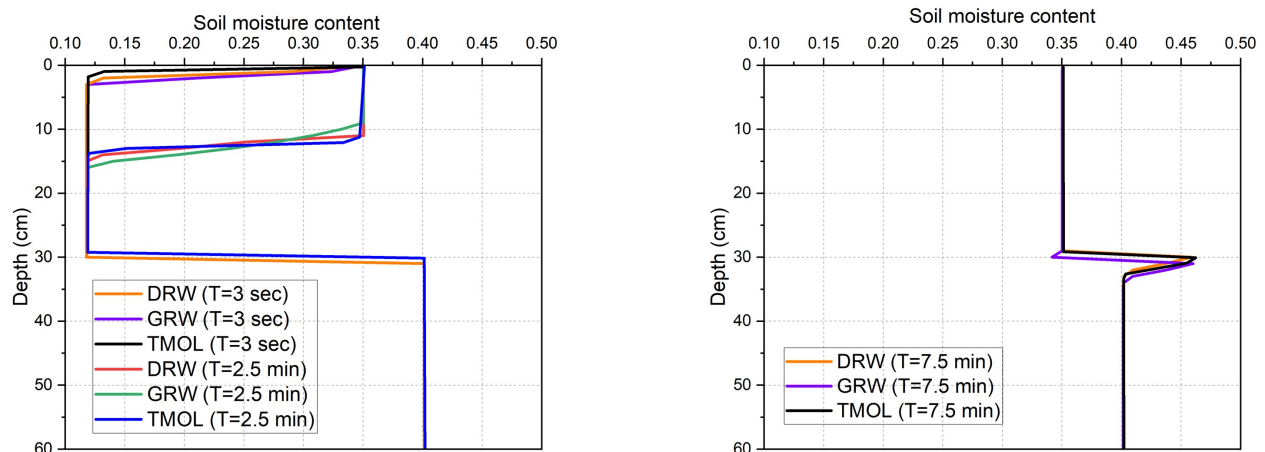


Figure 6: Comparison of soil moisture content profile obtained different methods with  $\Delta z = 1$  cm under (left) DRW, GRW and TMOL at  $t = T = 3$  sec and  $t = T = 2.5$  min and (right) DRW, GRW and TMOL at  $t = T = 7.5$  min. Note that TMOL by Berardi et al. [54] is not an iterative method. GRW and DRW are implemented for 500 iterations at every time step.

### 5.3 A 2-D Benchmark Problem

In the second example, we study the 2-D Richards equation for an infiltration process in a  $1\text{m} \times 1\text{m}$  loam soil field [52]. The spatial steps in both horizontal ( $\Delta x$ ) and vertical ( $\Delta z$ ) directions are set to be 0.02 m. And the time step used for this comparison study is  $\Delta t = 10$  seconds. The Mualem-van Genuchten model (see Table 1) was used in this case study. And the soil-specific parameters, given by Carsel and Parrish [59], are listed in Table 10. This problem also ignores the sink term.

Property	Symbol	Value	Units
Saturated hydraulic conductivity	$K_s$	$2.89 \times 10^{-6}$	m/s
Saturated water content	$\theta_s$	0.43	–
Residual water content	$\theta_r$	0.078	–
van Genuchten Constant	$\alpha$	3.6	$\text{m}^{-1}$
van Genuchten Constant	$n$	1.56	–
Total time	$T$	$1.26 \times 10^4$	s

Table 10: Soil-specific parameters and constants used in 2-D case study.

The initial and boundary conditions of this case study are given by:

$$\text{Initial condition: } \psi(x, z, t = 0 \text{ s}) = \begin{cases} 0 \text{ m,} & x \in [0.46, 0.54] \text{ m, } z = 0 \text{ m,} \\ -10 \text{ m,} & \text{otherwise.} \end{cases}$$

Boundary condition:  $\psi(x \in [0.46, 0.54] \text{ m, } z = 0, t) = 0 \text{ m}$ , no slip conditions for other boundaries.

Note that the initial and boundary conditions are symmetric along  $x = 0.5 \text{ m}$ . We first obtain 1,734 original reference solutions for neural network training from the 2-D GRW solver [42], which implements the standard linearization scheme (optimal linearization parameter is 0.5) and a spatial step of 0.05 m. Then, we apply data augmentation by adding Gaussian noises with  $\sigma^2$  values ranging from 0.1 to 0.5 to generate a total of 26,010 reference solutions (which also contain the original reference solutions). These reference solutions are used to train the two neural networks for our DRW algorithm. Each neural network contains 2 hidden layers and 25 neurons in each layer. ReLU activation function is adopted in each layer, and each neural network is trained by Levenberg-Marquardt optimization for 1,000 epochs. We set the total iteration number to be  $S = 500$ , at which the relative error calculated using Equation (22) for our DRW algorithm and the GRW solver are given by  $3.675 \times 10^{-6}$  and  $1.094 \times 10^{-5}$ , respectively. This indicates that our DRW algorithm achieves faster convergence per iteration than the GRW solvers.

Meanwhile, we also simulate this 2-D problem using HYDRUS 2D software (version 5.0) [60] and compare the pressure head results at  $t = T = 1.26 \times 10^4 \text{ sec}$  with our DRW algorithm and the GRW solver. From Figure 7, we can draw two observations. First, the pressure head solution profiles for both GRW and DRW algorithms appear to be symmetric along  $x = 0.5 \text{ m}$ , whereas HYDRUS 2D shows a clear asymmetric profile. As pointed out earlier, since the initial and boundary conditions are symmetric along  $x = 0.5 \text{ m}$ , symmetry in the pressure head solutions is expected. This suggests that both GRW and DRW based solvers can capture some degree of underlying physics of the original problem. Second, despite the asymmetric behavior in pressure head profile, the size of isolines for the HYDRUS 2D simulation result is more similar to our DRW solution than to the GRW solver solution. This observation is also consistent with the information presented in Figure 9a. In fact, both observations can be carried over to the soil moisture profile as well, as shown in Figures 8 and 9b.

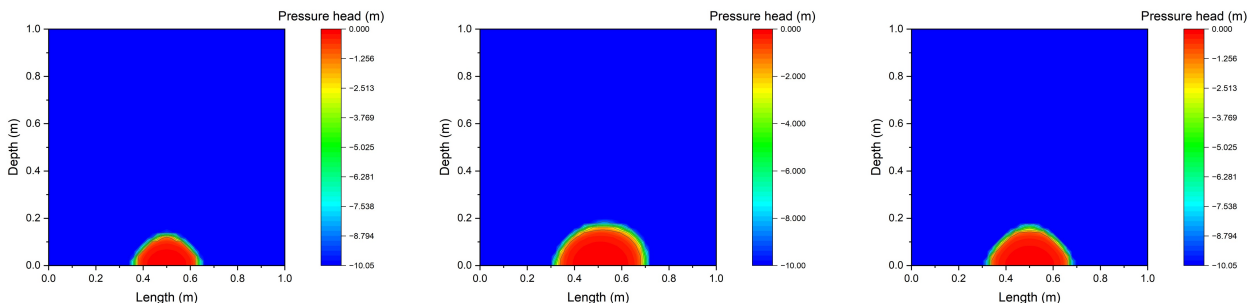


Figure 7: Pressure head solution profile obtained from three numerical methods: (left) GRW solver (linearization parameter = 0.5); (middle) HYDRUS 2D software; (right) our DRW algorithm.

On the other hand, when comparing the water flux results, we see from Figure 10 that the GRW solution no longer preserves the symmetry of water flux profile along the horizontal direction. In other words, among the three numerical methods considered in this case study, our DRW algorithm achieves the best performance in terms of preserving the symmetry implied by the problem. This result is also consistent with the mass conservation calculations using Equation (23), as our DRW

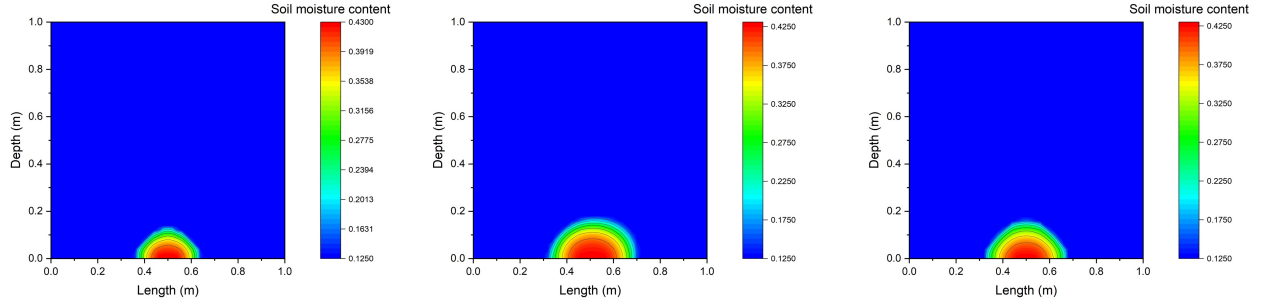


Figure 8: Soil moisture solution profile obtained from three numerical methods: (left) GRW solver (linearization parameter = 0.5); (middle) HYDRUS 2D software; (right) our DRW algorithm.

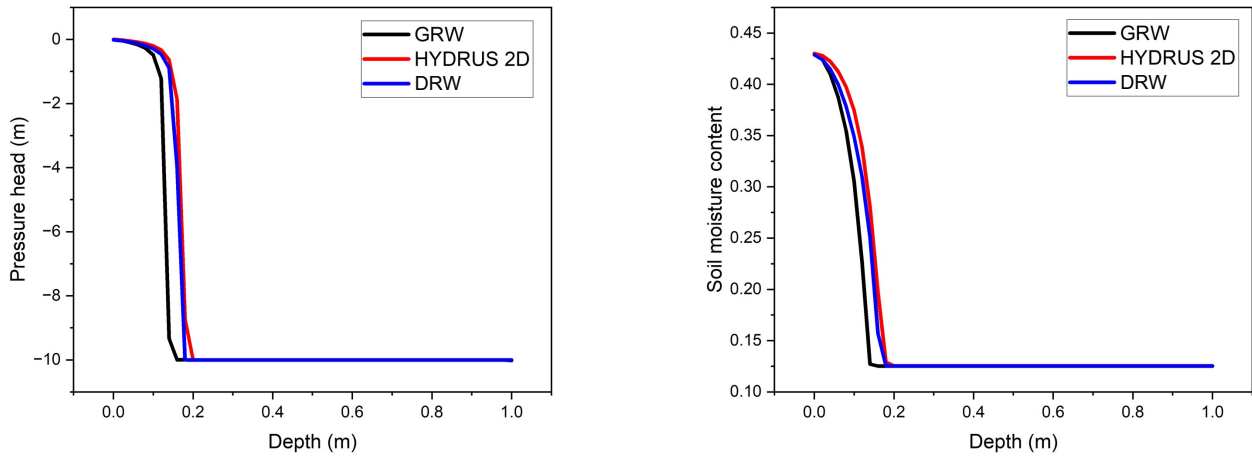


Figure 9: Cross-sectional view ( $x = 0.5\text{m}$ ) of: (left) the pressure head profile; (right) soil moisture profile.

algorithm achieves significantly higher MB value compared to other benchmark solver (see Table 11).

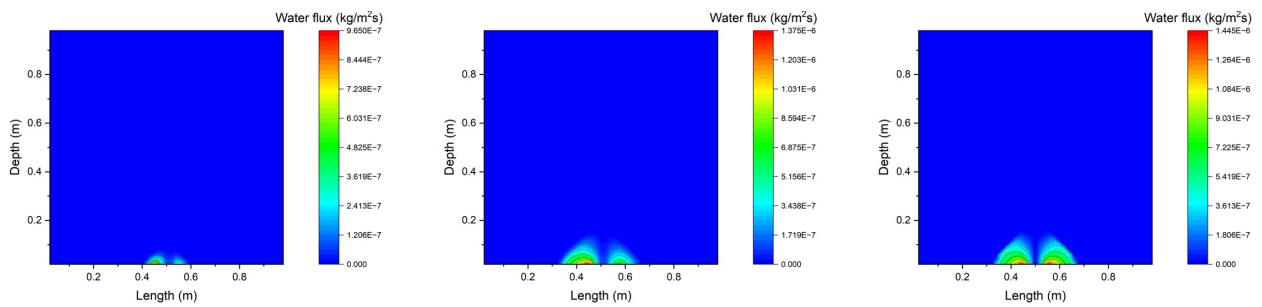


Figure 10: Magnitude of water flux along the horizontal ( $x$ -axis) direction for three numerical solvers: (left) GRW solver (linearization parameter = 0.5); (middle) HYDRUS 2D software; (right) our DRW algorithm. Note that, along the horizontal direction, the water flux is negative in  $[0, 0.5]$  m and positive in  $(0.5, 1]$  m.

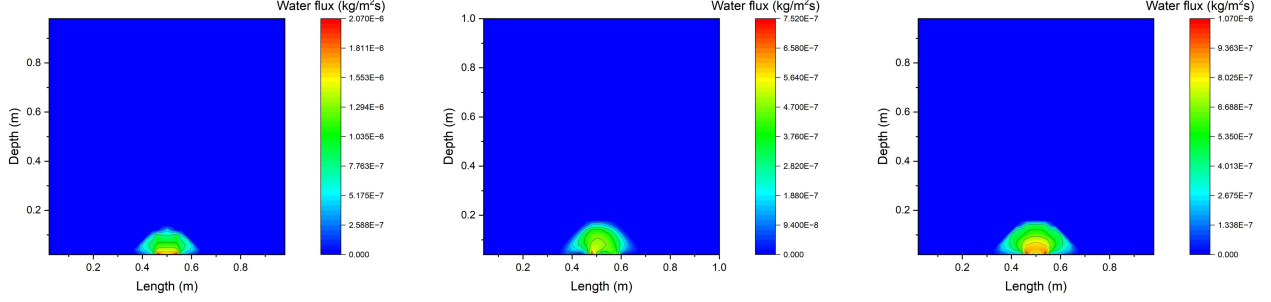


Figure 11: Magnitude of water flux along the vertical ( $z$ -axis) direction for three numerical solvers: (left) GRW solver (linearization parameter = 0.5); (middle) HYDRUS 2D software; (right) our DRW algorithm.

Method	MB ( $\Delta t = 10$ sec)
GRW algorithm	63.93%
HYDRUS 2D simulation	62.45%
DRW algorithm	<b>73.07%</b>

Table 11: MB results of three methods at  $x = 0.5$  m.

#### 5.4 A 3-D Benchmark Problem with Analytical Solutions

Lastly, we consider a 3-D water infiltration example where the analytical solution exists [53]. In this example,  $V$  is a 3-D cuboid  $[0, a] \times [0, b] \times [0, c]$ . The hydraulic conductivity function follows the Gardner's model [11] (see Table 1). The initial condition is given by:

$$\psi(x, y, z, t = 0) = h_r,$$

where  $h_r$  is a constant. And the boundary condition is given by:

$$\psi(x, y, z = c, t) = \frac{1}{\alpha} \ln \left[ \exp(\alpha h_r) + \bar{h}_0 \sin \frac{\pi x}{a} \sin \frac{\pi y}{b} \right],$$

where  $\bar{h}_0 = 1 - \exp(\alpha h_r)$ . Ignoring the sink term, the pressure head solution for this problem was derived in [53] as:

$$\psi = \frac{1}{\alpha} \ln \left\{ \exp(\alpha h_r) + \bar{h}_0 \sin \frac{\pi x}{a} \sin \frac{\pi y}{b} \exp \left( \frac{\alpha(c-z)}{2} \right) \left[ \frac{\sinh \beta z}{\sinh \beta c} + \frac{2}{zd} \sum_{k=1}^{\infty} (-1)^k \frac{\lambda_k}{\gamma} \sin(\lambda_k z) \exp(-rt) \right] \right\}, \quad (25)$$

where  $d = \frac{\alpha(\theta_s - \theta_r)}{K_s}$ ,  $\lambda_k = \frac{k\pi}{c}$ ,  $\gamma = \frac{\lambda_k^2 + \beta^2}{c}$  and  $\beta = \sqrt{\frac{\alpha^2}{4} + \left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2}$ .

The infinite series in Equation (5.4) is convergent by alternating series test, and we consider only the first 1000 terms of this series. Note from Equation (5.4) that the analytical solution depends only on the saturated ( $\theta_s$ ) and residual soil moisture content ( $\theta_r$ ). And the Mualem-van Genuchten correlation [9, 10] tabulated in Table 1 was used for the water retention curve  $\theta(\psi)$ . The constants and parameters used in this case study are listed in Table 12.

Our goal is to compare the accuracy of our DRW algorithm with GRW solvers using this analytical solution as the benchmark. To the best of our knowledge, there exists no GRW solver for 3-D Richards equation in the literature. Thus, we develop our own 3-D GRW-based solver in house

Property	Symbol	Value	Units
Saturated hydraulic conductivity	$K_s$	1.1	m/s
Saturated soil moisture	$\theta_s$	0.5	–
Residual soil moisture	$\theta_r$	0	–
Parameter in Gardner’s model	$\alpha$	0.1	$\text{m}^{-1}$
Parameter in initial and boundary conditions	$h_r$	–15.24	m
Length of $V$	$a$	2	m
Width of $V$	$b$	2	m
Depth of $V$	$c$	2	m
Total time	$T$	86,400	sec

Table 12: Soil-specific parameters and constants used in the 3-D case study.

to allow for comparison. We obtain 1,734 original reference solutions using the in-house developed 3-D GRW solver [42] that implements the standard linearization scheme with the optimal static linearization parameter of 0.5. Then, data augmentation is applied by introducing Gaussian noise, resulting in a total of 8,820 data points (which include the original reference solutions) for neural network training. For both GRW and DRW algorithms, we set the tolerance to be  $10^{-9}$ , which can be achieved in less than 500 iterations for each time step.

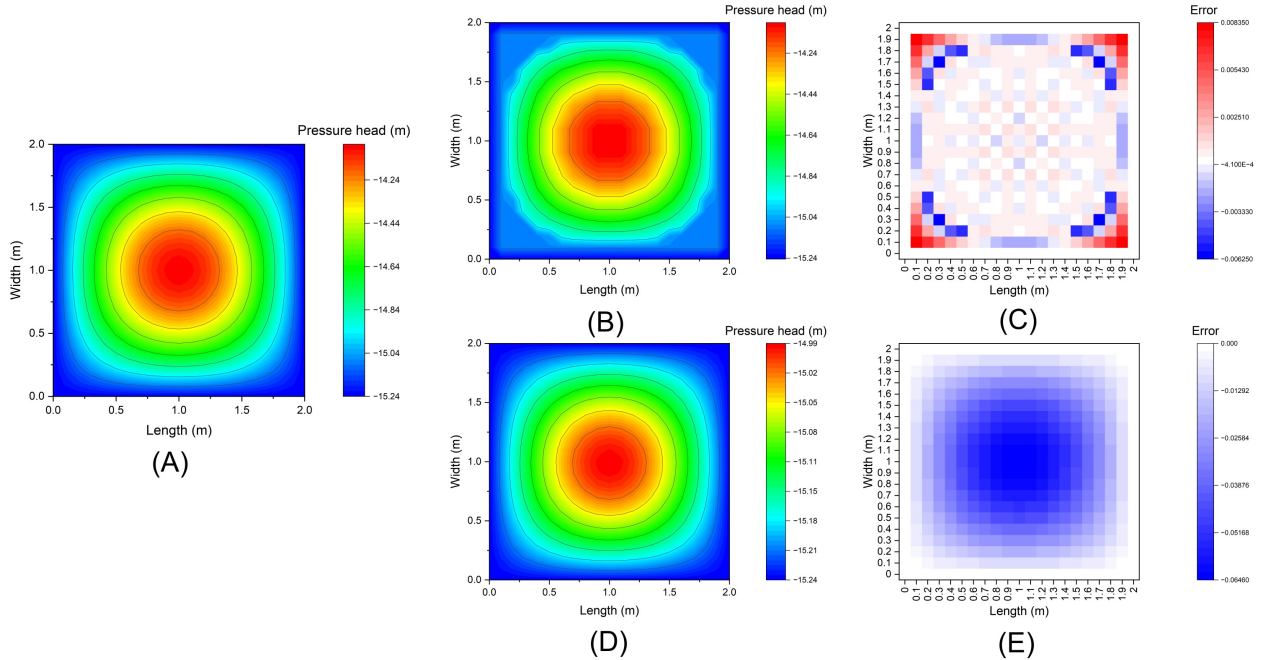


Figure 12: Pressure head solution at  $z = 0.5$  m of different methods: (A) analytical solution, (B) DRW algorithm, (C) the relative error between analytical and DRW based solutions, (D) GRW solver ( $L = 0.5$ ) and (E) the relative error between analytical solution and GRW based solutions.

We examine and compare the pressure head solutions at  $z = 0.5$  and 1m, which are shown in Figure 12 and 13, respectively. We quantify the differences between the numerical solution and the analytical solution by  $\frac{\psi_{\text{analytical}} - \psi_{\text{numerical}}}{\psi_{\text{analytical}}}$ . From the relative difference heat map of Figure 12c,e and 13c,e, we observe that, first, the magnitude of relative error of our DRW based solver is significantly

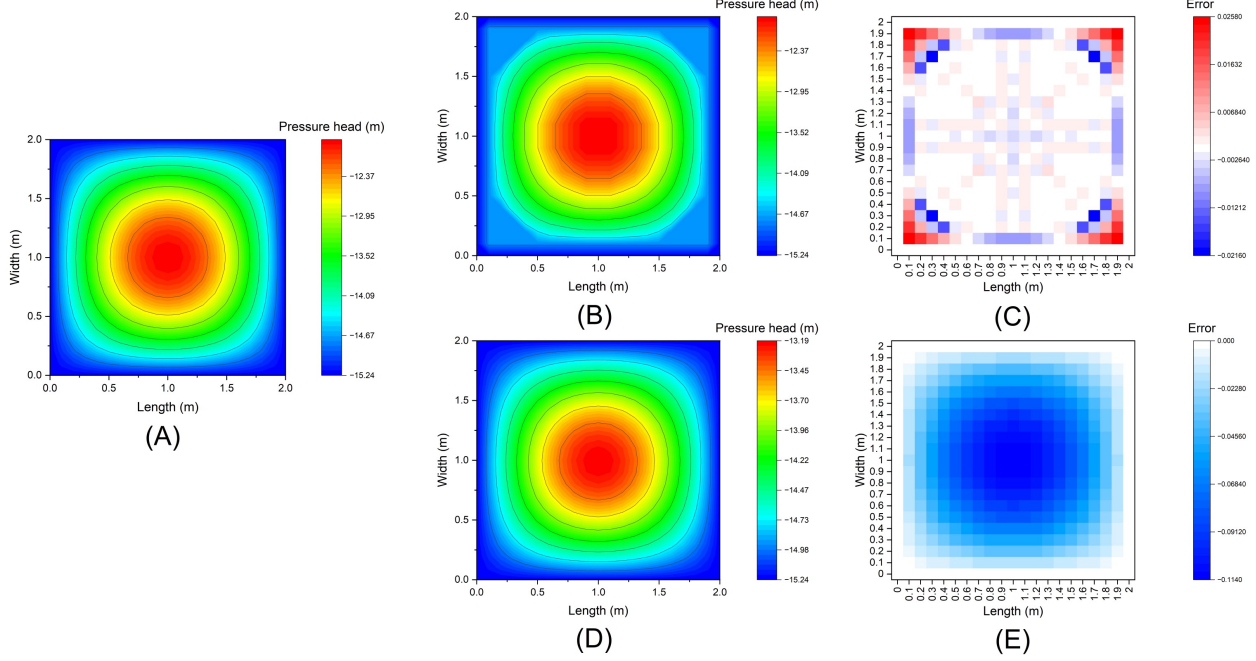


Figure 13: Pressure head solution at  $z = 1.0$  m of different methods: (A) analytical solution, (B) DRW algorithm, (C) the relative error between analytical and DRW based solutions, (D) GRW solver ( $L = 0.5$ ) and (E) the relative error between analytical solution and GRW based solutions.

lower than that of GRW based solver. And second, the largest relative error of our DRW pressure head solution occurs around the four corners of the  $x$ - $y$  domain, whereas the largest relative error of GRW solution occurs in the center of the  $x$ - $y$  domain. Furthermore, in each cell, the relative error of GRW-based pressure head solution is always non-positive, whereas that of DRW based pressure head solution can be positive or negative.

Here, we provide some justifications to these observations. First, for conventional GRW solver that embeds the standard linearization scheme formulation, we observe from Equations (8) that:

$$\psi_{\text{analytical}} - \psi_{\text{numerical}} \propto \left\{ \sum_{j \in \mathcal{N}_i} [K(\psi) \nabla(\psi + z)]_{\omega_{i,j}}^{m+1,s} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} - \partial_t \theta_i^{m+1} \text{vol}(V_i) \right\},$$

for any  $s$ , discretized cell  $V_i$ , and discretized time step  $m$ . Since the hydraulic conductivity function is positive and symmetric along  $x = 1$  m and  $y = 1$  m, and  $\nabla \psi|_{\omega^+ := [0,1] \times [0,1] \times z} = -\nabla \psi|_{\omega^- := [1,2] \times [1,2] \times z}$ , we have  $\sum_{j \in \mathcal{N}_i} [K(\theta(\psi)) \nabla(\psi + z)]_{\omega_{i,j}}^{m+1,s} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}} > 0$ . Meanwhile,  $\partial_t \theta_i^{m+1}(\psi) \text{vol}(V_i)$  is typically small due to the slow dynamics of water infiltration in soil and the fact that  $\text{vol}(V_i)$  is small. Thus, we have  $\psi_{\text{analytical}} - \psi_{\text{numerical}} > 0$  for GRW solution, which explains the negativity of the relative error. On the other hand, for our DRW algorithm, the use of neural networks to approximate the nonlinear maps  $f$  and  $f^{-1}$  complicates the behavior (including the sign) of the relative error.

Regarding the distribution of the magnitude of relative error in the GRW solver, since hydraulic conductivity function is an increasing function of  $\psi$ , and  $\psi$  is at its maximum at the center of the  $x$ - $y$  plain, it is expected that  $\sum_{j \in \mathcal{N}_i} [K(\psi) \nabla(\psi + z)]_{\omega_{i,j}}^{m+1,s} \cdot \mathbf{n}_{\omega_{i,j}} A_{\omega_{i,j}}$ , and hence the relative error, is maximized at and around the center of the  $x$ - $y$  plane. However, for the DRW based pressure head solution, we suspect that the higher relative error at the four corners may be attributed to the slight decrease in accuracy of neural networks in approximating  $f$  and  $f^{-1}$  near the boundaries of



domain.

Finally, we evaluate the MSE by summing the individual MSE values over all cells at  $z = 0.5$  m and  $z = 1$  m. For  $z = 0.5$  m,  $\text{MSE}_{\text{DRW}}$  and  $\text{MSE}_{\text{GRW}}$  are calculated to be  $8.044 \times 10^{-4}$  and 0.1972, respectively. For  $z = 1$  m,  $\text{MSE}_{\text{DRW}}$  and  $\text{MSE}_{\text{GRW}}$  are  $6.736 \times 10^{-3}$  and 0.5052, respectively. This indicates that the MSE of the GRW based pressure head solution is typically 1 to 2 orders of magnitude higher than our DRW based solution.

## 6 A Realistic Case Study

Finally, we consider a real-world case study adopted from Orouskhani et al. [55], where infiltration, irrigation, and root water extraction take place in circular agricultural field, equipped with a center-pivot irrigation system with a radius of 50 m, located at Lethbridge, Alberta. Soil moisture sensors are inserted at a depth of 25 cm across 20 different locations in this field to collect soil moisture data every 30 min from June 19 to August 13, 2019. To validate our DRW algorithm in solving 3-D real-world applications, we select one of the 20 locations where the Mualem-van Genuchten WRC and HCF model parameters are identified and given in [55]. We consider a cylindrical control volume  $V$  with a radius of 0.1 m and depth of 25 cm. We discretize  $V$  into 6, 40 and 22 nodes in the radial, azimuthal and axial directions, respectively. The time step size  $\Delta t$  is determined using the heuristic formula in [42]. Thus, we reformulate Equation (21) in cylindrical coordinate system as:

$$n_i^{m+1,s+1} = n_i^{m+1,s} + \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \hat{\mathbf{e}}_j \cdot \mathbf{n}_{\omega_{i,j}} \frac{n_j^{m+1,s} - n_i^{m+1,s}}{\text{dist}(V_j, V_i)} A_{\omega_{i,j}} + f^{-1}(J),$$

where  $\hat{\mathbf{e}}_j = (1, \frac{1}{r_j^2}, 1)^T$  and

$$\begin{aligned} J = & \frac{1}{L_i^{m+1,s}} \sum_{j \in \mathcal{N}_i} K_{\omega_{i,j}}^{m+1,s} \hat{\mathbf{e}}_j \cdot \mathbf{n}_{\omega_{i,j}} \frac{z_j^{m+1,s} - z_i^{m+1,s}}{\text{dist}(V_j, V_i)} A_{\omega_{i,j}} - \frac{1}{L_i^{m+1,s}} \frac{\theta_i^{m+1,s} - \theta_i^m}{\Delta t} \text{vol}(V_i) \\ & - \frac{1}{L_i^{m+1,s}} S(\psi_i^{m+1,s}) \text{vol}(V_i). \end{aligned} \quad (26)$$

Here, the sink term in  $S$  follows the Feddes model [61]:

$$S = \sigma(\psi) S_{\max}, \quad (27)$$

where  $S_{\max}$  is the maximum possible root extraction rate and  $\sigma$  denotes a dimensionless water stress reduction factor (see [48] for the detailed formulation).

The boundary conditions are given by:

$$\begin{aligned} \frac{\partial \psi(r, \omega, z)}{\partial r} &= 0 \quad \text{at } r = 0 \text{ m,} \\ \frac{\partial \psi(r, \omega, z)}{\partial r} &= 0 \quad \text{at } r = 0.1 \text{ m,} \\ \frac{\partial \psi(r, \omega, z)}{\partial z} &= 0 \quad \text{at } z = 0 \text{ cm,} \\ \frac{\partial \psi(r, \omega, z)}{\partial z} &= -1 - \frac{u_{\text{irr}}}{K(\psi)} \quad \text{at } z = 25 \text{ cm,} \\ \psi(r, \omega = 0, z) &= \psi(r, \omega = 2\pi, z), \end{aligned}$$

where  $u_{\text{irr}}$  is the irrigation rate (in m/s). And the initial condition is simply:

$$\psi(x, y, z, t = 0) = h_r,$$

where  $h_r$  is the starting pressure head recording.

Note that the boundary conditions are time-dependent due to  $u_{\text{irr}}$ . This poses a potential computational challenge as the neural networks typically need to be retrained whenever the initial or boundary conditions change [62, 63]. To overcome this practical challenge, we adopt a new approach of training the two neural networks with 3,000 epochs based on the boundary conditions for June 19, 2019 (no irrigation) when data collection began. Then, the trained weights within these two neural networks serve as the starting point for retraining when a new set of boundary conditions is adopted. This way, only 500 epochs are sufficient to retrain the neural networks. For each set of boundary conditions, we obtain the training set containing 84,480 reference solutions. In addition, the dataset provided by [55], after performing data augmentation by introducing Gaussian noises, is also included in our training dataset. Each neural network, which has 5 hidden layers with 256 neurons in each layer, is trained using SGD optimizer with a learning rate of 0.001. We set the stopping criterion to be  $\text{RE}_s = 10^{-9}$ , which can be achieved well within 500 iterations.

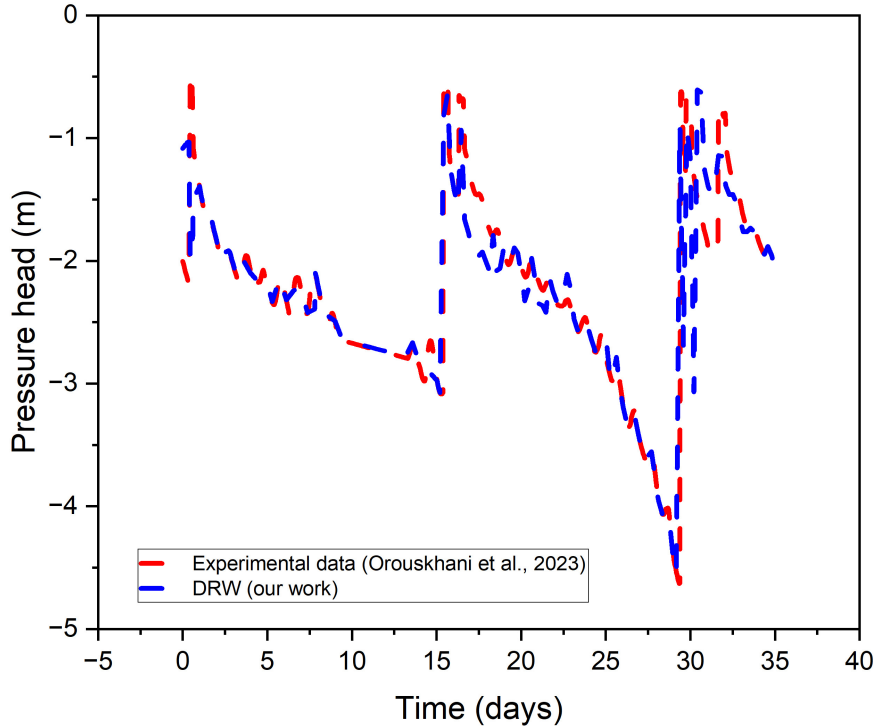


Figure 14: Comparison of pressure head profile at  $z = 25$  cm in a selected 0.1-m radius region (averaged for all  $6 \times 40 = 240$  cells at  $z = 25$  cm) in the field.

For this problem, we simulate the pressure head from 1:00 am on June 19, 2019 to 5:00 pm on July 28, 2019. As mentioned in [55], there are two irrigation instances between this time frame, one is on July 4 (the 15th day, 1.81 mm) and the other is on July 18 (the 30th day, 1.58 mm). Figure 14 shows the pressure head solution profile obtained by our DRW algorithms compared to the experimental measurements provided by Orouskhani et al. [55] over the course of 35 days. We observe that, most of the time, the our DRW solutions match with the experimental measurements very well. The only major mismatches between experimental measurements and DRW solutions occur on the 15th

day and the 30th day, which correspond to the time when the irrigation takes place. We believe both mismatches are a result of our simplifying assumption regarding the irrigation schedule. Due to the limited information we have about the exact irrigation schedule and intensity, we have to assume that both irrigation instances happened over the course of the entire day. Thus, we simply divide the irrigation amount by 86,400 seconds to obtain  $u_{\text{irr}}$ . However, in reality, the irrigation may finish within 24 hours, thus causing the mismatches. With more accurate  $u_{\text{irr}}$  model, our DRW algorithm is expected to produce highly accurate solutions that closely match with experimental measurements at all times. This makes our DRW algorithm an accurate and scalable numerical framework to solve Richards equation over a long period of time.

## 7 Conclusion

In this work, we present a novel data-driven solution algorithm named DRW for accurately and efficiently solving a general,  $d$ -dimensional Richards equation ( $d = 1, 2, 3$ ). Our DRW algorithm adopts an adaptive linearization scheme based on the FVM discretization of the Richards equation, which significantly improves convergence and stability of the solution process. The DRW framework then solves the adaptive linearization scheme formulation iteratively in a data-driven global random walk model. The DRW algorithm also uses two neural networks to accurately learn the nonlinear forward and inverse maps between pressure head and the number of particles in each cell. These neural networks are incorporated in the global random walk model embedded in the DRW framework to achieve synergistic improvement in solution accuracy. Furthermore, we also discuss effective ways, such as the “coarse-to-fine” approach, to perform data augmentation to facilitate neural network training using just a small number of low-fidelity reference solutions as training set. Overall, these innovative techniques work together seamlessly to improve the convergence and accuracy of DRW algorithm in solving the Richards equation. Indeed, via several 1-D through 3-D case studies that span across benchmark problems and real-world applications, we demonstrate that, compared to state-of-the-art numerical solvers, our DRW algorithm not only achieves significantly improved accuracy and convergence, but also better preserves the overall mass balance and conservation laws while being computationally efficient to implement. Moreover, the proposed data-driven numerical method is expected to be a generalizable computational framework for modeling a wide range of applications, including fractional diffusion [64], fluid transport in fibrous porous materials [65, 66, 67], liquid extraction from saturated granular materials [68], water and hydrocarbon flow in petroleum reservoirs [69], and so on.

## 8 Acknowledgments

We acknowledge financial support from the startup fund of College of Engineering, Architecture, and Technology at Oklahoma State University.

## References

- [1] National Academy of Engineering and National Academies of Sciences, Engineering, and Medicine, *New Directions for Chemical Engineering*, The National Academies Press, Washington, DC, 2022.
- [2] E. Babaeian, M. Sadeghi, S. B. Jones, C. Montzka, H. Vereecken, M. Tuller, Ground, proximal, and satellite remote sensing of soil moisture, *Reviews of Geophysics* 57 (2) (2019) 530–616.

- [3] D. Spelman, K.-D. Kinzli, T. Kunberger, Calibration of the 10HS soil moisture sensor for Southwest Florida agricultural soils, *Journal of Irrigation and Drainage Engineering* 139 (12) (2013) 965–971. doi:10.1061/(ASCE)IR.1943-4774.0000647.
- [4] M. Saavoss, J. Majsztrik, B. Belayneh, J. Lea-Cox, E. Lichtenberg, Yield, quality and profitability of sensor-controlled irrigation: a case study of snapdragon (*Antirrhinum majus* L.) production, *Irrigation Science* 34 (5) (2016) 409–420.
- [5] L. A. Richards, Capillary conduction of liquids through porous mediums, *Physics* 1 (5) (1931) 318–333.
- [6] J.-G. Caputo, Y. A. Stepanyants, Front solutions of Richards’ equation, *Transport in Porous Media* 74 (1) (2008) 1–20.
- [7] R. E. Smith, K. Smettem, P. Broadbridge, D. Woolhiser, *Infiltration Theory for Hydrologic Applications*, American Geophysical Union, 2002.
- [8] R. Haverkamp, M. Vauclin, J. Touma, P. Wierenga, G. Vachaud, A comparison of numerical simulation models for one-dimensional infiltration, *Soil Science Society of America Journal* 41 (2) (1977) 285–294.
- [9] Y. Mualem, A new model for predicting the hydraulic conductivity of unsaturated porous media, *Water Resources Research* 12 (3) (1976) 513–522.
- [10] M. T. Van Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Science Society of America Journal* 44 (5) (1980) 892–898.
- [11] W. Gardner, Some steady-state solutions of the unsaturated moisture flow equation with application to evaporation from a water table, *Soil Science* 85 (4) (1958) 228–232.
- [12] M. W. Farthing, F. L. Ogden, Numerical solution of Richards’ equation: A review of advances and challenges, *Soil Science Society of America Journal* 81 (6) (2017) 1257–1269.
- [13] V. S. Sizikov, et al., *Well-posed, ill-posed, and intermediate problems with applications*, De Gruyter, 2011.
- [14] L. C. Evans, *Partial differential equations*, Vol. 19, American Mathematical Soc., 2010.
- [15] W. Merz, P. Rybka, Strong solutions to the richards equation in the unsaturated zone, *Journal of Mathematical Analysis and Applications* 371 (2) (2010) 741–749.
- [16] O. Misiats, K. Lipnikov, Second-order accurate monotone finite volume scheme for Richards’ equation, *Journal of Computational Physics* 239 (2013) 123–137.
- [17] N. Abdellatif, C. Bernardi, M. Touihri, D. Yakoubi, A priori error analysis of the implicit Euler, spectral discretization of a nonlinear equation for a flow in a partially saturated porous media, *Advances in Pure and Applied Mathematics* 9 (1) (2018) 1–27.
- [18] P. R. Day, J. N. Luthin, A numerical solution of the differential equation of flow for a vertical drainage problem, *Soil Science Society of America Journal* 20 (4) (1956) 443–447.
- [19] M. A. Celia, E. T. Bouloutas, R. L. Zarba, A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resources Research* 26 (7) (1990) 1483–1496.

- [20] M. W. Farthing, F. L. Ogden, Numerical solution of Richards' equation: A review of advances and challenges, *Soil Science Society of America journal* 81 (6) (2017) 1257–1269.
- [21] B. Belfort, A. Younes, M. Fahs, F. Lehmann, On equivalent hydraulic conductivity for oscillation-free solutions of richard's equation, *Journal of Hydrology* 505 (2013) 202–217.
- [22] A. M. Ireson, R. J. Spiteri, M. P. Clark, S. A. Mathias, A simple, efficient, mass-conservative approach to solving Richards' equation (openre, v1. 0), *Geoscientific Model Development* 16 (2) (2023) 659–677.
- [23] D. Or, P. Lehmann, S. Assouline, Natural length scales define the range of applicability of the richards equation for capillary flows, *Water Resources Research* 51 (9) (2015) 7130–7144.
- [24] K. Roth, Scaling of water flow through porous media and soils, *European Journal of Soil Science* 59 (1) (2008) 125–130.
- [25] H.-J. Vogel, O. Ippisch, Estimation of a critical spatial discretization limit for solving richards' equation at large scales, *Vadose Zone Journal* 7 (1) (2008) 112–114.
- [26] K. Rathfelder, L. M. Abriola, Mass conservative numerical solutions of the head-based richards equation, *Water Resources Research* 30 (9) (1994) 2579–2586.
- [27] Z. Song, Z. Jiang, A data-driven modeling approach for water flow dynamics in soil, in: A. C. Kokossis, M. C. Georgiadis, E. Pistikopoulos (Eds.), 33rd European Symposium on Computer Aided Process Engineering, Vol. 52 of *Computer Aided Chemical Engineering*, Elsevier, 2023, pp. 819–824. doi:<https://doi.org/10.1016/B978-0-443-15274-0.50131-1>. URL <https://www.sciencedirect.com/science/article/pii/B9780443152740501311>
- [28] Z. Song, Z. Jiang, A data-driven random walk approach for solving water flow dynamics in soil systems, in: *Proceedings of Foundations of Computer-Aided Process Operations and Chemical Process Control (FOCAPO/CPC)*, 2023, pp. 1–6.
- [29] R. Eymard, M. Gutnic, D. Hilhorst, The finite volume method for Richards equation, *Computational Geosciences* 3 (3) (1999) 259–294.
- [30] W. Lai, F. L. Ogden, A mass-conservative finite volume predictor–corrector solution of the 1D Richards' equation, *Journal of Hydrology* 523 (2015) 119–127.
- [31] S. Bassetto, C. Cancès, G. Enchéry, Q.-H. Tran, On several numerical strategies to solve Richards' equation in heterogeneous media with finite volumes, *Computational Geosciences* 26 (5) (2022) 1297–1322.
- [32] D. Caviedes-Voullième, P. Garcı, J. Murillo, et al., Verification, conservation, stability and efficiency of a finite volume method for the 1D Richards equation, *Journal of Hydrology* 480 (2013) 69–84.
- [33] G. Manzini, S. Ferraris, Mass-conservative finite volume methods on 2-D unstructured grids for the Richards' equation, *Advances in Water Resources* 27 (12) (2004) 1199–1215.
- [34] L. Bergamaschi, M. Putti, Mixed finite elements and Newton-type linearizations for the solution of Richards' equation, *International Journal for Numerical Methods in Engineering* 45 (8) (1999) 1025–1046.

- [35] I. S. Pop, F. Radu, P. Knabner, Mixed finite elements for the richards' equation: linearization procedure, *Journal of Computational and Applied Mathematics* 168 (1-2) (2004) 365–373.
- [36] K. Mitra, I. S. Pop, A modified L-scheme to solve nonlinear diffusion problems, *Computers & Mathematics with Applications* 77 (6) (2019) 1722–1738.
- [37] G. Albuja, A. I. Ávila, A family of new globally convergent linearization schemes for solving Richards' equation, *Applied Numerical Mathematics* 159 (2021) 281–296.
- [38] V. Casulli, P. Zanolli, A nested Newton-type algorithm for finite volume methods solving Richards' equation in mixed form, *SIAM Journal on Scientific Computing* 32 (4) (2010) 2255–2273.
- [39] V. Zaburdaev, S. Denisov, J. Klafter, Lévy walks, *Reviews of Modern Physics* 87 (2) (2015) 483.
- [40] A. Zoia, M.-C. Néel, A. Cortis, Continuous-time random-walk model of transport in variably saturated heterogeneous porous media, *Physical Review E* 81 (3) (2010) 031104.
- [41] N. Suciu, *Diffusion in Random Fields: Applications to Transport in Groundwater*, Springer, 2019.
- [42] N. Suciu, D. Illiano, A. Prechtel, F. A. Radu, Global random walk solvers for fully coupled flow and transport in saturated/unsaturated porous media, *Advances in Water Resources* 152 (2021) 103935.
- [43] N. Suciu, C. Vamoş, I. Turcu, C. Pop, L. Ciortea, Global random walk modelling of transport in complex systems, *Computing and Visualization in Science* 12 (2) (2009) 77–85.
- [44] C. Vamos, N. Suciu, H. Vereecken, O. Nitzsche, H. Hardelauf, Global random walk simulations of diffusion, in: *Scientific Computing, Validated Numerics, Interval Methods*, Springer, 2001, pp. 343–354.
- [45] R. Zarba, *A Numerical Investigation of Unsaturated Flow*, Massachusetts Institute of Technology, Department of Civil Engineering, Cambridge, MA, 1988.
- [46] M. A. Celia, R. Zarba, A comparative study of numerical solutions for unsaturated flow, in: S. N. Atluri, G. Yagawa (Eds.), *Computational Mechanics '88*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988, pp. 1659–1662.
- [47] P. Broadbridge, E. Daly, J. Goard, Exact solutions of the richards equation with nonlinear plant-root extraction, *Water Resources Research* 53 (11) (2017) 9679–9691.
- [48] B. T. Agyeman, S. Bo, S. R. Sahoo, X. Yin, J. Liu, S. L. Shah, Soil moisture map construction using microwave remote sensors and sequential data assimilation (2020). [arXiv:2010.07037](https://arxiv.org/abs/2010.07037). URL <https://arxiv.org/abs/2010.07037>
- [49] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* 4 (2) (1991) 251–257.
- [50] A. Pinkus, Approximation theory of the MLP model in neural networks, *Acta Numerica* 8 (1999) 143–195. doi:10.1017/S0962492900002919.

- [51] I. B. V. Da Silva, P. J. Adeodato, PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets, in: The 2011 International Joint Conference on Neural Networks, IEEE, 2011, pp. 2664–2669.
- [52] D. Gašiorowski, T. Kolerski, Numerical solution of the two-dimensional Richards equation using alternate splitting methods for dimensional decomposition, *Water* 12 (6) (2020) 1780.
- [53] F. T. Tracy, Clean two-and three-dimensional analytical solutions of Richards’ equation for testing numerical solvers, *Water Resources Research* 42 (8) (2006).
- [54] M. Berardi, F. Difonzo, M. Vurro, L. Lopez, The 1d Richards’ equation in two layered soils: a filippov approach to treat discontinuities, *Advances in Water Resources* 115 (2018) 264–272.
- [55] E. Orouskhani, S. Sahoo, B. Agyeman, S. Bo, J. Liu, Impact of sensor placement in soil water estimation: a real-case study, *Irrigation Science* 41 (3) (2023) 395–411.
- [56] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. ICML, Vol. 30, Atlanta, Georgia, USA, 2013, p. 3.
- [57] T. Bandai, T. A. Ghezzehei, 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 (2) (2021) e2020WR027642.
- [58] R. G. Hills, I. Porro, D. B. Hudson, P. J. Wierenga, Modeling one-dimensional infiltration into very dry soils: 1. model development and evaluation, *Water Resources Research* 25 (6) (1989) 1259–1269.
- [59] R. F. Carsel, R. S. Parrish, Developing joint probability distributions of soil water retention characteristics, *Water Resources Research* 24 (5) (1988) 755–769.
- [60] J. Šimůnek, M. T. Van Genuchten, M. Šejna, Recent developments and applications of the HYDRUS computer software packages, *Vadose Zone Journal* 15 (7) (2016).
- [61] R. Feddes, H. Zaradny, Model for simulating soil-water content considering evapotranspiration — comments, *Journal of Hydrology* 37 (3) (1978) 393–397.
- [62] R. Matthey, S. Ghosh, A novel sequential method to train physics informed neural networks for allen cahn and cahn hilliard equations, *Computer Methods in Applied Mechanics and Engineering* 390 (2022) 114474. doi:10.1016/j.cma.2021.114474.  
URL <http://dx.doi.org/10.1016/j.cma.2021.114474>
- [63] R. Brecht, L. Bakels, A. Bihlo, A. Stohl, Improving trajectory calculations by flexpart 10.4+ using single-image super-resolution, *Geoscientific Model Development* 16 (8) (2023) 2181–2192.
- [64] E. Gerolymatou, I. Vardoulakis, R. Hilfer, Modelling infiltration by means of a nonlinear fractional diffusion model, *Journal of Physics D: Applied Physics* 39 (18) (2006) 4104.
- [65] Y. Jabbari, E. Tsotsas, C. Kirsch, A. Kharaghani, Determination of the moisture transport coefficient from pore network simulations of spontaneous imbibition in capillary porous media, *Chemical Engineering Science* 207 (2019) 600–610.
- [66] A. Ashari, H. Vahedi Tafreshi, A two-scale modeling of motion-induced fluid release from thin fibrous porous media, *Chemical Engineering Science* 64 (9) (2009) 2067–2075.

- [67] A. H. Tavangarrad, B. Mohebbi, C. Qin, S. M. Hassanizadeh, R. Rosati, J. Claussen, B. Blümich, Continuum-scale modeling of water infiltration into a stack of two thin fibrous layers and their inter-layer space, *Chemical Engineering Science* 207 (2019) 769–779.
- [68] M. Khammar, Y. Xu, Continuous liquid extraction from saturated granular materials, *Chemical Engineering Science* 173 (2017) 390–401.
- [69] Soledad Fioroni, A. E. Larreteguy, G. B. Savioli, An OpenFOAM Application for Solving the Black Oil Problem, *Mathematical Models and Computer Simulations* 13 (5) (2021) 907–918.