16th World Congress in Computational Mechanics (WCCM)
21-26 July 2024, Vancouver, Canada
A. Korobenko, M. Laforest, S. Prudhomme, R. Vaziri (Eds)

PHYSICS-INFORMED NEURAL NETWORKS VS FINITE ELEMENT METHOD FOR MODELING COUPLED WATER AND SOLUTE FLOW IN UNSATURATED SOILS

HAMZA KAMIL[1,2], AZZEDDINE SOULAÏMANI[2], AND ABDELAZIZ BELJADID[1,3]

[1] University Mohammed VI Polytechnic
Lot 660, Ben Guerir 43150, Morocco
hamza.kamil@um6p.ma

[2] École de technologie supérieure
1100 Notre-Dame St W, Montreal, Quebec H3C 1K3, Canada
azzeddine.soulaimani@etsmtl.ca

[3] University of Ottawa
75 Laurier Ave E, Ottawa, Ontario, ON K1N 6N5, Canada
abdelaziz.beljadid@um6p.ma

Key words: Unsaturated Soils, Water Flow, Richards equation, Solute Transport, Physics-Informed Neural Networks, Finite Element Method.

Abstract. Accurate modeling of water infiltration and solute transport in unsaturated soils is critical for various applications. These include optimizing irrigation practices to conserve water and minimize environmental impact, as well as predicting the fate of contaminants in soil and groundwater. This study explores the application of the vanilla physics-informed neural network (PINN) approach for modeling the coupled system of water flow and solute transport in unsaturated soils. We compare the performance of PINN with the Galerkin finite element method (FEM) to evaluate their effectiveness. Various techniques are implemented to improve the PINN solver, including adaptive activation functions. Numerical tests were carried out to evaluate the efficiency of the PINN solver in comparison to the FEM. The findings reveal that PINN can achieve accuracy comparable to FEM, albeit at a significantly higher computational cost during training, while maintaining fast inference times.

1 INTRODUCTION

Soil moisture and solute dynamics are fundamental to hydrological modeling, influencing a wide range of environmental and engineering challenges. Precise knowledge of water flow and solute behavior in soils is essential to tackle diverse environmental problems. Understanding these dynamics is critical for developing efficient irrigation systems [1], which ultimately leads to improved agricultural product quality. Modeling water flow and solute transport in unsaturated soils traditionally relies on a coupled system of the Richards equation [16] for water flow, and advection-dispersion equations [17] to describe solute movement. This approach necessitates solving a complex system of partial differential equations (PDEs). The highly nonlinear nature of the Richards equation, coupled with the intricate nature of realistic boundary conditions, renders analytical solutions for the coupled water flow and solute transport model highly challenging to obtain. Consequently, numerical methods provide a powerful alternative approach to solving these PDEs. While various numerical methods exist for solving the Richards equation alone [18], their application to the coupled model of water flow and solute transport remains less explored [19, 23]. Standard numerical solvers often face challenges due to their computational demands, particularly during mesh generation within the domain. Additionally, these methods require adherence to strict time step and mesh size limitations to ensure accurate solutions. Furthermore, they typically assume precise knowledge of initial/boundary conditions and soil properties, which can be unrealistic in practical scenarios where uncertainty and incomplete information about these conditions are common.

Data-driven approaches offer an alternative surrogate to standard numerical methods. A recent method, introduced as physics-informed neural networks (PINNs), aims to enhance efficiency and alleviate the requirement for extensive training data [20]. In the PINN approach, a deep learning solver is trained to solve a desired PDE while adhering to the underlying physical laws encoded within it. This methodology has demonstrated significant advancements in solving various engineering problems (e.g., [21]). Other variants of PINNs have been proposed to enhance the approach's efficiency in tackling complex PDEs with discontinuous and sharp solutions [22].

PINN solvers have emerged as powerful tools for simulating water flow and solute transport, consistently achieving impressive results. For instance, Depina et al. [2] employed PINNs to estimate unknown soil parameters, while Bandai et al. [3] developed a PINN framework to estimate key soil properties such as the water retention curve, hydraulic conductivity function, and surface water flux. Furthermore, Haruzi et al. [15] addressed the coupled model to predict pressure head and pore water electrical conductivity with unknown initial conditions.

A critical gap persists in the literature concerning coupled models for water flow and solute transport, particularly in scenarios where no data are available. Building upon prior

comparisons between the FEM and PINNs for various PDEs [14], this study specifically concentrates on a comparative analysis of their efficacy, with a focus on accuracy and efficiency, in solving the forward problem of the highly nonlinear Richards equation coupled with the advection-dispersion equation. We integrate the adaptive activation function [4] to enhance the training of the PINN solver. Two numerical tests, namely, 1D and 2D tests, are conducted to provide a detailed comparison of the two approaches.

2 THE PHYSICAL MODELS

2.1 Water flow

Modeling water flow in unsaturated soils is typically described using the Richards equation [16], which is represented by the following equation:

$$\begin{cases} \dfrac{\partial \theta}{\partial t} + \nabla \cdot \mathbf{q} = 0, \\ \mathbf{q} = -K \nabla (\Psi + z), \end{cases} \tag{1}$$

where Ψ is the pressure head, z is the vertical dimension, $\mathbf{q}$ is the water flux, $K = K_s k_r$ is the unsaturated hydraulic conductivity, K_s is the saturated hydraulic conductivity, k_r is the water relative permeability, and θ is the volumetric water content.

To solve the Richards equation, it is essential to provide explicit functions for K and θ as functions of Ψ. In this paper, we employed the Mualem-van Genuchten model [5] expressed by the following equations:

$$\theta = \begin{cases} \theta_r + (\theta_s - \theta_r)\left(1 + (\alpha_v |\Psi|)^{n_v}\right)^{-m_v}, & \text{if } \Psi \leq 0 \\ \theta_s, & \text{if } \Psi > 0 \end{cases} \tag{2a}$$

$$K = \begin{cases} K_s \dfrac{\left[1 - (\alpha_v |\Psi|)^{n_v - 1}\left(1 + (\alpha_v |\Psi|)^{n_v}\right)^{-m_v}\right]^2}{\left(1 + (\alpha_v |\Psi|)^{n_v}\right)^{m_v/2}}, & \text{if } \Psi \leq 0 \\ K_s, & \text{if } \Psi > 0 \end{cases} \tag{2b}$$

$$m_v = 1 - \frac{1}{n_v}\,;\ n_v > 1, \tag{2c}$$

where θ_s and θ_r represent the saturated and residual volumetric water contents, respectively; $n_v,\ m_v$ and α_v are the van Genuchten empirical parameters.

2.2 Solute transport

To account for solute transport alongside water flow, the equilibrium advection-dispersion equation is typically coupled with the Richards equation, resulting in the following governing model [17]:

$$\frac{\partial \theta c}{\partial t} = \nabla \cdot (\theta \mathbf{D} \nabla c - c\mathbf{q}), \tag{3}$$

where c represents the concentration of the solute in the liquid phase, $\mathbf{D}$ represents the dispersion tensor, θ and $\mathbf{q}$ represent the water content and water flux, respectively. The components of the dispersion tensor are computed as follows [17]:

$$\theta \mathbf{D}_{ij} = (D_T \|\mathbf{q}\| + \theta D_w \tau)\, \delta_{ij} + (D_L - D_T) \frac{\mathbf{q}_i \mathbf{q}_j}{\|\mathbf{q}\|}. \tag{4}$$

Here, D_w denotes the molecular diffusion constant, while D_L and D_T represent the longitudinal and transversal dispersivities, respectively. The symbol δ_{ij} denotes the Kronecker symbol, while τ represents the tortuosity factor in the liquid phase. The Euclidean norm of $\mathbf{q}$ is denoted by $\|\mathbf{q}\|$, where $\mathbf{q}_i$ and $\mathbf{q}_j$ represent the components of the water flux in the corresponding spatial directions. The tortuosity factor is determined using the formula [17]: $\tau = \dfrac{\theta^{7/3}}{\theta_s^2}$.

3 METHODS

3.1 Finite element method: FEM

Several finite element methods are available in the literature for solving the coupled model of water flow and solute transport (e.g., [19]). In this study, we adopt the Galerkin FEM [13] for both the Richards equation and the solute transport model.

The coupled model of water flow and solute transport is studied within the time-space domain $I \times \Omega$, where $I = [0, T]$, T is the final time of simulation, and $\Omega \subset \mathbb{R}^d,\ d = 1, 2, 3$, is a connected bounded open domain representing the soil and having a smooth boundary $\partial\Omega$. To numerically solve the problem, we discretize the time interval I into smaller sub-intervals. Each sub-interval has a uniform time step of size $\Delta t > 0$. We denote the corresponding discrete time points as $t_n = n\Delta t$, where n is an integer index. We are looking for two sequences $(\Psi^n)_{n=0}^N$ and $(c^n)_{n=0}^N$ such that the following time-discrete system holds:

$$\begin{cases} \dfrac{\theta^{n+1} - \theta^n}{\Delta t} = \nabla \cdot \left[K^{n+1} \nabla (\Psi^{n+1} + z)\right], \\ \dfrac{(\theta c)^{n+1} - (\theta c)^n}{\Delta t} = \nabla \cdot \left(\theta^{n+1} \mathbf{D}^{n+1} \nabla c^{n+1} - c^{n+1} \mathbf{q}^{n+1}\right). \end{cases} \tag{5}$$

We derived the system (5) using the backward-Euler time-stepping scheme, which is first-order accurate. Subsequently, the spatial domain Ω is discretized into small elements Ω^e, which collectively form a mesh denoted by $\mathcal{T}_h$. The mesh size h characterizes the scale of this partition and is defined as $h = \max_{\mathcal{T}_h} \delta(\Omega^e)$, where $\delta(\Omega^e)$ is the diameter of the element Ω^e. The Galerkin finite element space V_h is defined as follows:

$$V_h = \Big\{ v_h \in H^1(\Omega) \; ; \; v_{h_{|\Omega^e}} \in \mathcal{P}_1(\Omega^e), \; \forall \Omega^e \in \mathcal{T}_h \Big\},$$

where $H^1(\Omega)$ is the Sobolev space of functions with first-order weak derivatives in $L^2(\Omega)$, and $\mathcal{P}_1(\Omega^e)$ represents the space of linear polynomials on element Ω^e.

We project the system (5) onto V_h, which leads to the following weak formulations:

$$\begin{cases} \langle \dfrac{\theta_h^{n+1} - \theta_h^n}{\Delta t}, v_h \rangle + \langle \; K_h^{n+1} \nabla(\Psi_h^{n+1} + z), \nabla v_h \rangle = 0, \quad \forall v_h \in V_h, \\ \langle \dfrac{(\theta_h c_h)^{n+1} - (\theta_h c_h)^n}{\Delta t}, w_h \rangle + \langle \theta_h^{n+1} \mathbf{D}_h^{n+1} \nabla c_h^{n+1} - c_h^{n+1} \mathbf{q}_h^{n+1}, \nabla w_h \rangle = 0, \quad \forall w_h \in V_h, \end{cases} \tag{6}$$

where $\langle . \, , . \rangle$ denotes the standard inner product of the functional space $L^2(\Omega)$. For the sake of notation simplicity, we neglect the boundary term that arises from Green's formula in the system (6).

The first equation in system (6) is nonlinear due to the dependence of θ_h and K_h on the primary variable Ψ_h. To handle this, we employ a linearization technique known as the modified Picard scheme [13]:

$$\langle \frac{\theta_h^{n+1,m} + C_h^{n+1,m} \left(\Psi_h^{n+1,m+1} - \Psi_h^{n+1,m} \right) - \theta_h^n}{\Delta t}, v_h \rangle + \langle \; K_h^{n+1,m} \nabla(\Psi_h^{n+1,m+1} + z), \nabla v_h \rangle = 0, \; \forall v_h \in V_h, \tag{7}$$

where $C_h = \dfrac{\mathrm{d}\theta_h}{\mathrm{d}\Psi_h}$ is the specific water capacity, and the subscript m stands for the Picard iteration. At each time step n, we verify the following inequality before progressing to the subsequent time level:

$$\left\| \left(\Psi_h^{n+1,m+1} - \Psi_h^{n+1,m} \right) \right\|_{L^2(\Omega)} < 10^{-6}, \tag{8}$$

where $\|.\|_{L^2(\Omega)}$ denotes the L^2 norm.

We implemented the FEM for the water flow and solute transport model described above using the FreeFEM++ software [6]. To mitigate the problem of oscillatory solutions in the Richards equation, we implemented a mass lumping technique [13] using the row-sum method. The resulting linear systems were solved using UMFPACK [12].

In the 1D numerical test, the following settings were used: $\Delta t = 10^{-2}$ day and $\Delta z = 10^{-2}$ m. For the 2D numerical test, the time step was $\Delta t = 10^{-2}$day, and the computational domain was discretized into a 25×25 grid.

3.2 Physics-Informed Neural Networks: PINNs

In this study, we utilize the vanilla PINN [20] to address the coupled model of water flow and solute transport in unsaturated soils. We approximate the solutions of the PDEs given by equations (1)-(3), $\Psi(t,x)$ and $c(t,x)$, using a feed-forward neural network (FNN). The outputs of the FNN are represented as $\hat{\Psi}(t,x,\Theta)$ and $\hat{c}(t,x,\Theta)$, where Θ denotes the trainable parameters of the neural network, and the inputs consist of the time-space coordinates (t,x). We gather data comprising collocation points distributed throughout the time-space domain $I \times \Omega$. These data points are classified into residual collocation points $(t^r, x^r) \in (0,T] \times \Omega$, initial collocation points $(0, x^{ic}) \in \{0\} \times \Omega$, and boundary collocation points $(t^b, x^b) \in (0,T] \times \partial\Omega$. Subsequently, the following residual functions can be derived using automatic differentiation techniques [11]:

$$\begin{cases} \hat{f}_R(t^r,x^r) = \dfrac{\partial\hat{\theta}}{\partial t} - \nabla\cdot\left[\hat{K}\nabla(\hat{\Psi}+z)\right], \\ \hat{f}_C(t^r,x^r) = \dfrac{\partial\hat{\theta}\hat{c}}{\partial t} - \nabla\cdot\left(\hat{\theta}\hat{\mathbf{D}}\nabla\hat{c} - \hat{c}\hat{\mathbf{q}}\right). \end{cases} \tag{9}$$

The hat notation denotes predictions made by the FNN. The training of the PINN solver involves minimizing the following loss function:

$$J(\Theta) = \alpha_r J_r(\Theta) + \alpha_b J_b(\Theta) + \alpha_{ic} J_{ic}(\Theta), \tag{10}$$

where

$$J_r(\Theta) = \frac{1}{N_r}\sum_{i=1}^{N_r}\left(\hat{f}_R(t_i^r,x_i^r)\right)^2 + \left(\hat{f}_C(t_i^r,x_i^r)\right)^2, \tag{11a}$$

$$J_{ic}(\Theta) = \frac{1}{N_{ic}}\sum_{i=1}^{N_{ic}}\left(\Psi(0,x_i^{ic}) - \hat{\Psi}(0,x_i^{ic})\right)^2 + \left(c(0,x_i^{ic}) - \hat{c}(0,x_i^{ic})\right)^2. \tag{11b}$$

Here, J_r, J_b, and J_{ic} represent the residual, boundary, and initial condition losses, respectively. Further, N_r and N_{ic} denote the total number of residuals and initial collocation points. The term J_b accounts for the disparities between the predicted value of the FNN and the true boundary value. Its formulation depends on the specific type of boundary condition utilized.

In both the 1D and 2D numerical tests, we utilized a total of 10,000 residual collocation points and 2,000 initial/boundary collocation points. The FNN architecture consists of 5 hidden layers, each containing 50 neurons. The output function $-\exp(x)$ is applied to the neuron corresponding to the predicted pressure head to enforce its negativity, while

the identity function is employed for the neuron representing solute concentration. The activation function employed in the FNN is the hyperbolic tangent with an adaptive activation technique [4], with the learning parameter initialized to 0.05 and the scale factor set to 20. The weights of the FNN parameters were initialized using the Xavier random initialization method [10], while the bias parameters b were set to zero initially. A two-step training technique was implemented, employing two optimizers sequentially, starting with Adam [7] and followed by L-BFGS-B [9]. We employed the Adam optimizer with exponential decay of the learning rate for optimization. The initial learning rate was configured to 10^{-3}, with a decay rate of 0.9, and the decay step was set to 1000. Additionally, the maximum number of iterations was limited to 20000. Other parameters were left at their default values. The parameters of the L-BFGS-B optimizer were set as follows: ftol = 10^{-10}, gtol = 10^{-8}, maxcor = 50, maxls = 50, maxiter = 50,000, and maxfun = 50,000, while the remaining parameters were kept at their default values. To improve training, a mini-batching technique was applied for the Adam optimizer, whereas a full-batch approach was employed for the L-BFGS-B optimizer. The implementation of the PINN solver was performed using the TensorFlow library [8]. Figure 1 presents a conceptual illustration of the PINN solver for modeling water flow and solute transport in unsaturated soils.

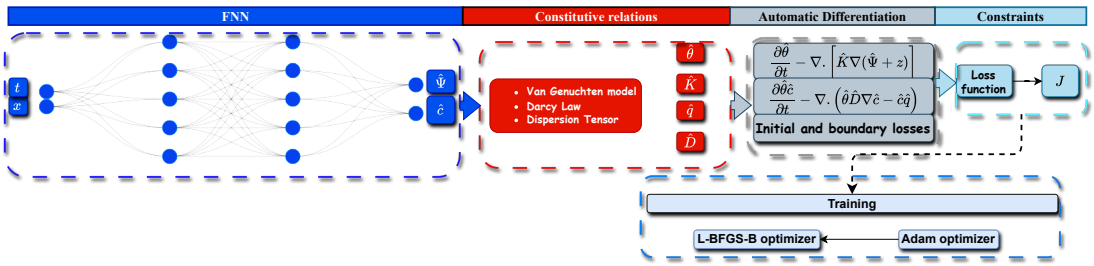

Figure 1: Diagram illustrating the PINN solver for the coupled water flow and solute transport model.

4 NUMERICAL EXPERIMENTS

In this section, we present two numerical experiments to compare the performance of PINN and FEM for solving the coupled model of water flow and solute transport. The first experiment involves a one-dimensional soil column test, while the second numerical test is conducted on a 2D soil with different boundary conditions.

For both numerical experiments, we used the following loamy soil parameters: $K_s = 0.2496\,\text{m/day}$, $\alpha_v = 3.6\,\text{m}^{-1}$, $\theta_s = 0.43$, $\theta_r = 0.078$, $n_v = 1.56$. For the one-dimensional

experiment, the initial and boundary conditions are outlined as follows: For $(t, z) \in (0, 1] \times [-1, 0]$, $\Psi(0, z) = -1\,\mathrm{m}$, $\mathbf{q}(t, 0) = -0.05\,\mathrm{m/day}$, $\dfrac{\partial \Psi}{\partial z}(t, -1) = 0$, $c(0, z) = 0\,\mathrm{mmol/m^3}$, $\bar{c}_0 = 50\,\mathrm{mmol/m^3}$, $\dfrac{\partial c}{\partial z}(t, -1) = 0$, where $\bar{c}_0$ indicates the concentration of the applied solute at the soil surface. The following solute parameters were used: $D_L = 0.1$ m, $D_w = 0.0$ m^2/day.

For the two-dimensional experiment, the initial and boundary conditions are outlined as follows: For $(t, x, z) \in (0, 1] \times [-1, 0] \times [-1, 1]$, $\Psi(0, x, z) = -1.3\,\mathrm{m}$, $\Psi(t, x, 0) = -0.2\,\mathrm{m}$, $\dfrac{\partial \Psi}{\partial z}(t, x, 0) = 0$, $c(0, x, z) = 0.1\,\mathrm{g/m^3}$, $c(t, x, 0) = 1\,\mathrm{g/m^3}$, $\dfrac{\partial c}{\partial z}(t, x, -1) = 0$. Additionally, a no-flux boundary condition was enforced along the lateral sides of the domain. The following solute parameters were used: $D_L = 0.5$ m, $D_T = 0.1$ m, $D_w = 0.0$ m^2/day.

To provide a benchmark for comparison with the PINN solution described in 3.2, and FEM with the settings outlined in 3.1, we computed a reference solution using FEM. For the 1D numerical test, the reference solution was obtained with $\Delta t_{\text{ref}} = 10^{-5}$ day and $\Delta z_{\text{ref}} = 10^{-4}$ m. In the 2D numerical test, the reference time step used was $\Delta t_{\text{ref}} = 10^{-4}$ day, and the computational domain was discretized into a 100×100 grid.

The obtained results are depicted in Figure 2, and a summary of accuracy and computational cost is illustrated in Table 1. Both FEM and PINN demonstrate good accuracy for pressure head and solute concentrations, with errors lower than 1% for both numerical tests. However, PINN exhibits significantly faster prediction times compared to FEM. Therefore, combining these two techniques could show promise for advancing subsurface modeling, particularly in terms of fast and accurate predictive capabilities.

Table 1: The obtained results for both PINN and FEM in solving the one- and two-dimensional infiltration-solute model.

1D numerical test					
	CPU of training (seconds)	CPU of prediction (seconds)	relative L^2-error for Ψ	relative L^2-error for c	Total loss
FEM	-	5.29	1.9e-3	2.2e-3	-
PINN	619.45	0.544	2.19e-3	3.42e-3	6.05e-3
2D numerical test					
FEM	-	58.75	1.5e-3	1.09e-3	-
PINN	1572.6	0.55	5.9e-3	3.06e-3	1.86e-05

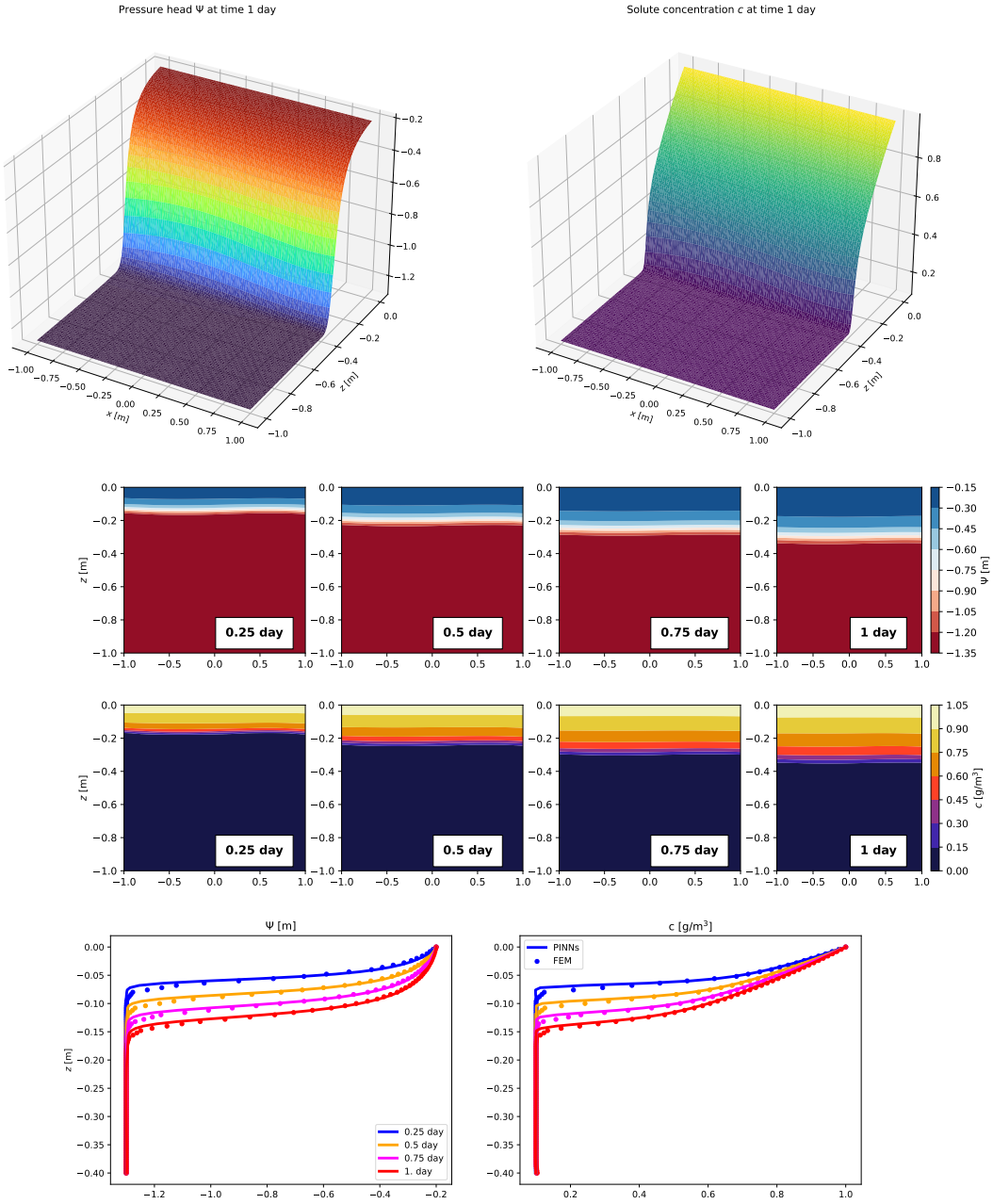

Figure 2: 2D numerical test- Snapshots of the obtained results from PINN and cross-sections at $x = 0$ of the pressure head and solute concentration obtained using PINN and FEM solvers.

5 CONCLUSION

This study highlights the accuracy and effectiveness of utilizing a vanilla physics-informed neural network (PINN) approach to solve the coupled model of water flow and solute transport. It compares the efficacy of this approach with the Galerkin finite element method (FEM). PINN is enhanced with techniques such as adaptive activation functions to better address the high nonlinearity inherent in the Richards equation. The results indicate that PINN can attain accuracy levels similar to FEM, despite requiring notably more computational resources during training. However, PINN maintains quick inference times, suggesting its potential as a viable alternative for modeling water-solute systems, albeit with a trade-off in computational cost.

Acknowledgments

This research was supported by UM6P/OCP Group of Morocco, the APRD research program from the Moroccan Ministry of Higher Education, Scientific Research and Innovation and the OCP Foundation, and NSERC-Canada.

REFERENCES

[1] Dastorani, M. T., Heshmati, M., & Sadeghzadeh, M. A. (2010). Evaluation of the efficiency of surface and subsurface irrigation in dryland environments. *Irrigation and Drainage: The journal of the International Commission on Irrigation and Drainage*, **59**(2), 129-137.

[2] Depina, I., Jain, S., Mar Valsson, S., & Gotovac, H. (2022). Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, **16**(1), 21-36.

[3] Bandai, T., & Ghezzehei, T. A. (2021). Physics-informed neural networks with monotonicity constraints for Richardson-Richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resources Research*, **57**(2), e2020WR027642.

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

[5] Van Genuchten, M. Th. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal*, **44**(5), 892-898.

[6] Hecht, Frédéric. (2012). New development in FreeFem++. *Journal of numerical mathematics*, **20**(3-4), 251-266.

[7] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[8] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 265-283.

[9] Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, **16**(5), 1190-1208.

[10] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249-256.

[11] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, **18**(153), 1-43.

[12] Davis, T. A. (2004). Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, **30**(2), 196-199.

[13] Celia, M. A., Bouloutas, E. T., & Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water resources research*, **26**(7), 1483-1496.

[14] Grossmann, T. G., Komorowska, U. J., Latz, J., & Schönlieb, C. B. (2023). Can physics-informed neural networks beat the finite element method?. *arXiv preprint arXiv:2302.04107*.

[15] Haruzi, P., & Moreno, Z. (2023). Modeling Water Flow and Solute Transport in Unsaturated Soils Using Physics-Informed Neural Networks Trained With Geoelectrical Data. *Water Resources Research*, **59**(6), e2023WR034538.

[16] Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *physics*, **1**(5), 318-333.

[17] Bear, J. (2013). Dynamics of fluids in porous media. *Courier Corporation*.

[18] Keita, S., Beljadid, A., & Bourgault, Y. (2021). Implicit and semi-implicit second-order time stepping methods for the Richards equation. *Advances in Water Resources*, **148**, 103841.

[19] Kamil, H., Beljadid, A., Soulaimani, A., & Bourgault, Y. (2024). Semi-implicit schemes for modeling water flow and solute transport in unsaturated Soils. *Advances in Water Resources*, Under Review.

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

[21] Roy, A. M., Bose, R., Sundararaghavan, V., & Arróyave, R. (2023). Deep learning-accelerated computational framework based on Physics Informed Neural Network for the solution of linear elasticity. *Neural Networks*, **162**, 472-489.

[22] Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2021). hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, **374**, 113547.

[23] Toutlini, N., Beljadid, A., & Soulaïmani, A. (2024). Analysis of second-order temporal schemes for modeling flow-solute transport in unsaturated porous media. *arXiv preprint arXiv:2404.03603*.