Predicting Soil Moisture Dynamics with Physics-Informed Neural Networks

Samuel Ghalayini
samuel.ghalayini@colorado.edu
University of Colorado Boulder
Boulder, Colorado, United States of America

Abstract

Sustainability and economic viability are pillars of modern farming. Irrigation systems play a foundational role in providing those pillars. Water management can help farmers reduce the environmental impact of runoff, save costs on wasted water, and can provide higher crop yields. Predicting the soil moisture profile accurately allows farmers to make better informed decisions to save water and produce more crop yield for the season. Current soil moisture measurements are taken at discrete depths between 0.05-1m. This process does not inform the farmer of subtle hydrodynamics in the soil. Nor does it give information on future soil moisture states. Farmers deal with varying types of soil and crop types that have varying root depths.

I propose a Physics-Informed Neural Network (PINN) model to predict the 1D soil moisture profile, initialized with real world observations, so farmers can predict when to irrigate. This model is constrained by the Richards Equation. The Richards Equation is a nonlinear stiff partial differential equation that is used to solve for hydrodynamics through unsaturated porous materials like soil. With this model farmers will be able to predict the moisture needs of their crops from live sensor data and weather forecasts. This project produces a PINN model that predicts long term with transfer learning. The PINN approach achieved a RMSE of 0.0056 m^3/m^3against the implicit Euler solver, a RMSE against real world observations at 0.018 m^3/m^3, and respected the governing physics equation with residuals 1×10^{-7} below our target 1×10^{-2}. Most notably, the model maintained computational efficiency for longer term predictions where the traditional numerical solvers could not efficiently compute.

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Neural networks**; • **Applied computing** → *Agriculture.*

Keywords

Physics-Informed Neural Networks, Richards Equation, Soil Moisture, Transfer Learning, Irrigation Management, Deep Learning

1 Introduction

In recent years we have seen fresh water become a scarce and expensive resource. Agriculture irrigation takes up 70% of the global freshwater supply [20]. The high demand for freshwater is also strained by climate change, pollution, and demand for water from other sectors. Poor irrigation management leads to unnecessary costs including higher water and energy bills, fertilizer and nutrient loss from over saturation and runoff, and crop production impacts. Currently this issue is being solved with data-driven automated irrigation systems, remote sensing, and precision drip irrigation infrastructure. These solutions address parts of irrigation issues but do not model the continuous soil moisture profile. Without a continuous profile it is difficult to tell how water is moving through the soil, the health of our soil, and crop specific irrigation needs. With a continuous prediction we can enable a robust foundation for a digital twin model. This foundation will be able to simulate complex hydrodynamics which lead to more crop growth with less water waste.

,
2025.

1.1 Problem Statement

The main two challenges with current methods are sparse field measurements and computational cost of solving the Richards Equation. Sparse measurements only tell us the soil moisture at a specific depth. Sparse measurements can be interpolated to create a 1D continuous profile. Different interpolation methods such as linear, constant, or spline make different assumptions about how water is moving through soil that is not consistent with the actual physical equations that predict flow through soil. The Richards Equation is used to model water flow through unsaturated soil but traditional numerical methods (implicit/explicit) are computationally expensive. The following PINN is constrained with the Richards Equation to learn water flow dynamics in soil. This model would be computationally less expensive for longer predictions and provide a continuous 1D profile to accurately model crop specific irrigation needs.

1.2 Contributions

There have been PINN solutions that use the Richards Equation in related work [10] but this will be the first project that applies ISMN station data to a model as initial conditions. Transfer learning is also an applicable method that hasn't been applied to Richards Equation PINN's. Transfer learning will allow predictions for longer timelines and leads to higher rates of convergence for this unstable PDE. This project models complex hydrodynamics and give decision support that is easily digestible to farmers. Different crops require different methods such as stressing the crops or drip irrigation. This model gives farmers more accurate information to implement their irrigation methods.

2 Related Work

2.1 Traditional Soil Moisture Prediction

Traditional soil moisture predictions take on a variety of approaches. Remote sensing predicts soil moisture without sensors in the ground. Two leading methods in remote sensing are satellite imagery and drone imagery. Satellite data can assess large (100km-9km) areas of surface moisture with CNN's and Random Forest models [6]. Satellite sensing aids in understanding Earth's water content but still compares accuracy with in-situ (on-site) measurements. Zhang et al. [18] have used drones to fly across farmland and take a combination of imaging techniques to detect soil moisture. Both of these methods can predict soil moisture but are focused on longer time frames. Remote sensing is usually focused on seasonal and operational planning, the PINN model focuses on tactical irrigation scheduling and when to irrigate next. Other in-situ measurements are sparse and focused on providing current soil moisture levels instead of the full dynamics of water movement through the entire soil profile over time. There are two main software packages that model hydrodynamics in soil, namely HYDRUS-1D and SWAP [21]. Both software packages use finite element methods for solving the Richards equation.

2.2 Physics-Informed Neural Networks

Raissi et al. [13] introduced Physics-Informed Neural Networks (PINN's), which constrain a neural network by incorporating physical laws into the loss function. This combines a data-driven approach with a physics-based approach. The network is guided by the loss function towards a solution that respects the physics. PINNs are commonly applied to model complex systems such as fluid dynamics, structural mechanics, heat transfer, electromagnetism, geophysics, and finance. The total loss is comprised of data loss + physics loss. This way the network evaluates how well the physics loss satisfies the governing PDE, forcing the network to learn solutions that respect both data and the underlying PDE. The Richards equation is particularly challenging because it is nonlinear, stiff, and has low convergence rates. Li et al. [12] fit a PINN with the Richards Equation to model hydrodynamics with synthetic data.

3 Proposed Work

3.1 Problem Formulation

In this project I will be using station data to calibrate the model in its prediction. The PINN is governed mainly by the Richards Equation(1). To understand this equation we will start with three main components: water flows from high to low hydraulic potential, the conservation of mass(2), and Darcy's law(3). The Richards Equation LHS defines the change in soil water content over time as a function of water flux through the soil. The RHS consists of substituting Darcy's law for the conservation of mass equations. Darcy's law RHS consists of hydraulic conductivity $K(h)$, the pressure gradient $\partial h/\partial z$, and a term accounting for gravity "+1".

$$\frac{\partial \theta}{\partial t} = \nabla \cdot [K(h)(\nabla h + \nabla z)] \tag{1}$$

$$\frac{\partial \theta}{\partial t} = \nabla \cdot q \tag{2}$$

$$q = -K(h)\left[\frac{\partial h}{\partial z} + 1\right] \tag{3}$$

With this mixed form(1) I propose solving this PINN first with the intuitive moisture form then also with the more stable pressure head form. Computing the 1D form we apply the chain rule and substitute soil water diffusivity to get our Richards Equation in moisture form.

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z} + 1\right)\right] \tag{4}$$

$$D(\theta) = K(\theta) \cdot \frac{dh}{d\theta} \tag{5}$$

Moisture based form:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[D(\theta)\frac{\partial \theta}{\partial z}\right] + \frac{\partial K(\theta)}{\partial z} \tag{6}$$

To convert the mixed form to the pressure form(9) is much less intensive and takes one chain rule step of taking out the specific moisture capacity $C(h)$ from the LHS.

$$\frac{\partial \theta}{\partial t} = \frac{d\theta}{dh} \cdot \frac{\partial h}{\partial t} \tag{7}$$

$$C(h) = \frac{d\theta}{dh} \tag{8}$$

Pressure based form:

$$C(h)\frac{\partial h}{\partial t} = \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z} + 1\right)\right] \tag{9}$$

Now that we have the Richards Equation in different forms, I first solve them with forward and backward Euler methods. I chose these methods to compare the PINN against because they are step-wise, easier to tune, show the nonlinearity of the PDE, and also show the challenges of slow convergence and small time steps. I fit one one PINN model to the moisture form and one to the pressure head form. I found that the pressure head form was bounded by larger ranges, versus the moisture form, this led to smoother gradients. For both the Euler methods and PINN approach, simulations solved the Richards equation over a 1-meter soil depth. The first development simulation was a simplified case with dry initial conditions and constant surface moisture change. Then I ran an irrigation/precipitation simulation with a wetting front and free drainage lower boundary.

Numerical Solution Methods:

Forward Euler:

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \frac{\partial}{\partial z}\left[K(h^n)\left(\frac{\partial h^n}{\partial z} + 1\right)\right] \tag{10}$$

Backward Euler:

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \frac{\partial}{\partial z}\left[K(h^{n+1})\left(\frac{\partial h^{n+1}}{\partial z} + 1\right)\right] \tag{11}$$

3.2 Datasets

For this project I will be accessing 57 datasets across 8 global soil moisture networks. I split the stations on a 70/16/14 training/validation/testing split.Each dataset has been pulled from the International Soil Moisture Network [7]. The geographical requirements for each station was that it was in a temperate climate and had sensors placed in crop land, rather than wet lands, grasslands, or urban areas. The sensor data requirements were that each station had soil moisture, soil temperature, air temperature, and precipitation. The data is a time series data structured by the hour. Each station has varying amounts of soil moisture sensors at different depths. Besides soil moisture datasets I gathered soil specific parameters to solve the Richards Equation such as described in Table 1 and Table 2.

Table 1: Soil specific parameter descriptions

Parameter	Description
θ_s	Saturated water content [dimensionless], maximum water content held by soil
θ_r	Residual water content [dimensionless], minimum water content held by soil
K_{sat}	Saturated hydraulic conductivity [L/T, m/s], Maximum rate water can flow through saturated soil
m, n, α	van Genuchten parameter [dimensionless], used to control the slope of the retention curve

3.3 Data

To ensure a high data quality I preprocessed the data by first interpolating the data linearly among any missing values. Then to clean the dataset I set realistic bounds for each variable. Soil moisture had a range of 0-0.6 $m^3/m^3$and precipitation had a range of 0-200 mm/hr. Soil and air temperature were bounded at -10-60C and -40-100C respectively. Data was sorted by timestamp and duplicate values were removed. For further exploratory data analysis, averages and gradients were calculated, see Figure 11 in Appendix A. By plotting the average I could tell the seasonal trends of soil moisture. By plotting the gradients I could capture the rate of moisture change through the soil profile among the different sensor depths.

Figure 1 shows a year long plot of 4 different soil moisture depths (0.05, 0.20, 0.5, and 1.0m). The viridis color pallette is used in the following figures; the lighter color represents a depth close to the surface while a darker color represents a deeper sensor. For this paper the Pessonada Station dataset was chosen which is located in the Catalonia region of Spain. Figure 1 also shows that the sensor closer to the ground is much more variable and dries/wets much faster than the deeper sensor. This is a representation of how soil specific parameters effect hydrodynamics. Two graphs that represent how soil moisture moves through different soils is shown in Figures 2a and 2b. The water retention graph represents how different soils hold onto water as the pressure head increases. The pressure head is a unit of representing how far a fluid's energy would travel in distance. Sand's rapid drop in moisture reflects its large pores and inability to hold onto water compared to silt or clay which hold onto its water at higher pressures. The second figure to help understand soil specific hydrodynamics is hydraulic conductivity. Hydraulic conductivity represents how easily water moves through porous material. Figure 2b shows why sandy soil needs frequent irrigations or else the ability for water to move rapidly declines. Whereas the other three soil types can supply water for a plant for longer and requires less frequent irrigation events(rain or watering).

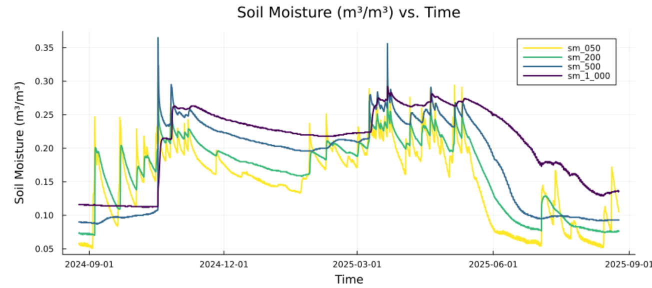
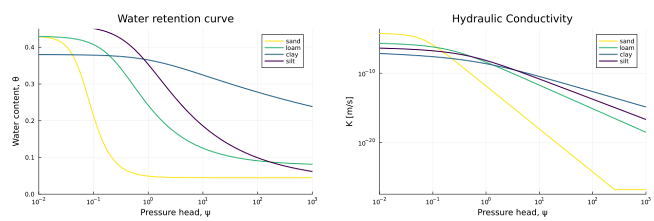


Figure 1: Soil moisture measurements across four sensor depths at the Pessonada station.



(a) **(b)**

Figure 2: Soil hydraulic properties for different textures.

Table 2: Van Genuchten-Mualem soil hydraulic parameters for different soil textures used in model simulations [3].

Soil Type	θ_r	θ_s	α	n	m	K_{sat}
Sand	0.045	0.43	14.5	2.68	0.627	5.8×10^{-5}
Loam	0.078	0.43	3.6	1.56	0.359	2.9×10^{-6}
Clay	0.068	0.38	0.8	1.09	0.083	5.6×10^{-7}
Silt	0.034	0.46	1.6	1.37	0.270	6.9×10^{-7}

3.4 PINN Architecture

For the PINN architecture I constructed a profile based system. Where the network size, batch size, and max iterations vary between the profiles minimal, development, and production. My input variables are depth and time. The output variables will be pressure head ψ, or moisture content θ. The network architecture is in Table 3 which shows the number of hidden layers, how many densely connected nodes there are, and the maximum number of iterations the model trained for. Due to the low convergence rates I also implemented an early stopping criterion to speed up development and prevent overfitting.

For this project I used the tanh activation function per Jagtap et al. [16]. Unlike the sigmoid function, the tanh activation is zero centered and the steeper gradients help with the vanishing gradient problem and help convergence rates. This PDE is nonlinear making it hard to find a convergence with specific soil parameters with boundary conditions. In creating the models I first developed the soil moisture form which had trouble converging and then I developed the PINN to train with the pressure head form of the PDE.

Table 3: Profile specific network architectures and training parameters

Profile Name	Network Architecture	Batch Size	Max Iters
Minimal	(2, 16, 16, 1)	500	100
Development	(2, 32, 32, 32, 1)	1500	250
Production	(2, 64, 64, 32, 1)	3000	1000

The prediction window I am concerned with is from 24-72hr. This amount of time is short enough to have a reliable PINN prediction and long enough to give decision support on irrigation. Due to the small time-steps of numerical methods and PINN's, I developed a transfer learning model where we can feed the parameters for a 1hr converged solution to the training for 24hr and 72hr predictions. The network profiles enable transfer learning size compatibility.

3.5 Loss Function Formulation

The key difference between this data-driven approach and an LSTM or NN approach is the formulation of the loss function. For a PINN the loss function consists of weighting the data loss, PDE loss, boundary condition loss, and initial condition loss(13). The three different losses come from the sensor locations, the error enforced by the Richards Equation, and error enforced by the initial moisture profile and boundary conditions that reflect precipitation or drying events. The package I will be using, NeuralPDE, allows for each loss to be set to

$$(\lambda_{PDE} = 4.0, \lambda_{IC}, \lambda_{BC} = 1.0) \tag{12}$$

Through testing I've found weighing the PDE loss four times as much allow for a higher convergence because the PINN values learning the Richards Equation. Additionally, rather than using sensor data in training, the solution is constrained with sensor data by its initial and boundary conditions.

$$L_{total} = \lambda_{data} \cdot L_{data} + \lambda_{PDE} \cdot L_{PDE} + \lambda_{BC} \cdot L_{BC} + \lambda_{IC} \cdot L_{IC} \tag{13}$$

3.6 Transfer Learning Methodology

To enable longer prediction times and high convergence rates on a non-uniform solution. I leveraged transfer learning. Shorter time frames provide optimized initial parameters for extended simulations. The main idea is for a short time frame PINN model to "warm-start" the longer networks with parameters that already respect the Richards equation and boundary conditions. I found a significant decrease in the number of iterations until loss plateaus, decrease in training time, and an improvement on final loss.

I save my network parameters in a binary BSON file. Then on my second training iteration I load the saved BSON file into the initial parameter weights for the longer model. I do to not freeze any parameters but allow each one to be fine tuned to fit to the longer time domain. A major constraint in transfer learning is the network compatibility. Each profile has a different network size and therefore can only transfer parameters among the same profiles.

3.7 Implementation Details

For implementing this project I used the programming language Julia. Julia is known to be a scientific programming language which benefits from speed and ease of use. Its syntax is closer to MATLAB with speed close to C++. It also has an active community of developers working on physics-informed ML packages such as NeuralPDE. I will be running this on my local computer with different profiles. Utilizing transfer learning, the longer prediction times of 100-1000hr will take longer but are significantly more likely to converge than a randomized setup of network parameters. For reproducibility I will include the project .toml files and my hardware specs in the project github.

Code Repository: All source code is available at https://github.com/sam-ghala/soil_moisture.

3.8 Tasks

This project consisted of eight tasks that ranged from EDA, model development, numerical solvers, transfer learning, evaluation, and documentation. I've organized the work into the following tasks:

Task 1: Create a data pipeline that handles different station network data types and different soil moisture depths.

Task 2: PDE formulation and problem setup for different drying and wetting fronts.

Task 3: PINN architecture design to handle different profiles for testing and production.

Task 4: Traditional solver implementation for implicit and explicit Euler method.

Task 5: Model training and validation to check that the solution is a non-trivial solution that also respects the physics.

Task 6: Transfer learning implementation to extend prediction times and complex wetting fronts.

Task 7: Create data visualizations for datasets, soil moisture retention, hydraulic conductivity, and solutions.

Task 8: Evaluation and documentation of project analyzing computational efficiency, accuracy to historical station data, physical compliance, and comparison to numerical solvers.

4 Evaluation

This project is evaluated on four tiers across numerical accuracy, physical compliance, real-world validation, and computational efficiency.

4.1 Numerical Baseline Comparison

To validate the PINN approach I compared three models. An implicit Euler, explicit Euler, and a PINN model. I chose to compare accuracy against the implicit model because it has fine temporal steps and is generally more stable than the other two. I chose to predict moisture for 1hr because longer than that led to convergence/instability issues. Each model had the same setup, each ran on the same initial conditions, had 50 nodes over the 1m depth, were measured 10 times over the 1hr run and had the same soil specific parameters(loamy soil) ($\theta_r = 0.078$, $\theta_s = 0.43$, $\alpha = 3.6$ m^{-1}, $n = 1.56$, $m = 0.359$, $K_{sat} = 2.9 \times 10^{-6}$ m/s). The initial condition was taken from sensor observations and interpolated to create a 1D gradient. The temporal discretizations will be in the range of 0.001s for the explicit method and 0.01s for the implicit method.

Table 4 shows the RMSE comparison between the implicit time steps simulation. The PINN approach achieved an RMSE of 0.005578 across the 1hr simulation, surpassing our target of < 0.03. The explicit method showed better error values of 0.000215, part of finetuning this evaluation setup comes down to soil specific parameters.

Table 4: RMSE Numerical method comparison for 1hr drying simulation across four sensor depths (0.05, 0.20, 0.50, 1m)

Time (hr)	Explicit RMSE (m^3/m^3)	PINN RMSE (m^3/m^3)
0.0	0.000000	0.001212
0.1	0.001724	0.006443
0.2	0.000403	0.007202
0.3	0.000102	0.007072
0.4	0.000034	0.006738
0.5	0.000020	0.006362
0.6	0.000016	0.005983
0.7	0.000016	0.005611
0.8	0.000016	0.005251
0.9	0.000016	0.004904
1.0	0.000016	0.004576
Mean	**0.000215**	**0.005578**

Key findings from this evaluation were that the PIN successfully modeled the Richards Equation below a 0.03 RMSE for a drying event. The figure below also plots the implicit, explicit, and PINN predictions. We can see a gradual decline in soil moisture reflecting a drying simulation. The traditional numerical solvers follow a smoother curve than the PINN. The difference between these can be attributed to the fact that the Euler methods calculate the solution at each timestep while the PINN learns a continuous function throughout the entire time domain. This is a fundamental difference in these approaches but the key result for this evaluation is that the PINN model modeled the physics within an acceptable error range.

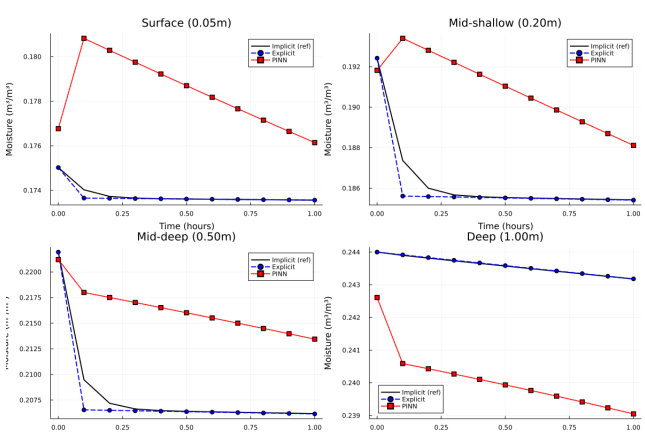


Figure 3: Comparison of numerical methods for 1-hour drying simulation at four depths.

The second test scenario is a more complex wetting front where the top boundary condition starts at $\theta = 0.1780$, then rains for 1mm/hr for 30min then dries for the remaining time. This evaluation was much more difficult to get all three methods to converge on a non-trivial solution. But I repeatedly saw the PINN just break the RMSE target of 0.03. In the following figure we can see that the traditional solvers accurately show the fast increase in moisture at the surface level and gradual drying over the second half of the simulation. We can also see an accurate representation of water taking more time to reach lower depths. The PINN model in this approach still jumps. The parameters being tuned for the PINN model were the boundary conditions which was outside the scope/timeline of this report and is discussed in the discussion/future work section below. The key findings we can see is that for a wetting front, the PINN model approaches our target error.

Table 5: Numerical method comparison for 1-hour wetting/drying simulation: RMSE against implicit reference across four sensor depths (0.05, 0.20, 0.50, 1.00m). Rainfall applied for first 30 minutes at 3.6 mm/hr.

Time (hr)	Explicit RMSE (m^3/m^3)	PINN RMSE (m^3/m^3)
0.0	0.000000	0.001937
0.1	0.001650	0.022949
0.2	0.003708	0.009701
0.3	0.004177	0.010973
0.4	0.004276	0.025814
0.5	0.004293	0.041472
0.6	0.000978	0.047929
0.7	0.000183	0.051783
0.8	0.000058	0.054912
0.9	0.000077	0.057658
1.0	0.000083	0.060103
Mean	**0.001771**	**0.035020**

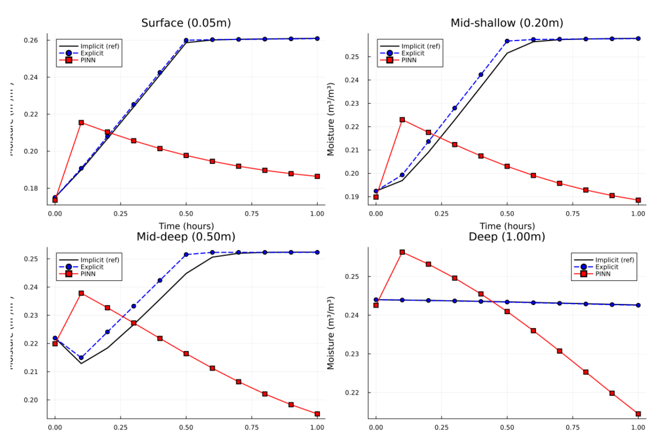


Figure 4: Comparison of numerical methods for 1-hour wet simulation at four depths.

4.2 Physical Compliance

After training I evaluated equation (14) at N locations to test the Richards Equation residual. The target residual is $< 1 \times 10^{-2}$ which indicates a solution that adhered to the Richards equation.

$$\varepsilon_{PDE} = \frac{1}{N} \sum \left| C(\psi) \frac{\partial \psi}{\partial t} - \frac{\partial}{\partial z} \left[K(\psi) \left(\frac{\partial \psi}{\partial z} + 1 \right) \right] \right| \tag{14}$$

During development I started with Quadrature training tolerances of 1×10^{-4} for faster prototyping then increased the tolerances to 1×10^{-5}. The tighter tolerances on the model forces more accurate loss gradients which calculates a more precise physics loss, which led to better PDE convergence.

The table below shows that the PINN successfully models the Richards Equation below the target residual. The Richards equation is a nonlinear stiff PDE which is unstable and can lead to low convergences for drying and especially more complex irrigation events. In training I added a weighted loss to the network which weighted the PDE loss 4 times greater than the boundary conditions and initial conditions loss. Modeling the hydrodynamics given just the initial condition of the soil demonstrates the PINN's ability to learn the physics for prediction. The key findings from this evaluation were that by weighing the PDE loss in the loss function formulation we could achieve mean residuals of 1×10^{-7} across "cold-start" networks and transfer learned networks. Another key insight from this evaluation is we can see the effect transfer learning has on our physical compliance. With a 72 times increase in the time domain we can still get lower mean and max residuals.

Table 6: PDE residuals for 1hr, 24hr, and 72hr predictions

Time Doamin	Initialization	Mean Residual	Max Residual
1 hour	Random	2.89×10^{-7}	3.97×10^{-6}
24 hours	1hr model	2.02×10^{-7}	5.65×10^{-7}
72 hours	24hr model	3.01×10^{-7}	5.52×10^{-7}

4.3 Real World Validation

The third type of evaluation is to validate the model's prediction to real station data. To validate against real world observations I initialized the PDE initial conditions with sensor data that was interpolated for 50 data points instead of the sparse sensors. After setting the initial condition I chose a date with no precipitation events in the following 72hr. This allowed the model to validate itself against a real world "drying soil" example. In the following figures we can see the heatmap of the 1D soil moisture profile dry over 24hr and 72hr. The heatmap shows us that the PINN is respecting the Richards Equation and modeling the soil dynamics. The left figure in the heatmaps show us the soil moisture profile prediction in terms of soil moisture and the right graph shows us the same prediction in terms of pressure head.

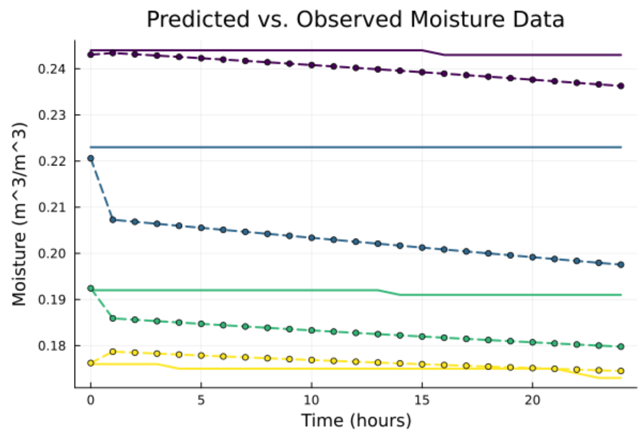


Figure 7: Comparison of PINN predictions (lines) with sensor observations (points) at four depths over 24 hours.

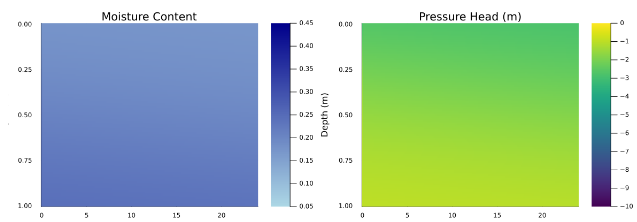


Figure 5: PINN prediction of soil moisture profile over 24 hours during a drying period. Left: moisture content; Right: pressure head.

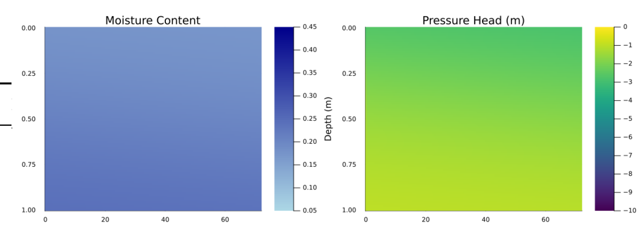


Figure 6: PINN prediction of soil moisture profile over 72 hours during a drying period. Left: moisture content; Right: pressure head.

To validate these results I've plotted the PINN predictions alongside the sensor predictions for the discrete depths. We can see in the top and bottom sensors the model predicts normal drying conditions but in the middle the soil moisture varies from observation. This can be attributed to soil specific parameters that are not available per site.

In Figure 7 and Figure 8 we can see large jumps after the initial profile is setup. This can be attributed to unknown soil parameters unique to Pessonada and not known from data. Table 7 shows the error of our model to observations. The model performed well on all depths (0.019 m/m (24hr) and 0.015m/m (72hr)). I knew the model respected the physics from the Physical compliance and

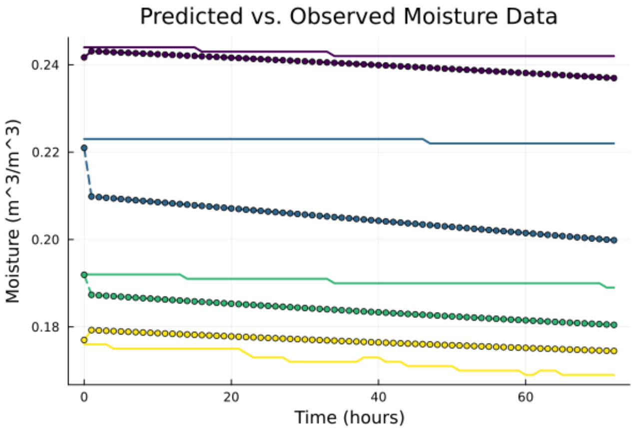


Figure 8: Comparison of PINN predictions (lines) with sensor observations (points) at four depths over 72 hours.

numerical baseline evaluation in 4.1 and 4.2. But this evaluation showed that the model can predict accurately to real world data which is contributing to the area of PINN's for soil moisture.

Table 7: PINN validation against sensor observations: RMSE and MAE for 24-hour and 72-hour drying predictions

	24-Hour		72-Hour	
Depth (m)	**RMSE (m^3/m^3)**	**MAE (m^3/m^3)**	**RMSE (m^3/m^3)**	**MAE (m^3/m^3)**
0.05	0.031505	0.028863	0.027245	0.026832
0.20	0.022421	0.018192	0.015979	0.014230
0.50	0.012458	0.010709	0.010525	0.009112
1.00	0.009154	0.007326	0.007153	0.006365
Mean	**0.018885**	**0.016273**	**0.015226**	**0.014135**

4.4 Computational Efficiency

The last form of evaluation on this model is to compare computational expenses between the PINN model and traditional numerical methods. The metrics I propose to measure are in Table 8.

First addressing the compute metrics for different solvers. The explicit Euler solver performed the best across the board. I believe the numerical stability of the traditional solvers is hard to beat with the stochastic nature of the PINN parameter initialization. Some of the most surprising findings from this project's evaluations were in the usefulness of transfer learning. After testing that the PINN adhered to the governing physics equation. The next task was to predict for longer time frames. The scalability of the traditional solvers struggled. A 24hr simulation could not finish its step-wise computation in a 30-min cutoff time. The training time for transfer learning PINN models for 24hr and 72hr took 27.72s and 26.55s respectively. These values are even lower than the 1hr PINN solve. These "warm-start" networks had the advantage of starting to train with fine tuned parameters already aimed at solving the same equation but for a longer time frame.

Table 8: Computational efficiency metrics

Metric	Description	Target
Training Time	Clock time for convergence	< 3 min in dev profile
Inference Time	Clock time to predict 1-hour profile	< 1 sec
Memory Footprint	Model size (parameters)	Min: ~1KB, Dev: ~5KB, Prod: ~15KB
Convergence Rate	Iterations to loss plateau	50% fewer with transfer

Table 9: Performance comparison for 1hr simulation.

Method	Train Time	Inference	Memory
Implicit	99.67 s	0.232 ms	0.39 KB
Explicit	23.92 s	0.010 ms	0.39 KB
PINN	56.76 s	43.357 ms	17.67 KB

Table 10: PINN performance using transfer learning for 24 h and 72 h simulation runs.

Simulation Run	Train Time	Inference	Memory
24 hr Explicit	DNF	DNF	DNF
24 hr Implicit	DNF	DNF	DNF
24 hr PINN	27.72 s	21.197 ms	17.67 KB
72 hr PINN	26.55 s	24.474 ms	17.67 KB

5 Discussion

5.1 Timeline

I've scheduled eight tasks laid out over the six week course. Week 1 starts with this project proposal and data pipeline development and PDE formulation. Weeks 2-3 focus on PINN architecture design and traditional solver implementation. Weeks 3-4 focus on model training and validation. Week 4-5 implements transfer learning for extended predictions. For weeks 5-6 I developed the evaluations and documentation. Data visualizations is an ongoing development over weeks 1, 3, and 5.

5.2 Challenges

There were many unforeseen challenges in developing this soil moisture prediction model. Some to note are developing the physics for the traditional numerical solvers, understanding how to set the boundary conditions for the PDE, getting each model to converge for a nonlinear stiff PDE, and understanding the role soil specific parameters play in defining an accurate soil moisture model. Soil specific parameters was the biggest hurdle to understand because the immense role they play in how water moves. Fine tuning parameters to fit site specific in-situ measurements required understanding Carsel et al. [3]. Alternative approaches I've taken while developing the ML approach is finding a balance between a complex form of the full Richards equation and a simple diffusion equation with gravity.

5.3 Future Research

Future work for this project consists of refining the analysis methods here and expanding the simulation scope. My first suggestion would be to incorporate soil specific parameters classification. Knowing the soil parameters help greatly in modeling the soil moisture and if they could be calculated per site then this would make the current soil moisture prediction site agonistic and therefore soil agonistic.

From a robust soil moisture prediction model I believe the next steps for this is to incorporate live calibration of the model using sensor data. If an online model could take in sensor data and update itself continuously by reading where its predictions have differed from the observations, this could enable a controls-like simulation to model crop yield. I envision this project to be the stepping stone for a full crop yield digital twin. After successfully modeling the soil moisture and soil specific parameters, the next steps would be to incorporate soil nutrients and a plant sink/source to model water and nutrients being removed from the soil.

6 Conclusion

Over the course of this project I have constructed a physics-informed neural network framework for solving the Richards equation that addresses the computational limitations of traditional numerical methods. This approach leverages transfer learning to achieve longer prediction time frames while maintaining computational efficiency. The resulting model enables irrigation management by providing farmers soil specific moisture hydrodynamics. By evaluating the PINN against Euler's methods the PINN model achieved a RMSE of $0.0056\text{m}^3/\text{m}^3$ against the implicit solver. The PINN evaluated against real world observations has achieved a RMSE below the target at $0.018\text{m}^3/\text{m}^3$. By weighing the PDE loss of for the loss function our model also respects the governing Richards Equation at levels below the target residual at 1×10^{-7}. The most beneficial approach came from applying transfer learning to achieve a PINN Richards Equation solver for long term predictions that failed to finish with Euler's methods (with a 30-min cutoff).

References

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

[2] Joaquim Bellvert, Ana Pelechá, Magí Pamies-Sans, Jordi Virgili, Mireia Torres, and Jaume Casadesús. 2023. Assimilation of Sentinel-2 Biophysical Variables into a Digital Twin for the Automated Irrigation Scheduling of a Vineyard. *Water* 15, 14: 2506.

[3] Robert F. Carsel and Rudolph S. Parrish. 1988. Developing joint probability distributions of soil water retention characteristics. *Water Resources Research* 24, 5: 755–769.

[4] Yang Chen, Yongfu Xu, Lei Wang, and Tianyi Li. 2023. Modeling water flow in unsaturated soils through physics-informed neural network with principled loss function. *Computers and Geotechnics* 161: 105546.

[5] Rowan Cockett, Lindsey J. Heagy, and Eldad Haber. 2018. Efficient 3D inversions using the Richards equation. *Computers & Geosciences* 116: 91–102.

[6] Deepanshu Lakra, Shobhit Pipil, Prashant K. Srivastava, Suraj Kumar Singh, Manika Gupta, and Rajendra Prasad. 2025. Soil moisture retrieval over agricultural region through machine learning and Sentinel 1 observations. *Frontiers in Remote Sensing* 5: 1513620.

[7] Wouter Dorigo, Irene Himmelbauer, Daniel Aberer, et al. 2021. The International Soil Moisture Network: serving Earth system science for over a decade. *Hydrology and Earth System Sciences* 25, 11: 5749–5804.

[8] D. Dourado Neto, Q. De Jong Van Lier, M.Th. Van Genuchten, K. Reichardt, K. Metselaar, and D.R. Nielsen. 2011. Alternative Analytical Expressions for the General van Genuchten–Mualem and van Genuchten–Burdine Hydraulic Conductivity Models. *Vadose Zone Journal* 10, 2: 618–623.

[9] Matthew W. Farthing and Fred L. Ogden. 2017. Numerical Solution of Richards' Equation: A Review of Advances and Challenges. *Soil Science Society of America Journal* 81, 6: 1257–1269.

[10] Jiajing Guan, Sophia Bragdon, and Jay Clausen. 2024. *Predicting soil moisture content using Physics-Informed Neural Networks (PINNs)*. Engineer Research and Development Center (U.S.).

[11] Hans Henriksen, Raphael Schneider, Julian Koch, et al. 2022. A New Digital Twin for Climate Change Adaptation, Water Management, and Disaster Risk Reduction (HIP Digital Twin). *Water* 15, 1: 25.

[12] Yanling Li, Qianxing Sun, Yuliang Fu, and Junfang Wei. 2025. Solving the Richards infiltration equation by coupling physics-informed neural networks with Hydrus-1D. *Scientific Reports* 15, 1: 18649.

[13] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 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.

[14] Maziar Raissi, Paris Perdikaris, Nazanin Ahmadi, and George Em Karniadakis. 2024. Physics-Informed Neural Networks and Extensions. Retrieved from http://arxiv.org/abs/2408.16806.

[15] Ari Rappaport. A posteriori error estimates and adaptivity in numerical approximation of PDEs: regularization, linearization, discretization, and floating point precision.

[16] Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. 2020. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics* 404: 109136.

[17] Mercedeh Taheri, Mostafa Bigdeli, Hanifeh Imanian, and Abdolmajid Mohammadian. 2025. An Overview of Machine-Learning Methods for Soil Moisture Estimation. *Water* 17, 11: 1638.

[18] Y. Zhang, W. Han, H. Zhang, X. Niu, and G. Shao. 2023. Evaluating soil moisture content under maize coverage using UAV multimodal data by machine learning algorithms. *Journal of Hydrology* 616: 128825.

[19] Kirill Zubov, Zoe McCarthy, Yingbo Ma, et al. NeuralPDE: Automating Physics-Informed Neural Networks (PINNs) with Error Approximations.

[20] The World Bank. 2024. Agricultural Water Withdrawal (Freshwater Withdrawal). World Development Indicators. Retrieved October 13, 2025 from https://data.worldbank.org/indicator/er.h2o.fwag.zs

[21] PC-Progress. 2024. HYDRUS-1D for Windows (Version 4.xx). Retrieved October 13, 2025 from https://www.pc-progress.com/en/Default.aspx?H1D-description

A Appendix

This appendix stores additional context for understanding soil moisture dynamics from the Pessonada station dataset.

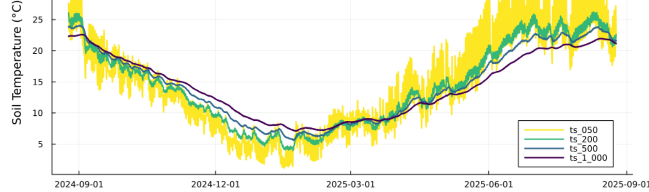


Figure 9: Soil temperature measurements across four sensor depths at the Pessonada station over one year.

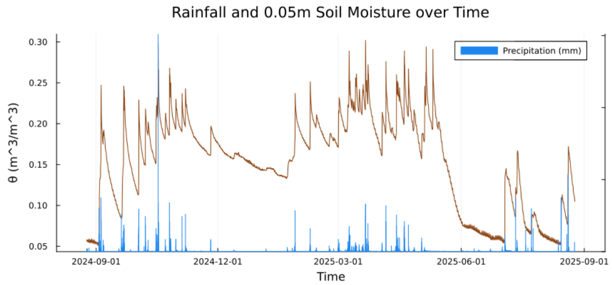


Figure 10: Precipitation events (right axis) and 0.05m soil moisture response (left axis).

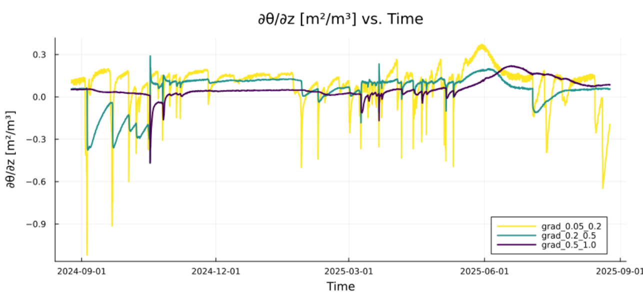


Figure 11: Vertical soil moisture gradients between sensor depths.

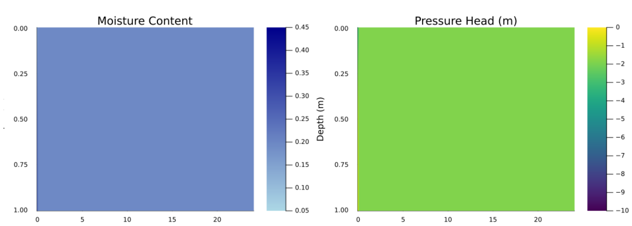


Figure 12: Example of a failed convergence for 24hr that results in a uniform and trivial solution.

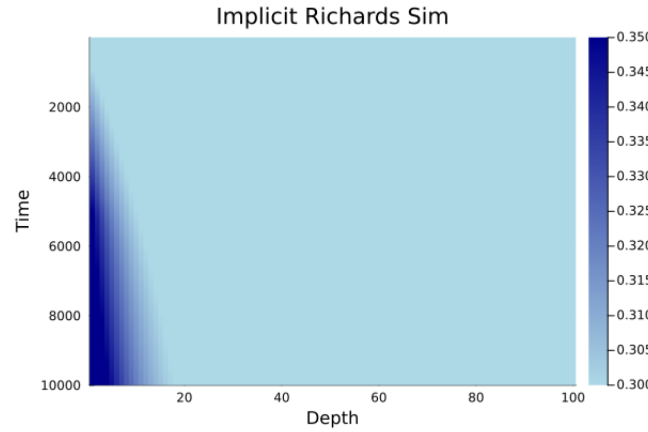


Figure 13: Example an Implicit Euler method for solving Richards Equation.

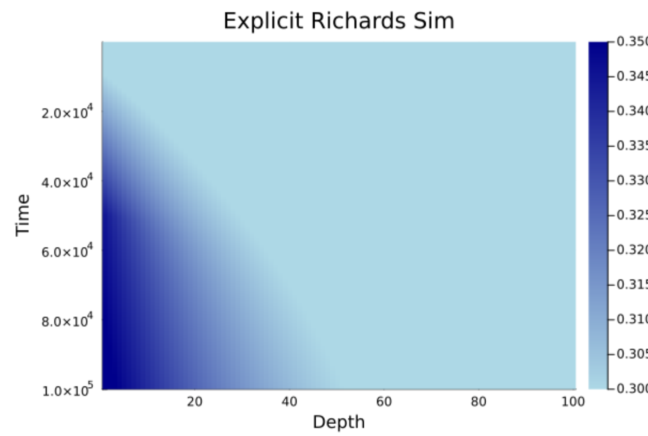


Figure 14: Example an Explicit Euler method for solving Richards Equation.

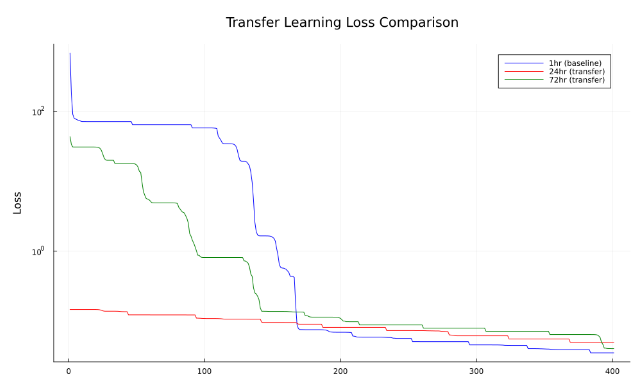


Figure 15: Transfer Learning loss vs. iteration for 1hr baseline model and 24hr/72hr transfer learning PINN.