Highlights

Physics-Informed Neural Networks for solving transient unconfined groundwater flow

Daniele Secci,Vanessa A.Godoy,J. Jaime Gómez-Hernández

- Unconfined aquifer transient flow is solved using PINN.
- PINNs accurately compute the time-varying phreatic surface and piezometric heads.
- PINNs have proven to be very effective in data-scarce environments.
- PINNs are a promising alternative to classical numerical methods in hydrogeology.

Physics-Informed Neural Networks for solving transient unconfined groundwater flow

Daniele Secci[a,b], Vanessa A.Godoy[b] and J. Jaime Gómez-Hernández[b]

[a] *Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy*
[b] *Institute for Water and Environmental Engineering, Universitat Politècnica de València, 46022 Valencia, Spain*

ARTICLE INFO

Keywords:
Physics-informed neural networks
Unconfined aquifer
Machine learning
Numerical modeling
Space and time-varying boundary condition

ABSTRACT

Neural networks excel in various machine learning applications; however, they lack the physical interpretability and constraints crucial for numerous scientific and engineering problems. This limitation hinders their ability to accurately capture and predict complex physical systems' behavior, potentially yielding inaccurate or unreliable results. Physics-Informed Neural Networks (PINNs) are a class of machine learning models that integrate the power of neural networks with the physical laws governing natural phenomena. PINNs provide an effective tool for solving intricate physical problems, ranging from fluid dynamics to materials science, by incorporating physical constraints into the neural network architecture. PINNs can substantially enhance the accuracy and efficiency of model predictions, even in data-limited situations. This work offers insight into recent developments in the PINN field, including their mathematical formulation and training algorithms, and emphasizes their application in solving transient unconfined groundwater flow. In this context, the phreatic surface acts as a spatiotemporally varying boundary condition, and properly accounting for its position is vital for precise predictions of unconfined groundwater flow and related environmental and engineering applications. The study's objective is to develop a reliable model for estimating the phreatic surface and the spatiotemporal distribution of piezometric heads in a vertical cross-section of an unconfined aquifer. Two cases are examined: the first involves a homogeneous and isotropic aquifer, while the second comprises a mildly heterogeneous and anisotropic one. The challenges and opportunities arising from this emerging research area are also explored, and essential directions for future research are underscored.

CRediT authorship contribution statement

Daniele Secci: Conceptualization, Methodology, Investigation, Software, Writing - original draft. **Vanessa A.Godoy:** Conceptualization, Methodology, Validation, Writing - review & editing. **J. Jaime Gómez-Hernández:** Conceptualization, Methodology, Writing - review & editing, Supervision.

1. Introduction

Understanding unconfined groundwater flow is crucial for managing water resources, safeguarding water quality, and mitigating environmental impacts such as land subsidence and saltwater intrusion in coastal aquifers. However, solving unconfined groundwater flow is not trivial, as it necessitates considering spatially and temporally varying boundary conditions. To simplify the problem, Dupuit and Forchheimer introduced some assumptions (Bear, 2012, Eq. 4-64), which Boussinesq later generalized (Bear, 2012, Eq. 5-76). Although the Boussinesq equation serves as a helpful but simplified model, it overlooks some complex physical processes that can occur in unconfined groundwater flow, such as vertical flows and high hydraulic gradients. Consequently, it may not be accurate in certain scenarios, such as near pumping wells, discharge points in coastal aquifers, or areas with steep topography. Researchers have been

ORCID(s): 0000-0002-0605-0741 (D. Secci); 0000-0002-2594-7351 (V. A.Godoy); 0000-0002-0720-2196 (J.J. Gómez-Hernández)

actively seeking optimal solutions to the simplified unconfined groundwater flow equation for some time (Meenal and Eldho, 2011; Pulido-Velazquez et al., 2007; Taigbenu and Nyirenda, 2010).

Addressing the phreatic surface as a spatiotemporal-variant boundary condition poses challenges due to its complexity and high computational cost (Guo, 1997). This paper investigates the possibility of solving the groundwater flow equation in an unconfined aquifer using Artificial Neural Networks (ANNs). ANNs have become increasingly popular in environmental and water resource studies, owing to their ability to process large amounts of data quickly and accurately (Sit et al., 2020; Tahmasebi and Sahimi, 2021; Mariethoz and Gómez-Hernández, 2021). ANNs are data-driven models that are more cost-effective than process-based models and may capture features that elude the latter. However, ANNs require a significant amount of data to achieve accurate results and lack the physical interpretation offered by process-based models. While some applications of ANNs such as surrogate models for the groundwater flow equation exist in the literature (Asher et al., 2015), their application to unconfined flow, particularly for forecasting purposes, remains limited.

Raissi et al. (2019) recently introduced Physics-Informed Neural Networks (PINNs) to enhance the physical interpretability of classical ANNs and improve their forecasting capabilities. PINNs offer several advantages over traditional physics-based models. By integrating physics-based constraints into the ANN architecture, they enable the model to better capture the underlying physics of the system being modeled. This integration can result in more accurate and reliable predictions, especially in scenarios where traditional models may struggle due to high complexity or data scarcity. PINNs provide flexibility and generalizability, as they can be trained on limited or noisy data, and can handle complex geometries and boundary conditions without implementing a specific mesh. This versatility makes PINNs a powerful tool for solving a wide range of physical problems, including unconfined groundwater flow, without requiring an in-depth understanding of the underlying physics. Computational efficiency is a key feature of PINNs. Once trained, they can be evaluated rapidly, making them an efficient tool for real-time decision-making or optimization problems. Moreover, PINNs can be parallelized and run on GPUs, allowing for faster simulations and higher throughput.

This physics-informed approach has been applied to various fields, including fluid dynamics, materials science, geophysics, and others (bin Waheed et al., 2021; Bajracharya and Jain, 2022; Cai et al., 2021; Mao et al., 2020; Lv et al., 2021; Zheng and Wu, 2023). For further information on the state of the art of PINNs, readers are referred to the recent publication by Lawal et al. (2022).

The ability of PINNs to incorporate physical constraints into neural networks, manage intricate geometries and boundary conditions, and work with limited data make them suitable for simulating unconfined groundwater flow. However, their use for this purpose has been seldom explored (Shadab et al., 2021; Zhang et al., 2022).

In this paper, we employ PINNs to compute the phreatic surface and piezometric heads in a synthetic unconfined aquifer. Unlike previous researchers (Shadab et al., 2021; Zhang et al., 2022), we consider the phreatic surface as a spatiotemporal-varying boundary condition with unknown geometry, which must satisfy the condition that piezometric head equals elevation at the phreatic surface. We use the groundwater flow partial differential equation (PDE) in a transient unconfined aquifer as the underlying model, without simplifications. We demonstrate the application of PINNs in two scenarios —an isotropic and homogeneous aquifer, and an anisotropic and heterogeneous one— and compare the results with the finite difference numerical solution provided by a numerical model implemented with MODFLOW 2005 (Harbaugh, 2005)

2. Material and Methods

In the next sections, we will briefly explore the basics of ANNs, the PDE that describes the groundwater flow in an unconfined aquifer, followed by the fundamentals of PINNs.

2.1. Artificial neural networks

ANNs are a powerful class of machine learning algorithms that have gained popularity in recent years due to their ability to model complex patterns in data. Inspired by the structure and function of the human brain and nervous system, ANNs have a rich history dating back several decades. The concept of ANNs can be traced to the work of McCulloch and Pitts (1943), who proposed a mathematical model of a neuron. This model laid the foundation for the first ANNs, which were used for pattern recognition and classification. Initially hindered by computational resource limitations, ANNs have now been applied to a wide range of problems across various fields (Fang et al., 2023; Naghipour et al., 2023; Juan and Valdecantos, 2022; Wang et al., 2022; Dimitriadou and Nikolakopoulos, 2022).

An ANN comprises a collection of interconnected processing units called artificial neurons, which can receive and transmit signals to one another. Each artificial neuron possesses a set of weights determining the influence of incoming signals on its output. Typically, the output of an artificial neuron is a nonlinear function of the weighted sum of its inputs plus a bias term, such as the sigmoid or rectified linear unit (ReLU) function. The output of one artificial neuron can serve as input for another artificial neuron in the subsequent layer, forming a multilayer network. The input layer receives external data, the output layer produces the desired response, and the hidden layers perform intermediate computations. This process is represented as follows

$$a_o = g(\boldsymbol{w}^T \boldsymbol{a} + b), \tag{1}$$

where a_o is the (scalar) neuron output, $\boldsymbol{a}$ is the input vector of all neurons from the previous layer connecting to this one, $\boldsymbol{w}$ is a weight vector, T stands for transpose, b is a bias term , and g is a non-linear activation function. (The term $\boldsymbol{w}^T \boldsymbol{a} + b$ is generally represented with symbol z.)

An ANN can learn from data by adjusting its weights using a learning algorithm, such as backpropagation or gradient descent. The algorithm compares the network output with the desired output and computes an error measure; then, the weights and biases are modified with the aim of reducing the error.

2.2. Partial differential equation for unconfined groundwater flow

The PDE that describes unconfined flow in a two-dimensional (in the vertical plane XZ) and heterogeneous aquifer under transient conditions is:

$$\begin{aligned}\frac{\partial}{\partial x}\left(K_{xx}(x,z)\frac{\partial h}{\partial x}(x,z,t)\right)+\frac{\partial}{\partial z}\left(K_{zz}(x,z)\frac{\partial h}{\partial z}(x,z,t)\right)\\ =S(x,z)\frac{\partial h}{\partial t}(x,z,t)+W(x,z,t)\end{aligned} \tag{2}$$

where $K_{xx}(x,z)$ and $K_{zz}(x,z)$ $[LT^{-1}]$ are the principal values of the hydraulic conductivity tensor, assuming that the principal directions are parallel to axes x and z, t $[T]$ is time, $h(x,z,t)$ $[L]$ is piezometric head, W $[T^{-1}]$ is an external flow extraction rate per unit volume, and S $[L^{-1}]$ is specific storage.

Subject to the initial condition $h(x,z,0)=h_0(x,z)$, with h_0 being a known function, and boundary conditions. The boundary conditions can be the standard ones in groundwater flow modeling, such as prescribed head, prescribed flow, or prescribed linear combination of head and flow. However, there is a specific boundary condition for unconfined aquifers that renders the solution of the above partial differential equation particularly challenging: the phreatic surface is the top boundary condition and must satisfy

$$h(x,z)=z \quad \forall(x,z) \quad \text{along the phreatic surface,} \tag{3}$$

with the complication that the phreatic surface position is not known a priori and that it will change over time.

2.3. Physics-informed neural networks (PINNs)

PINNs are a type of ANNs designed to include constraints during its training to ensure that it abides by certain fundamental relationships (Raissi et al., 2019), such as the conservation of mass. This type of ANNs has proven extremely effective in solving complex PDEs in a meshless domain, outperforming, in some cases, standard numerical methods (Raissi et al., 2019; Yang et al., 2021; He and Tartakovsky, 2021; Rezaei et al., 2022; Zhang et al., 2022). PINNs combine supervised learning and physics-based constraints. The supervised learning component involves minimizing a loss function that represents the error between predicted and observed labels, according to a classic ANN. The physics-based constraints are encoded as additional loss terms that penalize the model for violating physical laws, such as the PDE in (2). This hybrid approach allows PINNs to learn complex relationships between variables while also respecting physical constraints, with the ability to handle problems that are difficult to model using traditional physics-based approaches and work in data-scarce scenarios (He et al., 2020). They have also proven their ability to make good extrapolations (either in space or time), where traditional ANNs fail, thanks to the physics embedded in their training (Rezaei et al., 2022; Almajid and Abu-Al-Saud, 2022).

The goal of a PINN is to satisfy the governing PDE as well as the boundary and initial conditions such that its loss function is defined as the sum of the mean squared errors for the prediction residual, PDE residual, boundary residual, and initial conditions residual (Raissi et al., 2019)

$$\text{Loss} = \text{Loss PRED} + \text{Loss PDE} + \text{Loss BC} + \text{Loss IC}. \tag{4}$$

3. PINN to solve the unconfined groundwater flow: synthetic examples

In this section, we explain the architecture of the PINN using synthetic examples for easier reference. The key component of the PINN is the loss function defined above (4); how its terms are computed is presented further down.

3.1. Definition of the domain

A two-dimensional, heterogeneous, and unconfined aquifer is built on the domain $(x, z) \in [0, 1] \times [0, 1]$ (all parameters and simulation results will be given without units, the results remain valid as long as the units used are consistent). For a general case with arbitrary sizes and parameters, an appropriate normalization would transform the original case into this one, and the results should be brought back into the original space by a proper transformation. This procedure may appear non-trivial for complex domains, nevertheless, even when dealing with intricate geometries, it is feasible to scale both coordinates and parameters to a specific range. When working with ANNs in general, the initial step often involves normalizing input variables to a standardized range, typically between 0 and 1. This normalization plays a pivotal role in stabilizing the training of neural networks by mitigating issues such as vanishing gradients and expediting model convergence. Furthermore, by employing this normalized range in the synthetic example, the model acquires the ability to discern patterns and relationships that transcend the specific magnitudes and units of the input variables. This adaptability becomes crucial when applying the model to scenarios featuring diverse sizes and parameters. Additionally, the success of this approach hinges on the reversibility of the normalization process. By meticulously recording the scaling factors and means utilized for normalization, it becomes straightforward to apply an inverse transformation to the model's predictions, effectively reintroducing them into the original variable space. This ensures that the results maintain their interpretability and relevance within the original problem domain. In summary, the utilization of a normalized synthetic example in conjunction with subsequent normalization and transformation procedures for general cases not only bolsters model stability and generalization but also equips the model to address a wide spectrum of scenarios, thereby enhancing its versatility and adaptability.

Transient groundwater flow is simulated with time (t) ranging from 0 to 1. Four specific times are analyzed: $t = 0.01$, $t = 0.25$, $t = 0.5$ and $t = 1$. The bottom boundary is impermeable throughout the simulation. The following transient behavior is modeled. At time 0, the left and right boundary conditions correspond with reservoirs that prescribe heads equal to 1, and the initial heads correspond to the steady-state solution for these conditions, that is, $h(x, z) = 1$, with the phreatic surface coinciding with the top boundary. Suddenly, at time 0, the left reservoir lowers its level down to 0.4 and the right one down to 0.6. The new boundary conditions are, for the left boundary $h(0, z) = 0.4$ for $z \in [0, 0.4]$, undefined for $z > 0.4$, and for the right boundary, $h(1, z) = 0.6$ for $z \in [0, 0.6]$, undefined for $z > 0.6$. This sudden change in the boundary conditions induces a transient behavior that we aim to model with the PINN.

Two synthetic aquifers are analyzed: a homogeneous and isotropic unconfined aquifer (SC1), and a heterogeneous and anisotropic unconfined aquifer (SC2). Tab. 1 presents the geometric and hydraulic characteristics of these two cases. For the purposes of benchmarking, the MODFLOW solution is computed. While the PINN solution is meshless (as it will be explained below), MODFLOW needs the domain to be discretized, and a discretization into 20 by 20 cells in space and 1440 steps in time are chosen. Fig. 1 shows the discretization used, the spatial distribution of conductivity for the heterogeneous case, and the boundary conditions as implemented in MODFLOW after time 0.

For the purpose of training the artificial network, a number of observations within the saturated zone of the aquifer are considered. Specifically, 40 locations in the XZ plane were sampled from the MODFLOW solution for each of the four time steps considered, discarding some locations when they lie above the phreatic surface. In total, 130 observations were used. This way of choosing the observations may seem arbitrary, but its purpose is to have a few control points inside the domain for better training of the network. To ensure that the network also satisfies the PDE (2) with its initial and boundary conditions, we need to identify a number of points were the PDE verification must be done. For this purpose, 1000 points along the left boundary, chosen randomly in the segment $[0, 0] \times [0, 0.4]$, and similarly 1000 points are chosen along the right boundary in the segment $[1, 0] \times [1, 0.6]$. These two sets of points will be used to enforce that the trained network satisfies the prescribed head boundary conditions. Likewise, 1000 points are chosen randomly along the bottom boundary in the segment $[0, 0] \times [1, 0]$ to be used to enforce the network to satisfy the bottom impermeable condition. For the initial conditions, 500 point locations are chosen randomly from the simulation domain at time 0; these locations will serve to enforce the initial conditions. And finally, 25,000 points for SC1 and 35,000 points for SC2 were chosen randomly within the simulation domain $[0, 0] \times [1, 1]$ to be used as collocation points to enforce the reproduction of the PDE.

Table 1
Hydraulic and geometry characteristics of the study domain

Parameters	SC1	SC2
Specific yield	10^{-3}	10^{-3}
Horizontal hydraulic conductivity, K_{xx}	10^{-3}	$K_1 = 4 \cdot 10^{-3}, K_2 = 10^{-3}, K_3 = 2 \cdot 10^{-3}, K_4 = 3 \cdot 10^{-3}$
Vertical hydraulic conductivity, K_{zz}	10^{-3}	$K_1 = 4 \cdot 10^{-4}, K_2 = 10^{-4}, K_3 = 2 \cdot 10^{-4}, K_4 = 3 \cdot 10^{-4}$
Grid spacing in the x direction, Δx	0.05	0.05
Grid spacing in the z direction, Δz	0.05	0.05
Length of the stress periods, Δt	1	1
Total time steps	1440	1440

3.2. Artificial Neural Network Design

The basic design of an ANN is composed of input, output and hidden layers, number of neurons per layer, batch size, number of epochs, a loss function and the learning and decay rates. As the loss function is the key component of the PINNs its definition will be detailed in the next subsection.

In this work, two structurally identical neural networks (Fig. 2), with the only difference being their input and output layers, are used. The first network (ANN1) is trained to compute the piezometric head value (output) using the point coordinates (x, z) and the time (t) as inputs. The second network (ANN2), which takes the x coordinate and the time as input values, returns the z_s coordinate value (output) that indicates the position of the free surface at a specific time. Although both networks could be trained simultaneously from the beginning using a single loss function, we found that it is more efficient if there is a preliminary iteration in which ANN1 is trained first, and then ANN2 is trained next (with ANN1 fixed). The weights and biases found in this preliminary iteration are used as the starting values for the joint training of the two networks.

Each artificial network consists of an input layer, seven hidden layers (each comprising 20 neurons), and an output layer. Functionally, the ANN can be viewed as a differentiable system, consisting of a series of multivariable vector-valued functions, which include affine transformations and linear or nonlinear functions (the activation functions), mapping from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_3}$

$$\mathbb{R}^{d_1} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_2} \Rightarrow \mathbb{R}^{d_3}. \tag{5}$$

where d_1 and d_3 represent the dimensions of the input and the output layers, respectively. In this study, d_1 is three for ANN1 and two for ANN2, d_2 is the number of neurons in the hidden layer (20 in this case for ANN1 and ANN2) and d_3 is one for ANN1 and ANN2.

The choice of the number of hidden layers, the number of points where the evaluations of the performance of the network should be done, the choice of the activation function, and the rest of the hyperparameters needed for the definition of the networks were chosen after some initial tests. These initial tests were carried out manually and aimed to identify configurations that optimize the performance of the ANN in terms of minimizing errors while also ensuring efficient processing times. Specifically, the activation function is the hyperbolic tangent (tanh), the number of epochs is 200, the mini-batch size is 128, the initial learning rate is 0.01, and the decay rate is 0.005.

The signal moves from one layer to the next following (1) applied to all neurons in the layer

$$\boldsymbol{a_d} = g(\boldsymbol{W} \cdot \boldsymbol{a_u} + \boldsymbol{b}) \tag{6}$$

where subscripts d and u refer to downstream and upstream, respectively, $\boldsymbol{a}$ is the output vector of a given layer, matrix $\boldsymbol{W}$, with dimensions $d_d \times d_u$, contains all the weights applying to the current layer and $\boldsymbol{b}$, with dimensions $d_d \times 1$, contains all the bias terms. During the training phase, these weights and biases are optimized to minimize the loss functions using the Adam algorithm (Kingma and Ba, 2015), not only because of its robustness and popularity in deep learning but also due to its capacity to accelerate convergence by using adaptive learning rates for different parameters based on estimates derived from both the first and second moments of the gradients.

3.2.1. First neural network (ANN1)

The ANN1 (Fig. 3) is responsible for predicting the piezometric head as a function of three input parameters: the spatial coordinates (x, z) and the temporal variable (t). These spatiotemporal coordinates do not have to lie on a mesh, and therefore, once the network is trained, it provides a solution detached from any space-time discretization: it is a meshless solution.

The training of the network is based on the minimization of a loss function with the following components:

1. Loss associated with the error in reproducing the observed values, which equals the average of the squared differences between observations and predictions at the chosen locations and times.

$$L_{IC} = \sum_{\Gamma} \left(h_{\text{predicted}} - h_{\text{observed}}\right)^2 \tag{7}$$

   where:
   - the summation symbol $\sum_{\Gamma}$ represents the summation over space corresponding to the observed values.
   - $h_{\text{predicted}}$ is the predicted hydraulic head.
   - h_{observed} is the observed hydraulic head as a function of space and time.

2. Loss associated with the error in reproducing the initial conditions, which equals the sum of the squared differences between the network prediction at time zero and the known initial values at the sampled locations

$$L_{OBS} = \sum_{\Gamma} \left(h_{\text{predicted}} - h_{\text{initial}}\right)^2 \tag{8}$$

where:

- the summation symbol $\sum_{\Gamma}$ represents the summation over space corresponding to the initial conditions.

- $h_{\text{predicted}}$ is the predicted hydraulic head.

- h_{initial} is the initial hydraulic head as a function of space.

3. Loss associated with the error in reproducing the known heads at the prescribed head boundaries, which equals the average of the squared differences between the network prediction and the known prescribed heads at the four chosen time steps

$$L_{BC} = \sum_{\Gamma} \left(h_{\text{predicted}} - h_{\text{specified}}\right)^2 \tag{9}$$

where:

- the summation symbol $\sum_{\Gamma}$ represents the summation over spatial and temporal regions corresponding to the boundary conditions.

- $h_{\text{predicted}}$ is the predicted hydraulic head.

- $h_{\text{specified}}$ is the initial hydraulic head as a function of space and time.

4. Loss associated with the error in reproducing the no flow boundary. Using automatic differentiation (Griewank, 1998), it is possible to evaluate any partial derivative of the artificial network output (h) with respect to the input variables (x, z, t); therefore, $\frac{\partial h}{\partial z}$ can be evaluated at the selected points along the bottom boundary, and the average squared difference with respect to its known value of zero computed for each of the chosen time steps.

$$L_{\text{noflow}} = \sum_{\Omega} \left(f(x, z, t, h, \frac{\partial h}{\partial x}, \frac{\partial h}{\partial z}) \right)^2 \tag{10}$$

In this expression:

- Ω represents the spatial and temporal domain over which the no flow boundary conditions is solved.

- h is the predicted solution by the neural network.

- $f(x, z, t, h, \frac{\partial h}{\partial x}, \frac{\partial h}{\partial z})$ is the no flow residual, which depends on the predicted solution h and its derivatives with respect to x and z.

5. Loss associated with the error in reproducing the PDE. Again, thanks to automatic differentiation, and as displayed in Fig. 3, the partial derivatives involved in (2) can be computed at the collocation points selected. After rearranging all terms in (2) so that they equal zero, the average squared sum of the PDE values computed with the heads provided as output from the neural network at the collocation points and the four selected times will represent the associated error.

$$L_{\text{residual}} = \sum_{\Omega} \left(f(x, z, t, h, \frac{\partial h}{\partial x}, \frac{\partial h}{\partial z}, \frac{\partial h}{\partial t}) \right)^2 \tag{11}$$

In this expression:

- Ω represents the spatial and temporal domain over which the PDE is solved.

- h is the predicted solution by the neural network.

- $f(x, z, t, h, \frac{\partial h}{\partial x}, \frac{\partial h}{\partial z}, \frac{\partial h}{\partial t})$ is the PDE residual, which depends on the predicted solution h and its derivatives with respect to x, z, and t.

3.2.2. *Second neural network (ANN2)*

The objective of the ANN2 network (Fig. 4) is to identify the spatial coordinate (z_s) that corresponds to the phreatic surface, by taking (x, t) as input variables. The loss function is now defined as the sum of squared differences between the value of the elevation z_s given as output and the piezometric head predicted at that location by ANN1, which should equal the elevation. This sum is computed for 2500 points (SC1) and 3500 points (SC2) randomly distributed in [0,1] and the four times chosen.

The training of the ANN2 is based on the minimization of a loss function with two components:

1. Loss associated with the error in reproducing the phreatic surface, which equals the sum of the squared difference between the ANN2 predictions and the piezometric head predicted by ANN1.

$$L_{BC} = \sum_{\Gamma} \left(z_{s_{\text{predicted}}} - h_{\text{predicted}} \right)^2 \tag{12}$$

where:

- the summation symbol $\sum_{\Gamma}$ represents the summation over spatial and temporal regions corresponding to the phreatic surface prediction.

- $z_{s_{\text{predicted}}}$ is the predicted elevation of the phreatic surface by ANN2.

- $h_{\text{predicted}}$ is the predicted hydraulic head by ANN1.

2. Loss associated with the error in reproducing the initial conditions, which equals the sum of the squared differences between the network predictions and the known initial values for the phreatic surface.

$$L_{OBS} = \sum_{\Gamma} \left(z_{s_{\text{predicted}}} - z_{s_{\text{initial}}} \right)^2 \tag{13}$$

where:

- the summation symbol $\sum_{\Gamma}$ represents the summation over space corresponding to the initial position of the phreatic surface.

- $z_{s_{\text{predicted}}}$ is the predicted phreatic surface elevation at time $t = 0$.

- $z_{s_{\text{initial}}}$ is the initial phreatic surface elevation as a function of space.

As already mentioned, ANN1 is trained first, then ANN2 is trained using the output from ANN1; afterwards, both networks are trained simultaneously using as loss function the sum of the functions described for each network. We found this approach more efficient than trying to train both networks simultaneously from the beginning. Given the interdependence between the elevation of the free surface, denoted as the output of ANN2, and the hydraulic head, represented as the output of ANN1, within an unconfined aquifer, it was more efficient to predict the phreatic surface elevation using ANN2 subsequent to the training of ANN1. This approach optimizes computational processes by leveraging the information acquired from the initial neural network, thus enhancing predictive accuracy and efficiency with smaller values of the final loss function.

3.3. Performance evaluation

The solution of the PDE given by MODFLOW will be used to assess the performance of the implemented PINN. The Root Mean Squared Error (RMSE) is used to compare the results obtained by the fully trained PINN and the numerical model

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(\hat{h}_i - h_i)^2}{N}} \tag{14}$$

where N is the number of verification points (in space and time), h_i is the MODFLOW predicted value and $\hat{h}_i$ is the PINN predicted value.

Also the Mean Absolute Error (MAE) is selected as reference metric, since the RMSE could be sensitive to outliers.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{h}_i - h_i| \tag{15}$$

Furthermore, the Nash-Sutcliffe Efficiency (NSE) is used to add a goodness-of-fit measure

$$NSE = 1 - \frac{\sum_{i=1}^{n}(h_i - \hat{h}_i)^2}{\sum_{i=1}^{n}(h_i - \bar{h})^2} \tag{16}$$

where $\bar{h}$ is the mean of the MODFLOW predicted values.

In evaluating this comparison, one should not forget that the numerical solution is already based on an approximation of the PDE and may not be exact. This can be noticed, for instance, in the resolution with which the phreatic surface is represented in MODFLOW versus its representation by the PINN approximation, which is meshless and, therefore, can provide a much smoother result.

4. Results

4.1. Training phase

As already mentioned, we found that the most efficient way to train both neural networks was to train ANN1 first for 200 epochs, then train ANN2 for 200 epochs freezing ANN1, and then train both jointly for another 200 epochs. The evolution of the loss functions for the two case studies can be seen in Figs. 5 and 6, where the loss function values and the elapsed times on an Intel(R) Core(TM) i9-10920X CPU 3.50GHz RAM 32GB are displayed. In both scenarios, the final loss is around 10^{-5}, which represents a reduction of at least three orders of magnitude with respect to the initial loss computed with a random initialization of the weights and biases.

It is important to emphasize that the available data primarily consists of physics-based information, leaving only a limited amount of prior knowledge for training the model. As a result, the risk of encountering overfitting issues, where the PINN memorizes the limited training data instead of generalizing well, is significantly reduced. Moreover, by allocating most of the loss function's effort to enforcing physical constraints and using a smaller portion of the data for training, we prioritize the model's ability to capture the underlying physics while maintaining robust and reliable performance on unseen data, avoiding underfitting. This approach aligns with the inherent characteristics of PINNs and their effectiveness in tackling intricate, physics-driven problems. Consequently, we opted to incorporate all the available prior information into the training data, without creating a separate validation dataset.

Table 2
Homogeneous aquifer: RMSE of the estimated solution by the PINN compared to the one obtained by the numerical model

Time	RMSE
0.01	0.0423
0.25	0.0098
0.50	0.0093
1.00	0.0099

Table 3
Homogeneous aquifer: MAE of the estimated solution by the PINN compared to the one obtained by the numerical model

Time	MAE
0.01	0.0239
0.25	0.0070
0.50	0.0064
1.00	0.0053

Table 4
Homogeneous aquifer: NSE of the goodness-of-fit between the PINN solution compared to the one obtained by the numerical model

Time	NSE
0.01	0.90
0.25	0.0099
0.50	0.0099
1.00	0.0098

4.2. Testing phase

Once the networks have been trained, they are validated by comparing the network predictions with the results obtained by MODFLOW. The RMSE and the MAE are computed for the piezometric heads at the center points of the discretization grid and the elevation of the phreatic surface at times 0.01, 0.25. 0.5 and 1. The values predicted by the neural networks are obtained by feeding the coordinates (x, z, t) to ANN1 and (x, t) to ANN2. Also, a visual comparison of the piezometric head maps and phreatic surfaces is carried out.

4.3. Unconfined homogeneous isotropic aquifer (SC1)

Fig. 7 shows the discrepancy between the network predictions and the MODFLOW predictions at the four selected times, while Tables 2 and 3 show the RMSE and the MAE computed at the center of the active discretization cells.

Overall, the errors are small, with the largest errors occurring at $t = 0.01$ when both models are simulating the sudden drop of the prescribed heads along the boundaries. Then, the error decreases as the simulation approaches the stationary condition, as indicated by the decreasing value of the RMSE and the MAE. Table 4 depict the NSE highlighting the good fit between the PINN prediction and the numerical solution.

Fig. 8 shows the piezometric head maps at the selected time $t = 0.01$ as derived from the MODFLOW simulation

Table 5
Heterogeneous aquifer: RMSE of the estimated solution by the PINN compared to the one obtained by the numerical model

Time	RMSE
0.01	0.0437
0.25	0.0136
0.50	0.0122
1.00	0.0047

Table 6
Heterogeneous aquifer: MAE of the estimated solution by the PINN compared to the one obtained by the numerical model

Time	MAE
0.01	0.0231
0.25	0.0109
0.50	0.0103
1.00	0.0039

Table 7
Heterogeneous aquifer: NSE of the goodness-of-fit between the PINN solution compared to the one obtained by the numerical model

Time	NSE
0.01	0.87
0.25	0.95
0.50	0.96
1.00	0.99

and from the PINN prediction. It should be noticed that the MODFLOW maps are pixel maps based on the discretization used to solve the equation, whereas the PINN maps have been built with a denser discretization, taking advantage of the meshless nature of the neural network. This is particularly noticeable in Fig. 9 where the delineation of the phreatic surface, for the other three investigated times, is quite jaggy in the MODFLOW solution but smooth in the PINN one.

4.4. Unconfined heterogeneous anisotropic aquifer (SC2)

Fig. 10 shows the discrepancy between the network predictions and the MODFLOW predictions at the four selected times, while Table 5 and 6 show the RMSE and the MAE computed at the center of the active discretization cells.

The RMSE and the MAE errors between the PINN and numerical model predictions for all investigated times are reported in Tables 5 and 6, with generally small errors observed. Again, the largest error occurred at the initial starting time, which is expected, as both models are simulating a sudden drop along the boundaries at time zero. As the simulation approaches the stationary condition, the errors decrease. Table 7 depict the NSE highlighting the good fit between the PINN prediction and the numerical solution.

Fig. 11 and Fig. 12 show the piezometric head maps at the selected time $t = 0.01$ and $t = 0.25$, $t = 0.5$, $t = 1$, as derived from the MODFLOW simulation and from the PINN prediction. As for the homogeneous case, the piezometric heads obtained by the PINN and the numerical model display similar overall behavior and patterns.

4.4.1. Heterogeneous aquifer: PINN vs conventional ANN

Comparing the performance of PINNs to that of a conventional neural network is of interest. Conventional neural networks only rely on the utilization of a priori information, specifically, known data. In this particular case, to train a conventional neural network effectively, a substantial amount of groundwater level data over time would be required to serve as target data during network training. In this specific case study, the known data consists of assigned boundary head conditions, initial conditions, and only a sparse set of data within the domain (10% of active cells from the MODFLOW simulation, referencing only four specific time steps). Even the impermeable boundary condition is incorporated into the loss function as physics constrain through automatic differentiation. Furthermore, it is essential to note that, albeit synthetic, the treated case study is physically complex. The drastic change in boundary conditions initiates a transient flow behavior far from trivial. With the limited data used for PINN training, excluding the underlying physics, the network lacks the necessary information for effective training and achieving high-performance results, as observed in the metrics reported in Tables 5, 6 and 7. Moreover, training the network with sparse data within the domain is inadequate for describing an appropriate training range to represent the spatial extent of the domain itself. As thoroughly examined by literature, conventional neural networks are unable to extrapolate beyond the training range. Therefore, the use of PINNs, leveraging the underlying physics, enables the creation of a network capable of extrapolation based on physical knowledge, facilitating the development of a meshless model that yields reliable and functional results compared to a conventional neural network. In the accompanying Fig. 13, it is evident that the output of the conventional neural network, trained solely with available a priori information, is incapable of reproducing the flow field.

5. Discussion and Conclusions

This study demonstrates the successful application of Physics-Informed Neural Networks (PINNs) for solving forward groundwater flow problems in unconfined aquifers, both for homogeneous isotropic and heterogeneous anisotropic aquifers, keeping in mind that the partial differential equation controlling flow in unconfined aquifers has a spacetime-varying boundary condition associated to the position of the phreatic surface that makes it a specially difficult problem to handle. The following discussion emphasizes the major conclusions and implications of this research.

Firstly, our findings confirm the ability of PINNs to accurately compute piezometric head values in unconfined aquifer systems, as demonstrated by the small errors between the PINN and numerical model predictions (see Tables 2 and 5). The errors are notably smaller for later times when the simulation approaches a stationary condition. This corroborates previous research that has shown the effectiveness of PINNs in solving complex problems across various fields, such as fluid mechanics and geosciences (bin Waheed et al., 2021; Bajracharya and Jain, 2022; Cai et al., 2021; Mao et al., 2020; Lv et al., 2021; Zheng and Wu, 2023).

Secondly, we show that incorporating physical constraints can dramatically reduce the number of required observations for training a simple ANN, emphasizing the PINN's potential to work in data-scarce environments. ANNs are difficult, if not impossible, to train in data-limited scenarios. In contrast, PINNs efficiently leverage available data and incorporate physics constraints to address data scarcity. This makes PINNs particularly valuable for situations with limited data availability, such as remote or difficult-to-access regions, or cases where data collection is expensive or time-consuming.

Moreover, our findings underscore the potential of PINNs to complement or replace traditional numerical models in simulating unconfined aquifer flow problems. With faster computation times and the ability to handle complex datasets, PINNs are a promising alternative for modeling and simulation in hydrogeology.

However, certain limitations should be considered. The accuracy of PINN results can be influenced by the neural network architecture and the quality of the training data. Hyperparameters, such as the number of layers, neurons, and the learning rate, can significantly impact the results. Although this study aimed to evaluate the effectiveness of PINNs for solving the forward flow problems in an unconfined aquifer with manually calibrated hyperparameters, future work could explore sensitivity analysis or auto-selection tools for optimizing these parameters. Additionally, PINNs require a large number of collocation points in which to evaluate the physical constraints, which can make the training phase time-consuming, especially when observation data are limited.

Another topic worthy of discussion is the impact of data errors on the outcomes of the AI model. In the context of neural networks, this specific issue has been extensively scrutinized, corroborating the capacity of neural networks to undergo training that accommodates measurement inaccuracies (Coppola et al., 2003; Secci et al., 2022). The established procedure entails the delineation of a plausible range of measurement errors, followed by the specification of a corresponding error distribution. Subsequently, each individual data point is subjected to perturbation by an error drawn from this distribution a predetermined number of times, with the quantity of perturbations contingent upon the complexity of the underlying problem. Furthermore, each perturbed data point maintains a consistent association with the same target value as the "clean" data point. While this approach inherently demands a computationally more intensive training process, it furnishes the neural network with the capability to effectively manage and adapt to measurement errors. Given that PINNs exhibit a significant reduction in the requisite training data for even basic ANN models, forthcoming researches could incorporate this facet to enhance the reliability of models, especially in real-world scenarios.

Furthermore, future research endeavors could focus on the implementation of three-dimensional (3D) unconfined problems to evaluate whether the increased complexity introduced by an additional input dimension (the y-coordinate) imposes limitations on performance or necessitates a significant increase in the volume of data and collocation points required for effective training.

In conclusion, this study showcases the effectiveness of using PINNs to solve unconfined aquifer flow problems, with accurate estimates of time-varying phreatic surface and piezometric head values. The use of PINNs offers an alternative, efficient approach to addressing complex groundwater flow problems. This research contributes to the development of a more accurate and efficient tool for groundwater modeling, with potential applications across environmental management, civil engineering, and hydrogeology. Future research can focus on investigating the potential of PINNs for solving other groundwater problems, including contaminant transport, heterogeneity characterization, and anisotropy.

Acknowledgements

The corresponding author wishes to express his deep gratitude to the IAMG Student Affairs Committee for awarding him with the Computers and Geosciences Research Scholarship for the project "Physics-Informed Neural Networks (PINNs) for subsurface hydrology" that supported the completion of the current study. This work was developed under the scope of the InTheMED project. InTheMED is part of the PRIMA programme supported by the European Union's HORIZON 2020 research and innovation programme under grant agreement No 1923.

Code availability section

Name of code: Unconflow-PINN

Developer: Daniele Secci

Contact address: Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy

E-mail: daniele.secci@unipr.it

Year first available: 2023

Hardware required: none

Software required: Matlab

Program language: Matlab

Program size: 2.8 MB

Source codes: https://github.com/godoyva/Unconflow-PINN

References

Almajid, M.M., Abu-Al-Saud, M.O., 2022. Prediction of porous media fluid flow using physics informed neural networks. Journal of Petroleum Science and Engineering 208, 109205. URL: https://www.sciencedirect.com/science/article/pii/S0920410521008597, doi:https://doi.org/10.1016/j.petrol.2021.109205.

Asher, M.J., Croke, B.F., Jakeman, A.J., Peeters, L.J., 2015. A review of surrogate models and their application to groundwater modeling. Water Resources Research 51, 5957–5973.

Bajracharya, P., Jain, S., 2022. Hydrologic similarity based on width function and hypsometry: An unsupervised learning approach. Computers & Geosciences 163, 105097. URL: https://www.sciencedirect.com/science/article/pii/S0098300422000590, doi:https://doi.org/10.1016/j.cageo.2022.105097.

Bear, J., 2012. Hydraulics of groundwater. Dover Publications.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E., 2021. Physics-informed neural networks for heat transfer problems. Journal of Heat Transfer 143.

Coppola, E., Szidarovszky, F., Poulton, M., Charles, E., 2003. Artificial neural network approach for predicting transient water levels in a multilayered groundwater system under variable state, pumping, and climate conditions. Journal of Hydrologic Engineering 8, 348–360. doi:10.1061/(asce)1084-0699(2003)8:6(348).

Dimitriadou, S., Nikolakopoulos, K.G., 2022. Artificial neural networks for the prediction of the reference evapotranspiration of the peloponnese peninsula, greece. Water 14, 2027.

Fang, Y., Jairi, I., Pirhadi, N., 2023. Neural transfer learning for soil liquefaction tests. Computers & Geosciences 171, 105282. URL: https://www.sciencedirect.com/science/article/pii/S009830042200231X, doi:https://doi.org/10.1016/j.cageo.2022.105282.

Guo, W., 1997. Transient groundwater flow between reservoirs and water-table aquifers. Journal of Hydrology 195, 370–384.

Harbaugh, A.W., 2005. MODFLOW-2005, the US Geological Survey modular ground-water model: the ground-water flow process. volume 6. US Department of the Interior, US Geological Survey Reston, VA, USA.

He, Q., Barajas-Solano, D., Tartakovsky, G., Tartakovsky, A.M., 2020. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. Advances in Water Resources 141, 103610. URL: https://www.sciencedirect.com/science/article/pii/S0309170819311649, doi:https://doi.org/10.1016/j.advwatres.2020.103610.

He, Q., Tartakovsky, A.M., 2021. Physics-informed neural network method for forward and backward advection-dispersion equations. Water Resources Research 57, e2020WR029479.

Juan, N.P., Valdecantos, V.N., 2022. Review of the application of artificial neural networks in ocean engineering. Ocean Engineering 259, 111947.

Lawal, Z.K., Yassin, H., Lai, D.T.C., Che Idris, A., 2022. Physics-informed neural network (pinn) evolution and beyond: A systematic literature review and bibliometric analysis. Big Data and Cognitive Computing 6. URL: https://www.mdpi.com/2504-2289/6/4/140, doi:10.3390/bdcc6040140.

Lv, A., Cheng, L., Aghighi, M.A., Masoumi, H., Roshan, H., 2021. A novel workflow based on physics-informed machine learning to determine the permeability profile of fractured coal seams using downhole geophysical logs. Marine and Petroleum Geology 131, 105171.

Mao, Z., Jagtap, A.D., Karniadakis, G.E., 2020. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering 360, 112789.

Mariethoz, G., Gómez-Hernández, J.J., 2021. Machine learning for water resources. Frontiers in Artificial Intelligence 4, 63.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas imminent in nervous activity. Bull. Math. Biophys. 5, 115–133.

Meenal, M., Eldho, T., 2011. Simulation of groundwater flow in unconfined aquifer using meshfree point collocation method. Engineering Analysis with Boundary Elements 35, 700–707.

Naghipour, L., Aalami, M.T., Nourani, V., 2023. Collective dynamics analysis based on the multiplex network method to unravel the backbone of fluctuations in groundwater level data. Computers & Geosciences 172, 105310. URL: https://www.sciencedirect.com/science/article/pii/S0098300423000146, doi:https://doi.org/10.1016/j.cageo.2023.105310.

Pulido-Velazquez, D., Sahuquillo, A., Andreu, J., Pulido-Velazquez, M., 2007. A general methodology to simulate groundwater flow of unconfined aquifers with a reduced computational cost. Journal of Hydrology 338, 42–56.

Raissi, M., Perdikaris, P., Karniadakis, G., 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. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125, doi:https://doi.org/10.1016/j.jcp.2018.10.045.

Rezaei, S., Harandi, A., Moeineddin, A., Xu, B.X., Reese, S., 2022. A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method. Computer Methods in Applied Mechanics and Engineering 401, 115616. URL: https://www.sciencedirect.com/science/article/pii/S0045782522005722, doi:https://doi.org/10.1016/j.cma.2022.115616.

Secci, D., Molino, L., Zanini, A., 2022. Contaminant source identification in groundwater by means of artificial neural network. Journal of Hydrology 611. doi:10.1016/j.jhydrol.2022.128003.

Shadab, M.A., Luo, D., Shen, Y., Hiatt, E., Hesse, M.A., 2021. Investigating steady unconfined groundwater flow using physics informed neural networks. arXiv preprint arXiv:2112.13792 .

Sit, M., Demiray, B.Z., Xiang, Z., Ewing, G.J., Sermet, Y., Demir, I., 2020. A comprehensive review of deep learning applications in hydrology and water resources. Water Science and Technology 82, 2635–2670.

Tahmasebi, P., Sahimi, M., 2021. Special issue on machine learning for water resources and subsurface systems.

Taigbenu, A., Nyirenda, E., 2010. Revisiting the stream-aquifer flow problem with a flux-based Green element model. Water SA 36, 287 – 294. URL: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S1816-79502010000300009&nrm=iso.

bin Waheed, U., Haghighat, E., Alkhalifah, T., Song, C., Hao, Q., 2021. Pinneik: Eikonal solution using physics-informed neural networks. Computers & Geosciences 155, 104833. URL: https://www.sciencedirect.com/science/article/pii/S009830042100131X, doi:https://doi.org/10.1016/j.cageo.2021.104833.

Wang, G., Jia, Q.S., Zhou, M., Bi, J., Qiao, J., Abusorrah, A., 2022. Artificial neural networks for water quality soft-sensing in wastewater treatment: a review. Artificial Intelligence Review 55, 565–587.

Yang, L., Meng, X., Karniadakis, G.E., 2021. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. Journal of Computational Physics 425, 109913.

Zhang, X., Zhu, Y., Wang, J., Ju, L., Qian, Y., Ye, M., Yang, J., 2022. GW-PINN: A deep learning algorithm for solving groundwater flow equations. Advances in Water Resources 165, 104243. URL: https://doi.org/10.1016/j.advwatres.2022.104243, doi:10.1016/j.advwatres.2022.104243.

Zheng, Y., Wu, Z., 2023. Physics-informed online machine learning and predictive control of nonlinear processes with parameter uncertainty. Industrial & Engineering Chemistry Research 62, 2804–2818.

List of Figures

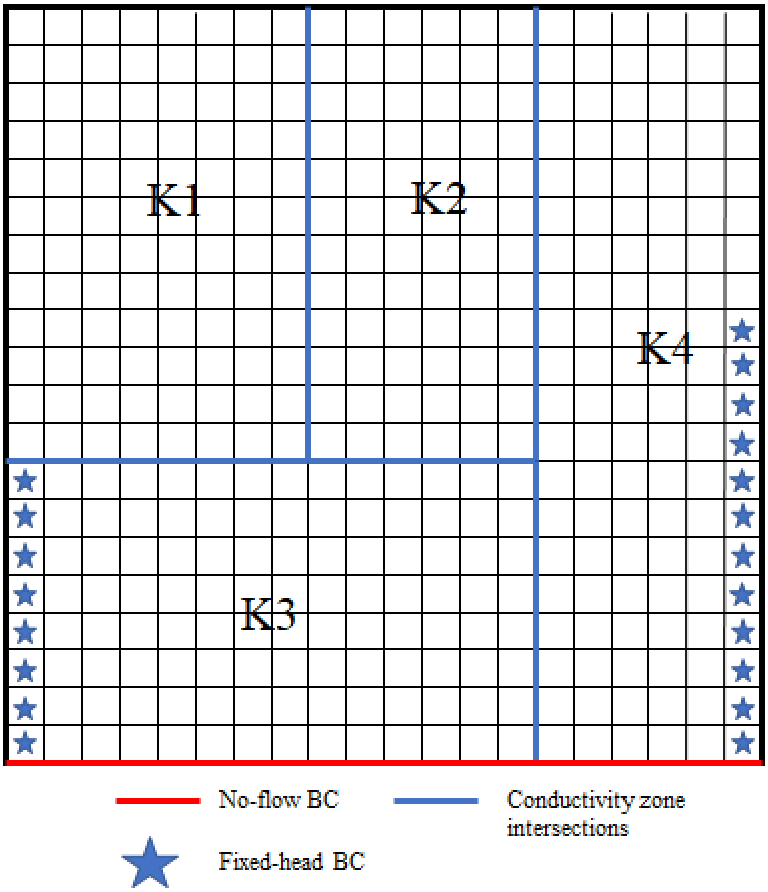

Figure 1: Synthetic domain.

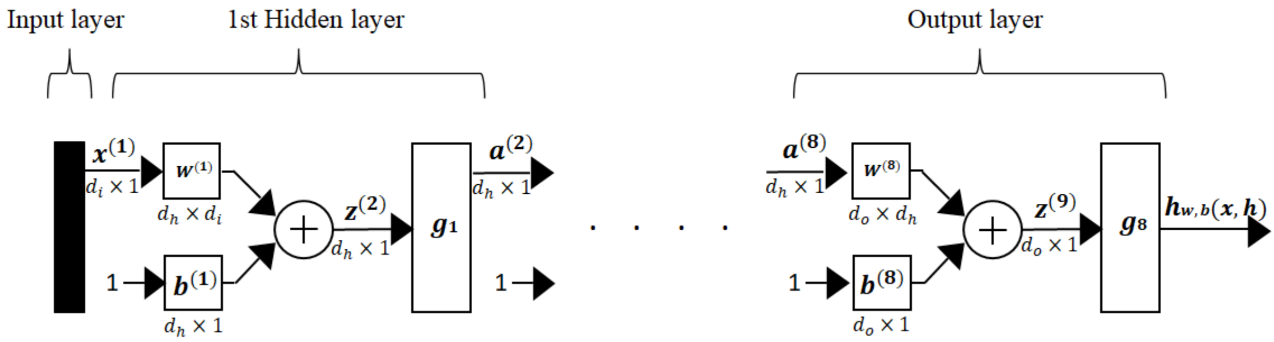

Figure 2: Sketch of the implemented neural networks.

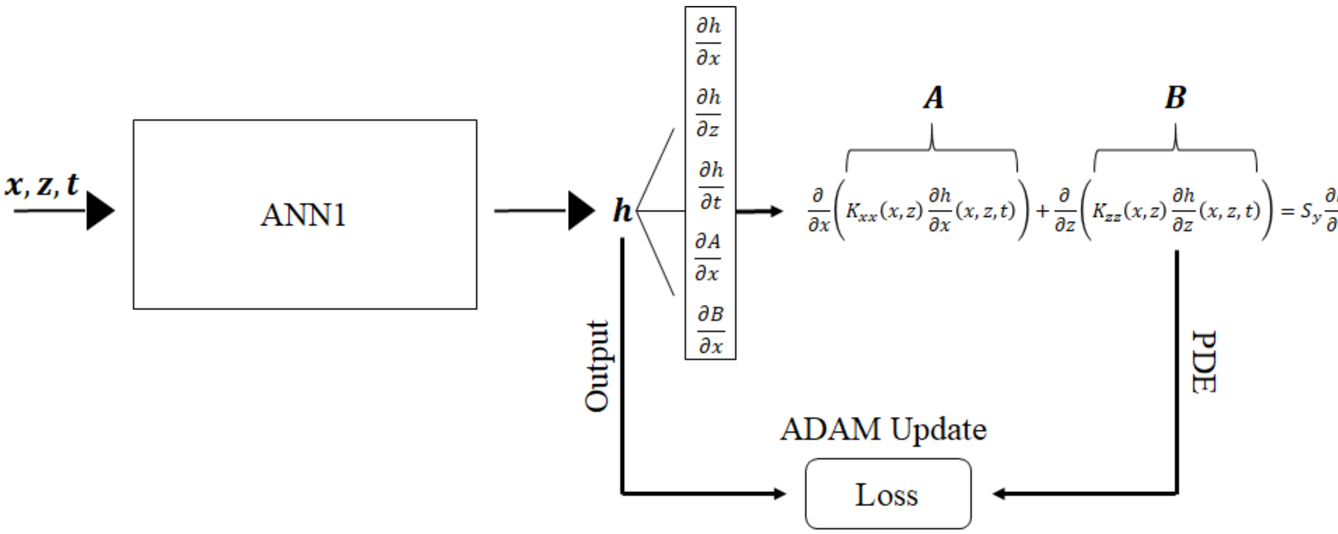

Figure 3: Sketch of ANN1.

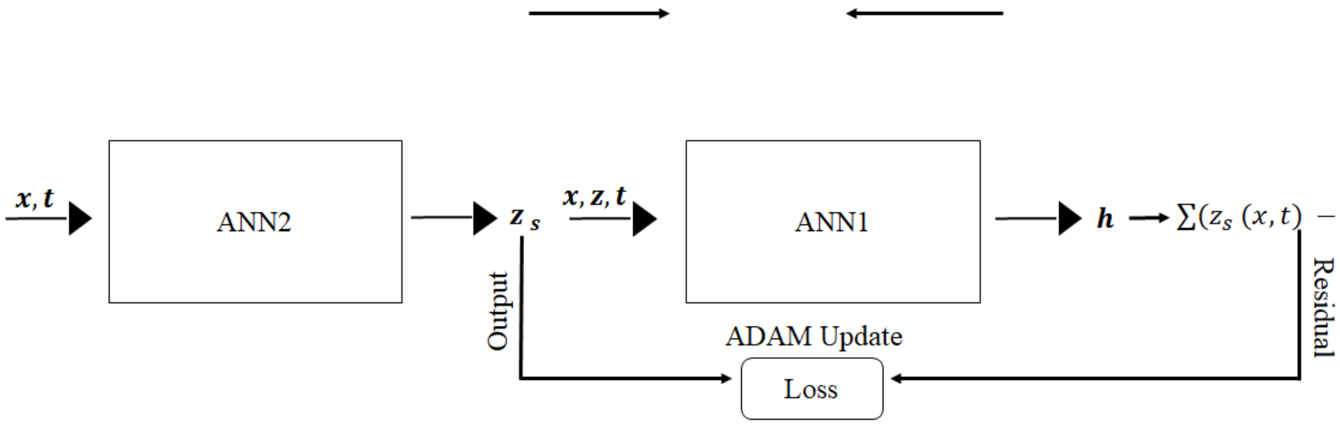

Figure 4: Sketch of ANN2.

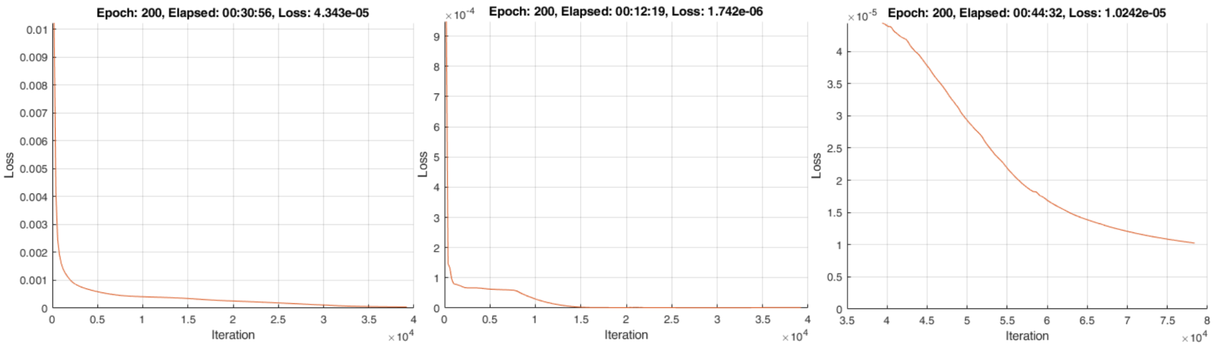

Figure 5: Scenario SC1. Training loss for ANN1 alone (left), ANN2 alone with ANN1 frozen (center) and ANN1 jointly with ANN2 (right). The iteration axis in the right plot starts at the number of iterations already performed in the previous training.

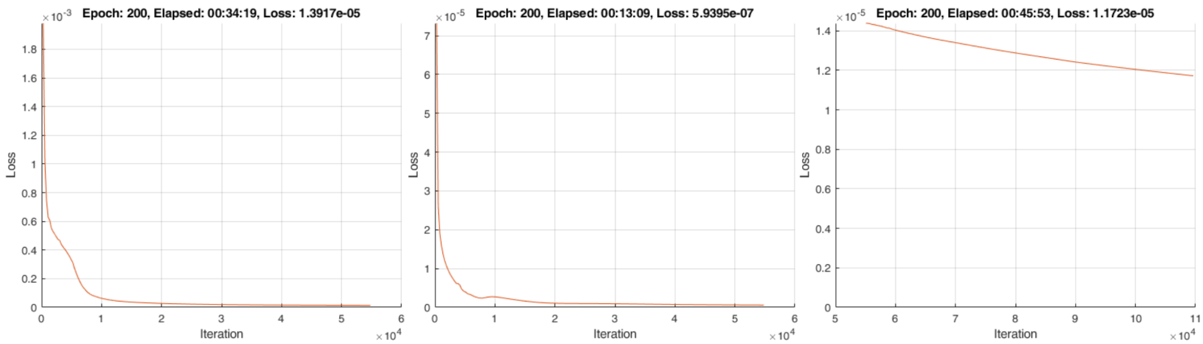

Figure 6: Scenario SC2. Training loss for ANN1 alone (left), ANN2 alone with ANN1 frozen (center) and ANN1 jointly with ANN2 (right). The iteration axis in the right plot starts at the number of iterations already performed in the previous training.

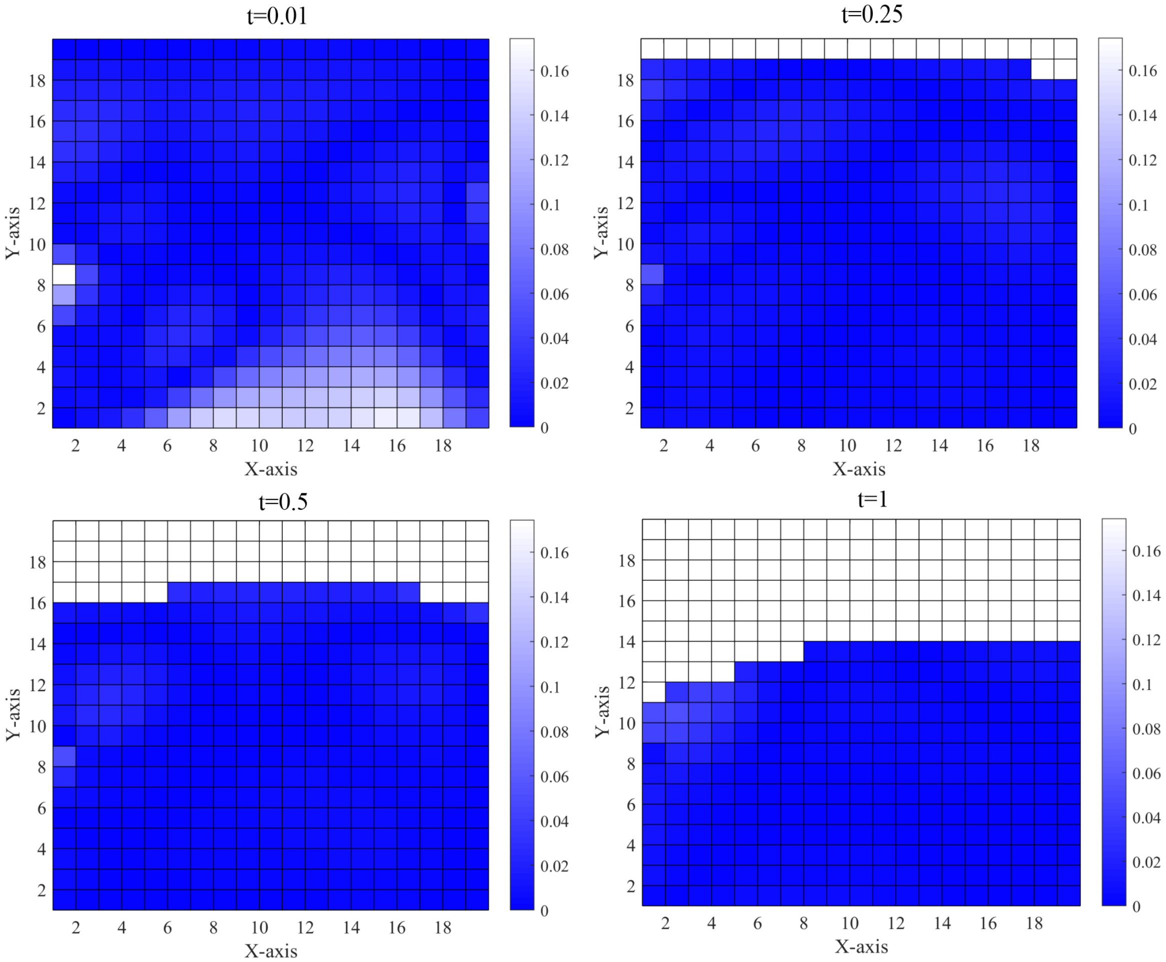

Figure 7: Homogeneous aquifer: Error plot of the estimated piezometric field (PINN minus MODFLOW), using the active cells in the numerical model.

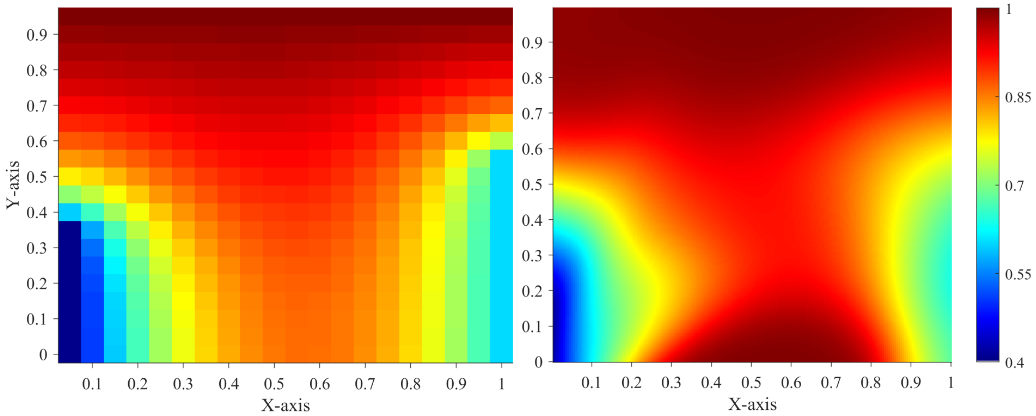

Figure 8: Homogeneous aquifer: Estimated piezometric field by the numerical model (left) and PINN (right) for the selected time $t = 0.01$.

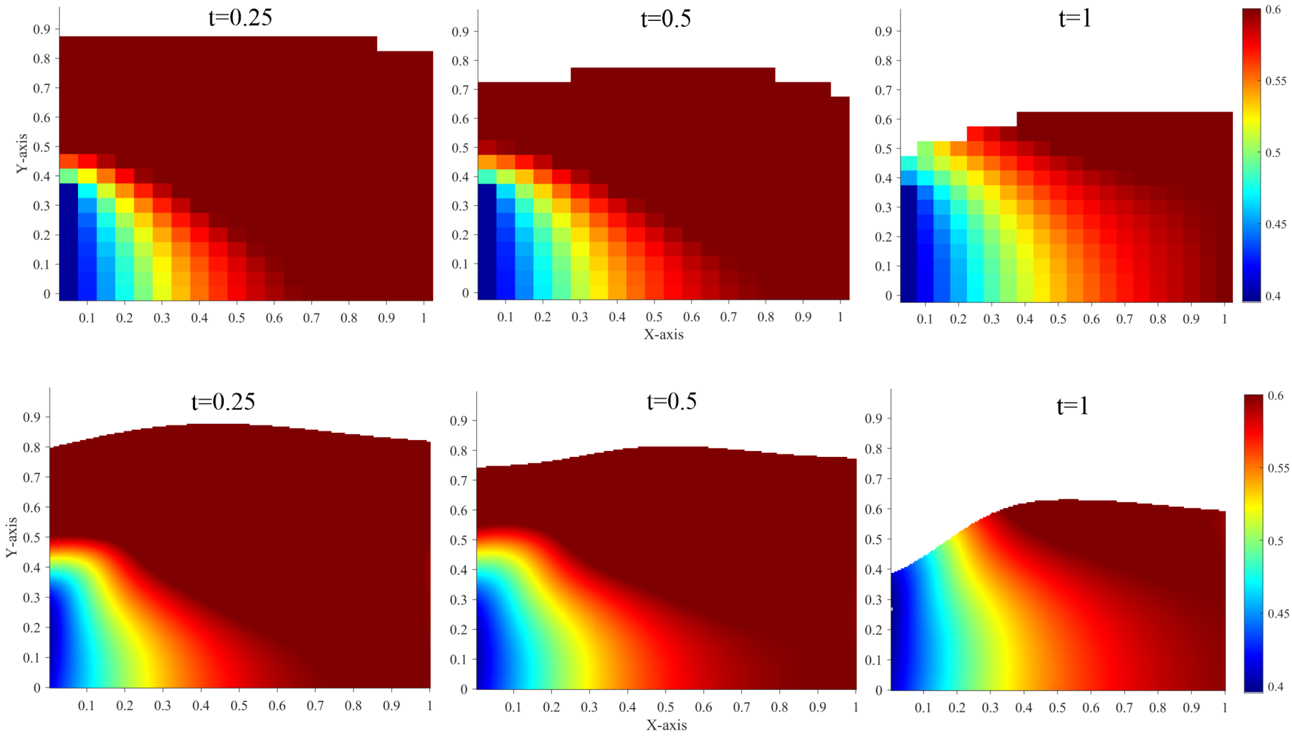

Figure 9: Homogeneous aquifer: Estimated piezometric field by the numerical model (top) and PINN (bottom) for the selected time $t = 0.25$, $t = 0.5$, $t = 1$.

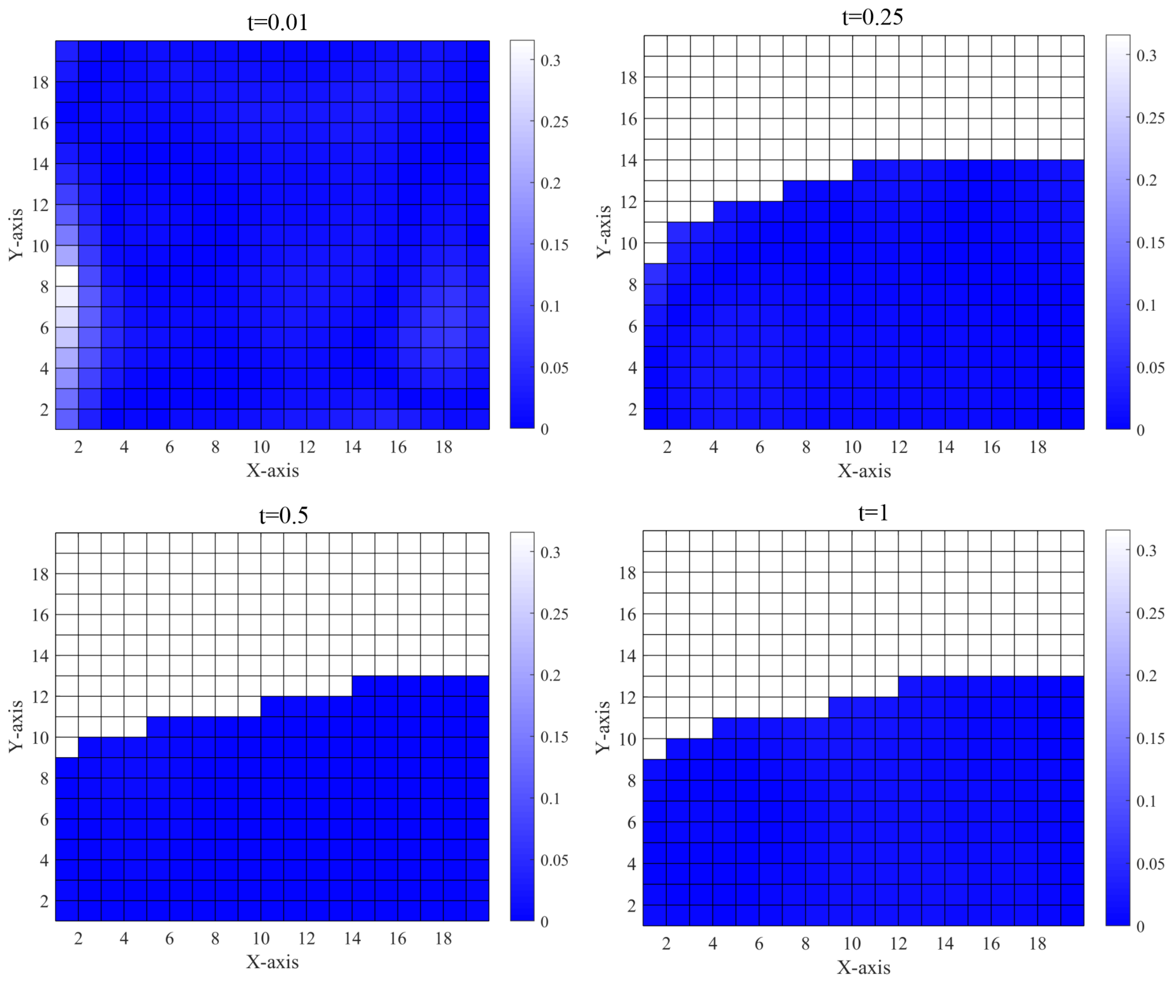

Figure 10: Heterogeneous aquifer: Error plot of the estimated piezometric field (PINN minus MODFLOW), using the active cells in the numerical model.

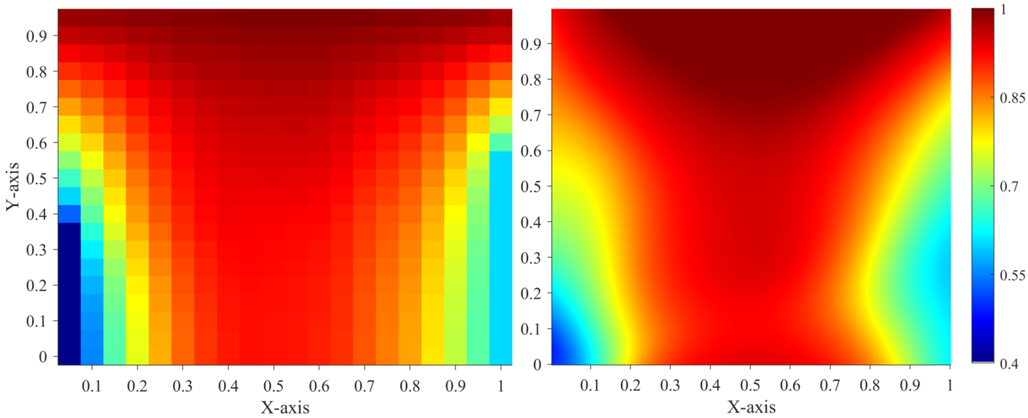

Figure 11: Heterogeneous aquifer: Estimated piezometric field by the numerical model (left) and PINN (right) for the selected time $t = 0.01$.

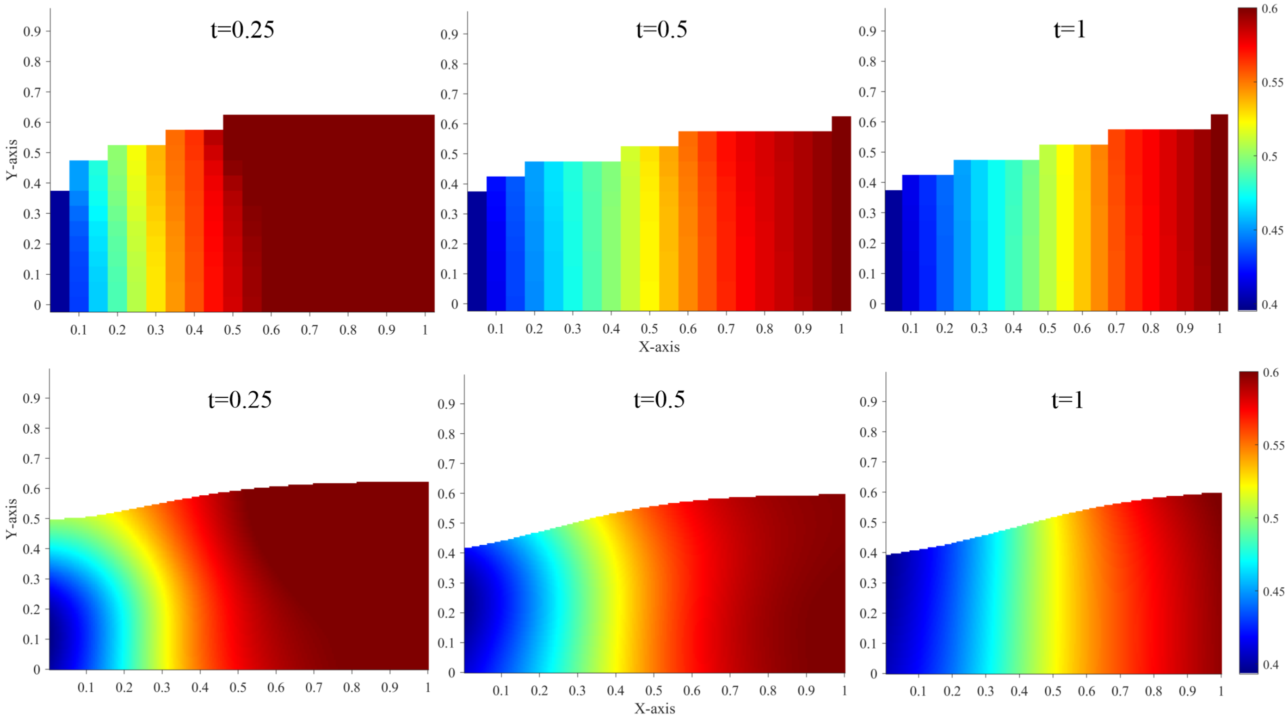

Figure 12: Heterogeneous aquifer: Estimated piezometric field by the numerical model (top) and PINN (bottom) for the selected time $t = 0.25$, $t = 0.5$, $t = 1$.

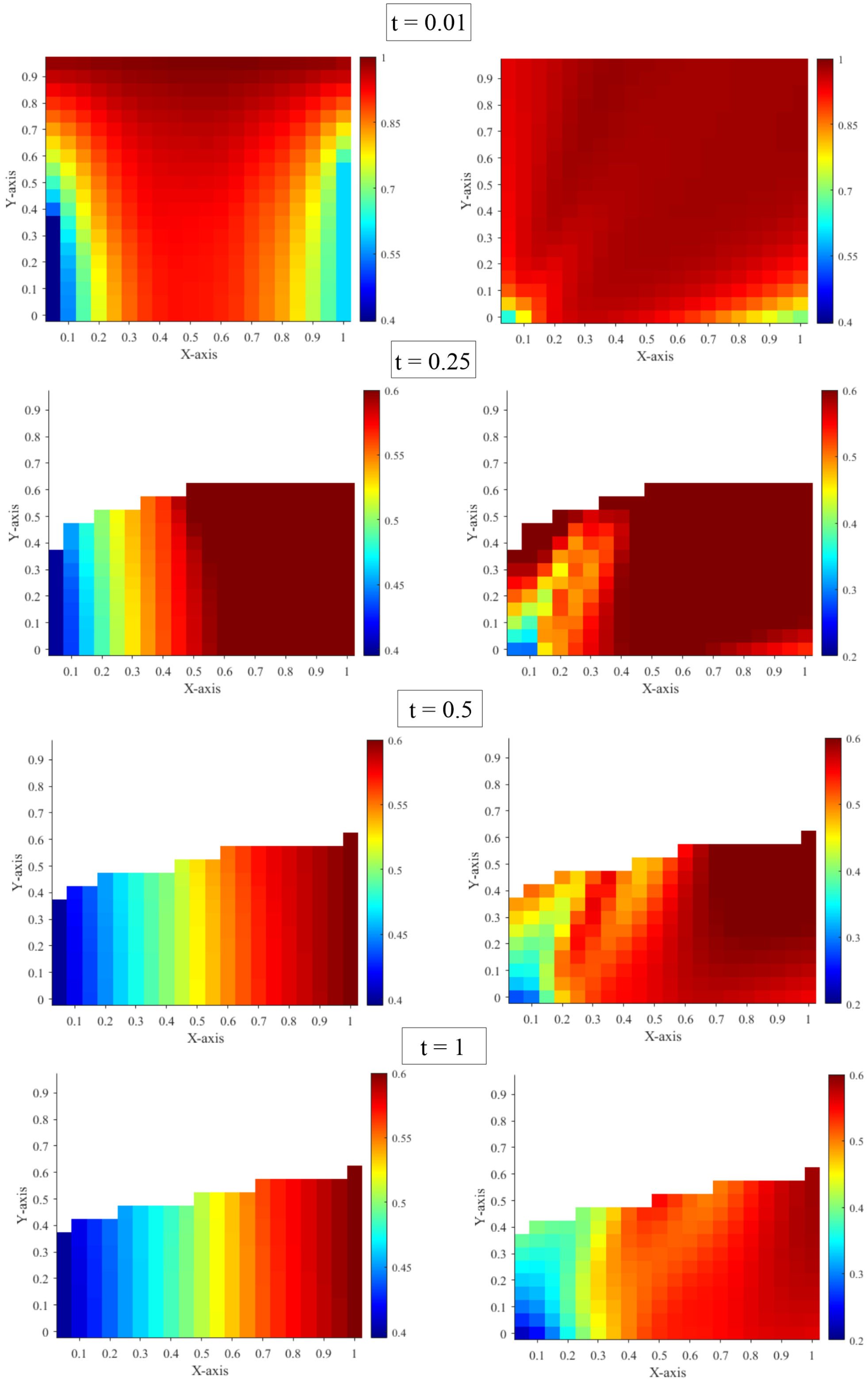

Figure 13: Heterogeneous aquifer. Left: numerical solution of the piezometric field with respect to the active cells at time $t = 0$, $t = 0.25$, $t = 0.5$ and $t = 1$. Right: conventional ANN prediction of the piezometric field with respect to the active cells at time $t = 0$, $t = 0.25$, $t = 0.5$ and $t = 1$.