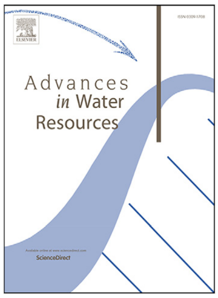
# Journal Pre-proof




Investigating steady unconfined groundwater flow using Physics Informed Neural Networks

Mohammad Afzal Shadab, Dingcheng Luo, Eric Hiatt, Yiran Shen, Marc Andre Hesse

PII: S0309-1708(23)00080-5
DOI: https://doi.org/10.1016/j.advwatres.2023.104445
Reference: ADWR 104445

To appear in: *Advances in Water Resources*

Received date : 20 September 2022
Revised date : 19 April 2023
Accepted date : 24 April 2023

Please cite this article as: M.A. Shadab, D. Luo, E. Hiatt et al., Investigating steady unconfined groundwater flow using Physics Informed Neural Networks. *Advances in Water Resources* (2023), doi: https://doi.org/10.1016/j.advwatres.2023.104445.

# Highlights

## Investigating Steady Unconfined Groundwater Flow using Physics Informed Neural Networks

Mohammad Afzal Shadab,Dingcheng Luo,Eric Hiatt,Yiran Shen,Marc Andre Hesse

- PINNs predict hydraulic conductivity and water table heights using experimental data.
- PINNs eliminate inability of Dupuit-Boussinesq equation when predicting seepage face.
- Inclusion of physics improves PINNs predictions compared to plain neural networks.

# Investigating Steady Unconfined Groundwater Flow using Physics Informed Neural Networks

Mohammad Afzal Shadab[a,b,*], Dingcheng Luo[a], Eric Hiatt[b,c], Yiran Shen[a] and Marc Andre Hesse[a,c]

[a] Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 E. 24th Street, C0200, Austin, TX 78712, United States
[b] University of Texas Institute for Geophysics, 10601 Exploration Way, Austin, TX 78758, United States
[c] Department of Geological Studies, Jackson School of Geosciences, The University of Texas at Austin, 2305 Speedway, C1160, Austin, TX 78712, United States

ARTICLE INFO

*Keywords*:
Physics-informed neural networks
Unconfined groundwater flow
Laboratory experiments
Boussinesq approximation
Di Nucci model

ABSTRACT

A deep learning technique called Physics Informed Neural Networks (PINNs) is adapted to study steady groundwater flow in unconfined aquifers. This technique utilizes information from underlying physics represented in the form of partial differential equations (PDEs) alongside data obtained from physical observations. In this work, we consider the Dupuit-Boussinesq equation, which is based on the Dupuit-Forchheimer approximation, as well as a recent, more complete model derived by Di Nucci (2018) as underlying models. We then train PINNs on data obtained from steady-state analytical solutions and laboratory based experiments.

Using PINNs, we predict phreatic surface profiles given different input flow conditions and recover estimates for the hydraulic conductivity from the experimental observations. We show that PINNs can eliminate the inherent inability of the Dupuit-Boussinesq equation to predict flows with seepage faces. Moreover, the inclusion of physics information from the Di Nucci and Dupuit-Boussinesq models constrains the solution space and produces better predictions than training on data alone. PINNs based predictions are robust and show a little effect from added noise in the training data. Furthermore, we compare the PINNs solutions obtained via the Di Nucci and Dupuit-Boussinesq flow models to examine the effects of higher order flow terms that are included in the Di Nucci formulation but are neglected by the Dupuit-Boussinesq approximation. Lastly, we discuss the effectiveness of using PINNs for examining groundwater flow.

## 1. Introduction

Large-scale groundwater flow in an unconfined aquifer is often modeled using vertically integrated models resulting in the Dupuit-Boussinesq (or Boussinesq) equation, which reduce the dimensionality of the problems (Boussinesq, 1904; Bear, 1972). These approaches exploit the "shallow nature" of most unconfined aquifers, i.e., their small aspect ratio, $H \ll L$, where $H$ is the average thickness of the saturated zone and $L$ the horizontal extent of the aquifer. The Dupuit-Boussinesq equation, given in Equation (1), is based on the Dupuit-Forchheimer approximation and neglects the effect of vertical flow via the shallow water assumption that results from the order of magnitude analysis of the mass balance, i.e., $v_z/v_x = \mathcal{O}(H/L)$, where $v_z$ and $v_x$ are vertical and horizontal flow velocities respectively (Dupuit, 1863; Forchheimer, 1901; Bear, 1972). The Dupuit-Boussinesq equation has been extended to include the effect of vertical velocity on overall flow dynamics by a series of extended Boussinesq equations (Di Nucci, 2018). These equations have been used to describe the water wave propagation in porous media as a consequence of wave interactions with structures and tide-induced fluctuations (Di Nucci, 2018).

One important problem in using Boussinesq-type equations is the inability to account for the formation of a seepage face. A seepage face typically forms at steep lateral boundaries of the aquifer, where groundwater debouches into atmospheric pressure (Figure 1). The seepage face, by definition, is a boundary at which the hydraulic pressure head becomes zero or equivalently, the potentiometric head becomes the height of the saturated groundwater table. Analysis of the seepage face is a central component of many geotechnical, hydrogeological and geomorphological studies. In hydrology, seepage analysis is of interest for the design of hydraulic structures such as earth dams or river embankments (Simpson, Clement and Gallop, 2003; Scudeler, Paniconi, Pasetto and Putti, 2017; Hiatt, Shadab, Hesse and Gulick, 2021). Some models attempt to include seepage face dynamics by computational means, such as boundary cell deactivation or simplified extensions of the Boussinesq equation, however these approaches lack the underlying physics of the system (Baird, Mason and Horn, 1998; Di Nucci, 2018; Rushton and Youngs, 2010). Though few models attempt to capture the physics, a recent mathematical model developed by Di Nucci (2018) accounts for both vertical flow effects and seepage face development while still neglecting capillary fringe effects. To understand the hydrologic conditions in which either Dupuit-Boussinesq or Di Nucci model is most applicable, it is imperative to compare both models with experimental data. Di Nucci's formulation requires a far field, free boundary gradient of zero in order to obtain steady state analytic results. Due

*Corresponding author
mashadab@utexas.edu (M.A. Shadab); mhesse@jsg.utexas.edu (M.A. Hesse)
https://mashadab.github.io/ (M.A. Shadab); https://www.jsg.utexas.edu/hesse/marc-hesse/ (M.A. Hesse)
ORCID(s): 0000-0002-0797-5017 (M.A. Shadab); 0000-0002-2532-3274 (M.A. Hesse)

to the scale of laboratory experiments, this is not possible. Consequently, a data-based comparison of the two ordinary differential equation models is required.

In the past, artificial neural networks have been used to predict the behavior of seepage flows (Ma, Huang, Liu, Morin, Aziz and Meints, 2020; Rehamnia, Benlaoukli, Jamei, Karbasi and Malik, 2021; Tayfur, 2014; Nourani and Babakhani, 2013). However, artifical neural networks, alone, lack the essential physics described by partial differential equation (PDE) models. To incorporate the underlying physics, information provided by PDE models can be integrated into the training of the neural networks. In particular, Dissanayake and Phan-Thien (1994) proposed a method of solving PDEs by representing the PDE solution as a neural network, and minimizing a loss function defined in terms of the residual of the PDE. This approach was further developed and popularized more recently by Raissi, Perdikaris and Karniadakis (2019) to tackle both forward and inverse problems, referring to it as "Physics Informed Neural Networks (PINNs)". In addition to improving the accuracy of predictions, the physics based PINNs method can simultaneously infer PDE model parameters, such as hydraulic conductivity. Furthermore, the PINNs method overcomes the inability of Dupuit-Boussinesq equation to predict the seepage face by supplementing the Dupuit-Boussinesq equation with additional information through the training data. The PINNs method has also been successfully implemented in diverse fields such as fluid mechanics (Brunton, Noack and Koumoutsakos, 2020; Raissi, Yazdani and Karniadakis, 2020; Jin, Cai, Li and Karniadakis, 2021), ocean engineering (Jagtap, Mitsotakis and Karniadakis, 2022), nondestructive testing Shukla, Di Leoni, Blackshire, Sparkman and Karniadakis (2020); Shukla, Jagtap, Blackshire, Sparkman and Karniadakis (2021), cardiology (Sahli Costabal, Yang, Perdikaris, Hurtado and Kuhl, 2020) and optics (Chen, Lu, Karniadakis and Dal Negro, 2020; van Herten, Chiribiri, Breeuwer, Veta and Scannell, 2020).

In groundwater applications, PINNs have been employed to invert for model parameters and constitutive relationships for steady-state cases using synthetically generated data (Meng and Karniadakis, 2020; Tartakovsky, Marrero, Perdikaris, Tartakovsky and Barajas-Solano, 2020; He, Barajas-Solano, Tartakovsky and Tartakovsky, 2020; Bandai and Ghezzehei, 2020; Zhang, Zhu, Wang, Ju, Qian, Ye and Yang, 2022). However, Depina, Jain, Mar Valsson and Gotovac (2021) is the only work that uses PINNs technique with data from laboratory scale, porous media experiments, and considers unsaturated groundwater flow using Richards' equation to find van-Genuchten (van Genuchten, 1980) model parameters, soil moisture profiles from synthetic data, and measurements of a one-dimensional vertical water infiltration column test. In contrast, we focus on the two-dimensional problem of steady unconfined flow with a seepage face. In this aim, a data-based comparison of Dupuit-Boussinesq and Di Nucci models is required to understand the effects of higher order, vertical flow terms and the conditions for which each approximation remains appropriate.

In this work, we apply the PINNs technique to investigate the dynamics of the water table with a seepage face. First, we train PINNs using synthetic data, where "ground truths" are available, to demonstrate its predictive capabilities. We then apply this technique to experimental data, and go on to predict free surface profiles and recover the hydraulic conductivity from training data. Next, we compare the two models of unconfined groundwater flow using PINNs. Finally, we discuss the effectiveness of using PINNs to examine steady groundwater flows and predict free surface profiles and seepage face heights.

The remainder of this paper is summarized as follows: Sections 2 and 3 revisit the theories underpinning the two physics-based groundwater flow models and physics informed neural networks. Section 4 focuses on the specific application of PINNs to investigate steady unconfined groundwater flow. Section 5 discusses the methods involved in generating synthetic and experimental data. Section 6 and 7 summarize the salient results when applying PINNs and plain neural network on synthetic and experimental data respectively. Section 8 discusses the result's implications on groundwater flow, and it is followed by conclusions in section 9.

## 2. Physics based groundwater flow models

### 2.1. Boussinesq equation

For unsteady and unconfined flows in a homogeneous porous media, the Dupuit-Boussinesq equation is the most widely used to approximate flow. (Boussinesq, 1904). It is based on the Dupuit-Forchheimer approximation, which assumes dominant horizontal flow driven by the gradient of the groundwater table (Dupuit, 1863; Forchheimer, 1901). By implication, the water column at any horizontal location is in hydrostatic equilibrium and the gradients are only due to the lateral variance of pressure in the groundwater table. In the absence of a source term, i.e., no recharge, and a level, impervious base, the Dupuit-Boussinesq equation can be written as

$$\phi\frac{\partial h}{\partial t} - \frac{\partial}{\partial x}\left(Kh\frac{\partial h}{\partial x}\right) = 0, \quad t \in (0,\infty), \quad x \in (0,L), \tag{1}$$

where $x$ is the horizontal spatial coordinate (m), $h(x)$ is the height of the free surface above the impervious base (m), $\phi$ (-) is the porosity of the medium (-), and $K$ is the hydraulic conductivity (m/s). The porous medium is assumed to be homogeneous and isotropic. At steady-state, Equation (1) reduces to the following nonlinear boundary value problem

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(Kh\frac{\mathrm{d}h}{\mathrm{d}x}\right) = 0, \quad x \in (0,L), \tag{2}$$

which can be solved analytically given appropriate boundary conditions. For the steady seepage problem shown in Figure 1 we have the following boundary conditions

$$h(x=0,\infty) = \text{constant}, \quad q(x=L,\infty) = -Kh\frac{\mathrm{d}h}{\mathrm{d}x}\bigg|_{x=L}. \tag{3}$$

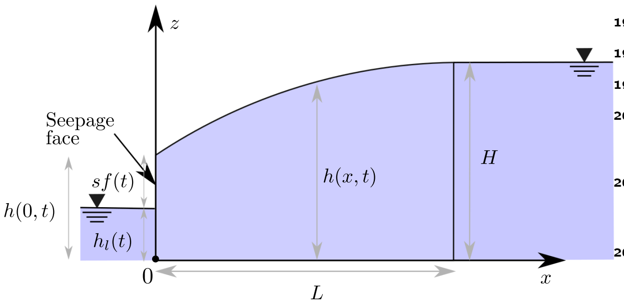



**Figure 1:** Schematic of the Di Nucci model showing constant head $H$ at $x = L$, transient seepage face height $sf(t)$ at $x = 0$, transient lake height $h_l(t)$, and transient free surface height $h(x,t)$. The heights are calculated from the impermeable base at $z = 0$. The domain extends from $x = 0$ to $x = L$, and the seepage face height is given as $sf(t) = h(0,t) - h_l(t)$. The same figure can be used for the Dupuit-Boussinesq model by changing two underlying assumptions. First, the Dupuit-Boussinesq model assumes the seepage face height $sf(t)$ to be zero. Second, the far-field head $h(x,L)$ is not specified.

Here, the seepage face is located at $x = 0$, and $t = \infty$ refers to the variable value at the steady-state, $q$ is the discharge per unit width in the third dimension (m$^2$/s). Note that in the original model, $h(0,\infty)$ is strictly the steady hydraulic head level where the aquifer discharges $h_l(\infty)$ but here we also consider the seepage face height $sf(\infty)$ as it is necessary to accurately predict $h(x,\infty)$ experimental values. However, the seepage face height is typically not known a-priori. Integrating (2) twice and using the boundary conditions yields the Dupuit-Forchheimer discharge formula (4) (Hantush, 1962; Kirkham, 1967; Hesse and Woods, 2010; Bear, 1972).

$$h(x,\infty) = \sqrt{h(0,\infty)^2 + \frac{2qx}{K}}, \quad x \in [0, L]. \tag{4}$$

The inherent difficulty of this method lies in the need for a boundary condition that is at the seepage face, $x = 0$, whose height is a combination of the known water level in the reservoir, $h_l$, and the unknown height of the seepage face (Figure 1). This problem is commonly neglected and the groundwater table is set equal to the surface water table where the water debouches.

### 2.2. Di Nucci model

The model derived by Di Nucci (2018) couples a Dupuit-Boussinesq type equation with Darcy's law and solves a one-dimensional PDE resulting from the two-dimensional unsteady free surface flow in a homogeneous, isotropic medium, as shown by the schematic diagram in Figure 1. The vertical flow is included by considering a higher-order, implicit term in the flux formulation. This term is given in Equation (5), as well as the first-order term associated with Darcy's law. A unique solution is possible using a boundary condition with time dependent flux at the seepage face, $x = 0$, given by Equation (7) and a constant hydraulic head level at the upstream boundary, $x = L$, given by Equation (8). Moreover, the seepage face development is accounted for by a mass and momentum balance as well as Cauchy's integral relation theorem for potential and stream function relationships (Bear, 1972; Di Nucci, 2018). The resulting governing equations take the form:

$$\frac{q}{K} = -\frac{\partial}{\partial x}\left[\frac{h^2}{2} - \frac{1}{K}\frac{\partial}{\partial x}\left(\frac{q}{h}\right)\frac{h^3}{3}\right], \tag{5}$$

$$\frac{1}{K}\frac{\partial q}{\partial x} = -\frac{\phi}{K}\frac{\partial h}{\partial t}, \quad t \in (0,\infty), \quad x \in (0, L), \tag{6}$$

subject to boundary conditions:

$$\frac{q}{K}(0,t) = g(t), \tag{7}$$

$$h(L,t) = H = \text{constant}, \tag{8}$$

where $q(x,t)$ is again the discharge per unit width (m$^2$/s). Moreover, $g(t)$ is considered a function of time to reproduce the boundary condition of the 2D problem, which can be considered as

$$\frac{q}{K}(0,t) = \frac{H^2 - h_l^2(\infty)}{2L} \tag{9}$$

for a steady-state lake level of $h_l(\infty)$. The integral relation arising from Cauchy theorem is

$$\frac{1}{2}h_l^2(t) = \frac{1}{2}H^2 - \int_0^L \frac{1}{K}q(x,t)\mathrm{d}x, \tag{10}$$

where $h_l(t)$ is the time varying height of lake which is not considered in Dupuit-Boussinesq approximation. The transient seepage face height $sf(t)$ (in m) can then be calculated using

$$\begin{aligned} sf(t) &= h(0,t) - h_l(t) \\ &= h(0,t) - \sqrt{H^2 - 2\int_0^L \frac{1}{K}q(x,t)\mathrm{d}x}. \end{aligned} \tag{11}$$

For steady-state, Equation (5) and (10) take the form

$$\frac{q}{K} = -\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{h^2}{2} + \frac{q}{K}\frac{\mathrm{d}h}{\mathrm{d}x}\frac{h}{3}\right) \quad \text{and} \tag{12}$$

$$\frac{q}{K} = \frac{H^2 - h_l^2(\infty)}{2L}. \tag{13}$$

Also, $q(x,\infty) = q$ becomes a constant in both space and time, stemming from Equation (6). For the boundary conditions,

$$\frac{\mathrm{d}h}{\mathrm{d}x}(L,\infty) = 0 \quad \text{and} \quad h(L,\infty) = H, \tag{14}$$

the analytical result for free surface height $h(x,\infty)$ is

$$h(x,\infty) = \sqrt{H^2 - \frac{2q(L-x)}{K} + \frac{2}{3}\frac{q^2}{K^2}\left[1 - \exp\left(-\frac{3K(L-x)}{q}\right)\right]}. \tag{15}$$

Coupling (15) with (13) gives the steady-state seepage face height as

$$sf(\infty) = h(0,\infty) - h_l(\infty) = \frac{2q^2}{3K^2} \frac{\left(1 - \exp\left(-\frac{3KL}{q}\right)\right)}{h(0,\infty) + h_l(\infty)}. \quad (16)$$

As such the Di Nucci model determines the unknown steady height of the groundwater table, $h(0,\infty)$, at the seepage face.

## 3. Physics informed neural networks

### 3.1. Deep neural networks for function approximations

Deep neural networks have been extensively studied for the purpose of approximating arbitrary functions (Hornik, Stinchcombe and White, 1989). Dissanayake and Phan-Thien (1994) first utilized neural networks to forward solve PDEs by assembling the residual form of a given PDE and its boundary conditions as soft constraints for training the neural network model. We refer to Goodfellow, Bengio and Courville (2016) for a full exposition of neural networks and its training, and Lu, Meng, Mao and Karniadakis (2021a) for its application to the context of approximating solutions to PDEs. Here, we present the formulation for a standard, feed-forward neural network, such as that described in Lu et al. (2021a). A feed-forward neural network defines the mapping from an input $\mathbb{R}^{n_{\text{in}}}$ to output space $\mathbb{R}^{n_{\text{out}}}$ based on successive, nonlinear transformations through layers of neurons. We refer to the first layer as the input layer, the final layer as the output layer, and additional layers as hidden layers. Activation values are passed from one layer to the next via an activation function composed along with a linear transformation. The neural network mapping, $u_{NN}(x)$, given an input vector, $\mathbb{R}^{n_{\text{in}}}$, can be mathematically represented as

$$u_{NN}(x;\theta) := (v_{N-1} \circ v_{N-2} \circ ... \circ v_1)(x), \quad (17)$$

where $\circ$ denotes the composition of two functions (i.e. $(v_2 \circ v_1)(x) = v_2(v_1(x))$ and $v_i$ maps the $i^{\text{th}}$ layer to its following layer through

$$v_i(x) = \sigma_i(W_i x + b_i) \text{ for } i = 1, 2, ..., N. \quad (18)$$

In this representation, transformations between the layers are parameterized by weights $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and biases $b \in \mathbb{R}^{n_i}$, collectively written as $\theta = \{W_i, b_i\}_{i=1}^{N-1}$. Here $N$ is the total number of layers and $n_i$ is the width of the $i^{\text{th}}$ layer. The function $\sigma_i(\cdot)$ is the activation function for the $i^{\text{th}}$ layer, which is typically a nonlinear function applied element-wise to its input vector. The possible choices for the activation function are numerous and include common implementations such as the sigmoid, ReLu and softplus functions (Goodfellow et al., 2016; Lu et al., 2021a). The activation function, for the output layer, can be chosen based on the desired output of the neural network. Derivatives of the neural network output with respect to the inputs, weights, and biases, can be obtained using automatic differentiation (Rumelhart, Hinton and Williams, 1986; Baydin, Pearlmutter, Radul and Siskind, 2018).

Given a training dataset $\mathcal{S}_t = \{(x_i, u_i)\}_{i=1}^{N_t}$ consisting of $N_t$ inputs $x_i$ and outputs $u_i$, the neural network is trained by minimizing a loss function. This is commonly taken to be the mean squared error (MSE) between the neural network outputs and the training data. Thus, we can write

$$\theta^* = \arg\min_{\theta} \frac{1}{N_t} \sum_{i=1}^{N_t} (u_{NN}(x_i;\theta) - u_i)^2, \quad (19)$$

where $\theta^*$ represents the optimal weights and biases. The optimization problem within training the neural network is frequently solved using gradient based optimization algorithms such as stochastic gradient descent (Bottou, 2010), ADAM (Kingma and Ba, 2014), and limited-memory BFGS (L-BFGS) (Liu and Nocedal, 1989).

To avoid over-fitting, additional regularization terms may be included in the loss function such as $l_1$ or $l_2$ norms of the weights and biases (Goodfellow et al., 2016). For deep neural networks with a large number of neurons, a process known as dropout can also be employed during training as a form of regularization. This technique omits random weights and biases during training (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014).

### 3.2. PINNs for solving forward and inverse problems

#### *3.2.1. Learning forward solutions*

Physics informed neural networks (Raissi et al., 2019) aim to enforce physics based constraints on the neural network to improve the effectiveness of the technique when applied to physical systems (Tartakovsky et al., 2020). Supposing a physical system has state $u(x,t)$ which is governed by a nonlinear PDE of the form

$$u_t + \mathcal{N}(u;\lambda) = 0, \quad (20)$$

where $\mathcal{N}$ is a nonlinear differential operator and $\lambda$ consist of model parameters defining the PDE. Within the PINNs framework, the state $u(x,t)$ is approximated by a feedforward neural network $u_{NN}(x,t)$, as defined in (17). Information given by the PDE is incorporated into the training of the neural network by defining the loss function as

$$\mathcal{L}(\mathcal{S}_t, \mathcal{S}_c, \theta) = \text{MSE}_u + \alpha \text{MSE}_f, \quad (21)$$

where

$$\text{Data misfit, MSE}_u = \frac{1}{N_t} \sum_{i=1}^{N_t} (u_{NN}(x_i,t_i) - u_i)^2, \quad (22)$$

$$\text{PDE misfit, MSE}_f = \frac{1}{N_c} \sum_{i=1}^{N_c} |f(u_{NN}(x_i,t_i);\lambda)|^2. \quad (23)$$

Here, MSE is the mean-squared error loss term and is referred to as the misfit term in this paper. Moreover,

$f(u(x,t);\lambda) := u_t(x,t) + \mathcal{N}(u(x,t);\lambda)$ is the PDE residual, $N_t$ is the number of data points in the training set $\mathcal{S}_t = \{(x_i,t_i,u_i)\}_{i=1}^{N_t}$, $N_c$ is the number of collocation points of the form $\mathcal{S}_c = \{(x_j,t_j)\}_{j=1}^{N_c}$, and $\alpha$ is the PDE regularization parameter. The data misfit term, $\text{MSE}_u$, is evaluated on the training data points where the state is known. The PDE misfit term, $\text{MSE}_f$, is evaluated via automatic differentiation on $N_c$ collocation points $(x_i,t_i) \in \mathcal{S}_c$ where the state is not necessarily known. The $\text{MSE}_f$ adds physics information to the neural network by encouraging the satisfaction of the governing PDE on the collocation points. The parameter $\alpha$ can be chosen to balance the relative effects of data and PDE in training the neural network. Once trained, the optimal weights and biases are determined as $\theta^*$

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\mathcal{S}_t, \mathcal{S}_c, \theta) \tag{24}$$

and the resulting neural network $u_{NN}$ is used to predict the state at desired points $(x,t)$.

This PINNs formulation can be used as a solver for the PDE by supplying initial and boundary conditions as training data and then using points on the interior of the domain as collocation points for evaluation of the PDE misfit (Raissi et al., 2019). The neural network is then trained to fit the initial and boundary data while satisfying the PDE. Alternatively, initial and boundary conditions can be enforced as hard constraints by directly building them into the neural network approximation, $u_{NN}$, through an auxiliary function (Lagaris, Likas and Fotiadis, 1998; Lu, Pestourie, Yao, Wang, Verdugo and Johnson, 2021b), or through the use of constrained optimization algorithms such as the penalty and augmented Lagrangian methods (Basir and Senocak, 2022).

### 3.2.2. Learning parameterized forward-solutions

We also consider a parameterization involving an additional input variable, $\mu$. For example, $\mu$ can parameterize a range of source terms over which the neural network is to be predictive. In this case, we construct the neural network approximation, $u_{NN}(x,t,\mu)$, with the additional input variable, $\mu$. We train the neural network using training data $\mathcal{S}_t = \{(x_i,t_i,\mu_i,u_i)\}_{i=1}^{N_t}$ corresponding to different values of input variables. We adopt the same loss function as in (21) with

$$\text{Data misfit, } \text{MSE}_u = \frac{1}{N_t}\sum_{i=1}^{N_t}(u_{NN}(x_i,t_i,\mu_i) - u_i)^2, \tag{25}$$

$$\text{PDE misfit, } \text{MSE}_f = \frac{1}{N_t}\sum_{i=1}^{N_t}|f(u_{NN}(x_i,t_i,\mu_i),\mu_i;\lambda)|^2, \tag{26}$$

in which we use the training data points to evaluate both the data and PDE misfits. Again, we can optimize the weights and biases to obtain our neural network approximation.

In this approach, the neural network is essentially trained on data while using the PDE as a form of regularization. The resulting neural network predictions represent a fitting of training data that is also informed by the physics associated with the PDE and scaled with the weighing parameter $\alpha$. Crucially, the PDE used does not need to capture all of the physics. Instead, we can adopt this approach even when initial or boundary conditions are not specified because the PDE is only used as regularization and does not need to be solved in training.

### 3.2.3. Inverting for model parameters

When model parameters $\lambda$ are unknown, they can be inverted for, during training, by defining them as additional optimization variables along with the weights and biases $\theta$. The optimization problem then takes the form

$$(\theta^*, \lambda^*) = \arg\min_{\theta,\lambda} \mathcal{L}(\mathcal{S}_t, \mathcal{S}_c, \theta, \lambda). \tag{27}$$

It must be noted that in either case, (24 or 27), the PDE does not need to be satisfied exactly by the trained neural network. Instead, the PDE misfit is only minimized to the extent achievable by the training process. Therefore, the recovered parameter values have a meaningful physical interpretation only when the PDE is well satisfied by the neural network. Otherwise, the recovered parameters serve only to improve predictions made by the neural network. Recent improvements aim to address this issue. For example, Basir and Senocak (2022) ensures that the PDE misfit vanishes in training through the use of the augmented Lagrangian method.

## 4. PINNs for examining steady unconfined groundwater flows

We apply PINNs in the context of steady groundwater seepage in homogeneous porous media. Physics information is incorporated into the training of the PINNs through PDE models of quasi-1D seepage flow. In particular, we consider both the Dupuit-Boussinesq equation and Di Nucci's equation as potential models.

### 4.1. PDE models

Under steady-state conditions, the Dupuit approximation given by Equation (2) can be integrated with the flow boundary condition (3) to yield

$$q + Kh\frac{\mathrm{d}h}{\mathrm{d}x} = 0, \quad x \in (0, L), \tag{28}$$

and similarly integrating the Di Nucci's model ODE with flow boundary condition takes the form previously derived in Equation (12) as

$$q + Kh\frac{\mathrm{d}h}{\mathrm{d}x} + \frac{q}{3}\frac{\mathrm{d}}{\mathrm{d}x}\left(h\frac{\mathrm{d}h}{\mathrm{d}x}\right) = 0, \quad x \in (0, L). \tag{29}$$

In both equations, $q$, the flow rate per unit width, is constant in space due to the absence of recharge, and parametrizes the flow profile $h(x)$. For the purpose of training, we normalize the two equations by this non-zero constant such that the source term is of $\mathcal{O}(1)$. In this case, the residual of the Dupuit equation can be re-written as

$$f_{\text{Dupuit}}(h, q; K) := 1 + \frac{K}{q} h \frac{\mathrm{d}h}{\mathrm{d}x} = 0, \quad x \in (0, L) \tag{30}$$

and the residual of the Di Nucci equation becomes

$$f_{\text{DiNucci}}(h, q; K) := 1 + \frac{K}{q} h \frac{\mathrm{d}h}{\mathrm{d}x} + \frac{1}{3} \frac{\mathrm{d}}{\mathrm{d}x}\left(h \frac{\mathrm{d}h}{\mathrm{d}x}\right) = 0, x \in (0, L). \tag{31}$$

## 4.2. Learning flow-parametrized solutions to seepage equations

Our goal is to be able to predict the phreatic surface profiles parameterized by the flow rate per unit width $q$. To accomplish this, we seek a neural network approximation, $h_{NN}(x, q)$, which takes the longitudinal position, $x$, and the flow rate per unit width, $q$, as input variables. Training data is given in terms of the free surface height, $h_i$, corresponding to the inputs $(x_i, q_i)$. Furthermore, we incorporate the physics information provided through either the Dupuit or Di Nucci equations under the PINN framework. This formulation is the steady-state and therefore, the time component can be neglected.

Instead of directly approximating $h(x, q)$, we construct the neural network approximation by scaling the input and output variables by their maximal values within the training data, $x_{\max}$, $q_{\max}$, and $h_{\max}$. That is, we define the scaled inputs and outputs,

$$\tilde{x} = \frac{x}{x_{\max}}, \quad \tilde{q} = \frac{q}{q_{\max}}, \quad \tilde{h} = \frac{h}{h_{\max}}, \tag{32}$$

and construct a neural network $\tilde{h}_{NN}(\tilde{x}, \tilde{q})$ that takes the scaled position and flow variables as inputs, and outputs the scaled free surface height. We can recover the approximation for free surface height by

$$h_{NN}(x, q) = \tilde{h}_{NN}\left(\frac{x}{x_{\max}}, \frac{q}{q_{\max}}\right) h_{\max}. \tag{33}$$

Scaling of the variables helps to ensure that the input variables $\tilde{x}$ and $\tilde{q}$ are of similar magnitudes, which can help to accelerate training of the neural network (Priddy and Keller, 2005). Furthermore, scaling the output variable also simplifies the interpretation of the regularization parameter, which will be discussed later.

In addition to the flow rate, the hydraulic conductivity $K$ enters as a model parameter, which is treated as a constant throughout the domain. Thus, in the steady-state case, we have PDEs of the form

$$f(h(x), q; K) = 0, \quad x \in \Omega, \tag{34}$$

using either $f = f_{\text{Dupuit}}$ or $f = f_{\text{DiNucci}}$. This allows us to define the training loss as

$$\mathcal{L}(\mathcal{S}_t, \theta, K) = \frac{1}{N_t} \sum_{i=1}^{N_t} (\tilde{h}_{NN}(\tilde{x}_i, \tilde{q}_i; \theta) - \tilde{h}_i)^2 + \frac{\alpha}{N_t} \sum_{i=1}^{N_t} |f(h_{NN}(x_i, q_i), q_i; K)|^2 \tag{35}$$

given training data $\mathcal{S}_t = \{(x_i, q_i, h_i)\}_{i=1}^{N_t}$, with $\tilde{x}_i$, $\tilde{q}_i$, and $\tilde{h}_i$ denoting their scaled values. Note that we evaluate the PDE misfit using re-dimensionalized variables on the same locations as the training data, as in Equations (25) and (26), and the corresponding derivatives of $h_{NN}$ are computed by a simple change of variable based on Equation (33).

Typically, boundary conditions are also required to solve for the complete flow profile using the PDEs. However, it is difficult to determine appropriate boundary conditions for both the Dupuit-Boussinesq and Di Nucci equations. As previously discussed, when a seepage face is present, the piezometric head where water debouches the media differs from the surface water height at that point. This piezometric head is unknown a priori. However, the PINNs formulation does not require imposing a boundary condition. Instead, the PDE is used as regularization for the flow profile in the interior of the domain and the data helps to inform the neural network about the boundary. Therefore, we will not explicitly employ a boundary misfit term in the loss function.

When accurate estimates for hydraulic conductivity, $K$, are not available, we can invert for the value of $K$ during training based on the training data. To do so, we consider $K$ as a variable that may be optimized in training, which is updated based on the loss function (35). Due to the uncertainties associated with the experimentally measured $K$, inverting for $K$ in training can produce a model that better fits the training data.

## 4.3. PINNs implementation

This study's investigations are performed with fully connected, feed-forward neural networks. Figure 2 shows the architecture diagrams of the PINNs based on the Di Nucci model. The default architecture utilized involves 4 hidden network layers that are each 20 neurons wide. The hyperbolic-tangent activation function is used for all hidden layers, while a softplus activation function is used for the output layer to ensure the predicted free surface heights are non-negative. The output of the neural network $h_{NN}$ is automatically differentiated with respect to $x$ in order to compute the PDE misfit term.

The loss function (35) is then minimized to predict the optimal weights and biases $\theta^*$ (24), and model parameters $\lambda^*$ (hydraulic conductivity $K$) (27). Since hydraulic conductivity can vary by orders of magnitudes and cannot be negative, we invert for the log of hydraulic conductivity $K$. We employ a combination of the ADAM and L-BFGS optimization algorithms to train the neural networks. In all training cases, we perform 50,000 ADAM iterations followed by L-BFGS until convergence to a tolerance of $\epsilon = 10^{-8}$ on the norm of the gradient of the loss function.

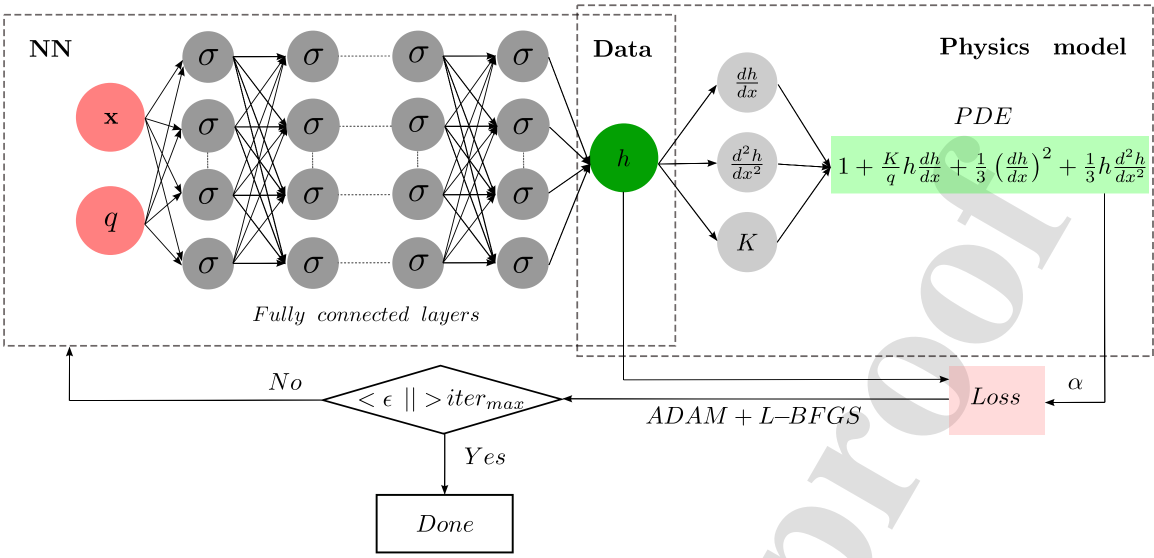



**Figure 2:** Neural network architecture diagrams of PINN for investigating steady-state groundwater flows using Di Nucci model.

### 4.4. Selection of a regularization parameter for the PDE misfit term

There exist sophisticated, adaptive regularization schemes such as learning rate annealing, neural tangent kernels and minimax weighting (McClenny and Braga-Neto, 2020; Wang, Teng and Perdikaris, 2021), predominantly used to improve the forward solution of PDE systems. However, in this work, we choose non-adaptive regularization in order to reduce the complexity of the loss term while getting sufficiently accurate predictions. A scaling analysis of the competing terms in the loss function aids in selecting the PDE misfit regularization parameter $\alpha$. Considering a trivial neural network $h_{NN} = 0$, we observe that the data misfit term is

$$\mathrm{MSE}_h = \frac{1}{N_t}\sum_{i=1}^{N_t}(\tilde{h}_i - \tilde{h}_{NN}(x_i, q_i))^2 \sim \mathcal{O}(1)$$

due to the scaling of the output variable. The PDE misfit term is

$$\mathrm{MSE}_f = \frac{1}{N_t}\sum_{i=1}^{N_t} f(h_{NN}(x_i, q_i), q_i; K)^2 \sim \mathcal{O}(1)$$

due to our choice of normalization for the PDE. With a choice of $\alpha = \mathcal{O}(1)$, we expect the significance of the data misfit to be comparable to the PDE misfit. In this paper, we select the regularization parameter as a fixed hyperparameter to minimize the testing errors of the neural networks. Moreover, we conduct a comprehensive investigation of testing errors with different hyperparameters, training data and PDE models, with $\alpha = 1$ as a reference point.

## 5. Data generation

Synthetic data is generated using the analytical solutions of the two PDEs; (4) for the Dupuit-Boussinesq and (15) for the Di Nucci models respectively. The analytical results $h(x)$ at selected values of $(x_i, q_i)$ are then corrupted by Gaussian white noise with standard deviation that is 2% of the maximum $h(x)$ in the dataset. Synthetic data is used to test the performance of the neural networks as both the model and its parameters are known.

We also perform our analysis on experimental data of steady groundwater flow that was obtained using the experimental design shown in Figure 3. The setup consists of an acrylic cell of length 167 cm, height 45 cm and width 2.54 cm (in the third dimension) which contains a porous region filled with 1 or 2 mm diameter beads. Dyed water is pumped from the right boundary $x = L$ at a specified flow rate which subsequently drains from the seepage face on the left boundary $x = 0$ with zero head at the gravity well, i.e., $h_l = 0$. A camera, placed orthogonally in front of the cell, takes pictures which are then processed using a Matlab code to digitize and extract the free surface profiles.

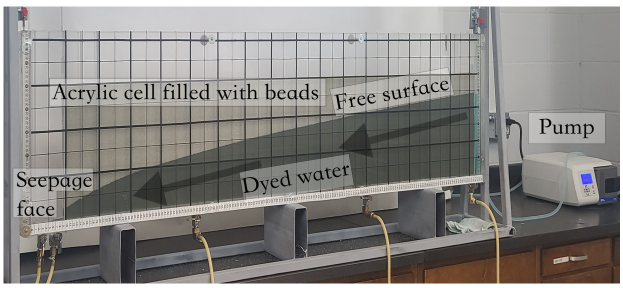



**Figure 3:** A picture of the experimental setup.

# 6. Steady-state results using synthetic data

## 6.1. Learning parametrized solutions from synthetic data

Synthetic data $(x_i, q_i, h_i)$ is generated from 8 linearly spaced flow values of $q \in [10^{-4}, 10^{-3}]$ m$^2$/s and 30 equidistant points of $x \in [0, 1.65]$ m with $K = 0.002$ m/s. These values are selected to resemble those found in the experimental data. PINNs are then trained using the synthetic data along with its corresponding PDE as regularization. The PDE misfit is evaluated using the same locations $(x_i, q_i)$ as the training data. We highlight that the seepage face height in the synthetic data using the conventional Dupuit-Boussinesq model is zero prior to adding noise, but is non-zero for Di Nucci model.

We first examine the effects of the regularization parameter $\alpha$. PINNs are trained with increasing values of $\alpha$ from $\alpha = 0$ up to $\alpha = 10^3$. Here $\alpha = 0$ corresponds to a plain neural network which does not incorporate any physics information. We consider both fixing the hydraulic conductivity $K$ and simultaneously inverting for $K$ in training. For the training values of $q$, we plot the noisy free surface data along with the predictions of the neural networks trained using each value of $\alpha$. Examples of the resulting free surface profiles are shown in Figures 4 (also see Supplementary Figure 1) and 6 (S.F. 3) for the Dupuit and Di Nucci equations using PINNs with fixed $K$ and in Figures 10 (S.F. 7) and 12 (S.F. 9) for the Dupuit and Di Nucci equations using PINNs with inverted $K$. All results are available in the supplementary file with corresponding figure references given in the parenthesis pertaining to each case. These profiles are plotted alongside the training data, as well as the true noise-free profiles from the underlying PDE solutions.

In general, we observe extreme overfitting for small values of $\alpha$ due to the lack of regularization, both with and without inversion for $K$. The overfitting is reduced by increasing $\alpha$, as the PDE is more strongly respected relative to the training data. This happens due to the introduction of the physics information to the neural network from the PDE misfit term in objective function. Increasing $\alpha$ increases the accuracy of when compared with the underlying noise-free PDE solution. In the case with fixed $K$, the range of $\alpha \in [10^{-2}, 10]$ lead to similar predictions of the profile. However, for $\alpha$ values beyond this (e.g. $\alpha = 10^3$), the predictions deviate from the data due to the excessive weighting on the PDE misfit. This is unhelpful in this case as the PDE regularization alone does not determine the flow profile due to the lack of boundary conditions. Instead, data is needed to provide information about the seepage face to constrain the solution. Similar observations can be made for the profiles arising from the PINNs with inverted $K$. This suggests that it is important to find an optimal regularization parameter $\alpha$.

This effect is illustrated further in the plots of PDE residuals inside the domain, corresponding to predicted free surface profiles. These are shown for Dupuit model in Figure 5 (S.F. 2) and Di Nucci model in Figure 7 (S.F. 4) for PINNs with fixed $K$, and Dupuit model in Figure 11 (S.F. 8) and Di Nucci model in Figure 13 (S.F. S10) for PINNs with inverted $K$. From these plots, it is evident that increasing $\alpha$ decreases the PDE residual. Close to the seepage face ($x \to 0$), the PDE residual typically increases. This likely a result of both noise in the data and rapid changes in free surface height near the seepage face, which are difficult to capture with sparse data points. The case of no PDE misfit, $\alpha = 0$, generally has the highest residual. The residual is less than $10^{-2}$ at most points in the domain for $\alpha \geq 0.1$. We note that the inverted spikes in the residuals are artifacts of the log scale, and correspond to points where the sign of the PDE residual changes.

To compare the generalization capabilities of the neural networks, we plot in Figure 8 the testing errors ($\mathrm{MSE}_h$), averaged across 10 different initializations of neural network weights and biases in training, as a function of the regularization parameter $\alpha$. The testing data are generated from randomly sampled flow values $q \in [10^{-4}, 10^{-3}]$ m$^2$/s that are not in the training set. The left figure corresponds to the predictions made by PINNs with fixed $K$ while the right figure corresponds to predictions made by PINNs with inverted $K$. Here, we observe that the optimal choice for the regularization parameter is around $\alpha = 0.1 - 1$ where the testing errors are below $2 \times 10^{-5}$. The testing errors of the neural networks trained on the Dupuit and Di Nucci models are close, indicating that the PINNs models perform similarly for data generated by the two different underlying models.

In addition to our default setup considered here, we repeat the analysis for testing error in terms of the regularization parameter, and use different values of noise ratio, hydraulic conductivity, amounts of training data, and neural network architectures. The results are provided in supplementary figures 5 and 6 for the testing errors using fixed and inverted $K$ respectively. The optimal choice of regularization parameter appears to be consistent across these variations, remaining relatively unchanged near $\alpha = 1$. The exceptions to this are the cases with very small noise, where small values of $\alpha$ can perform well, and very large network sizes, where a much larger value of $\alpha$ is required to prevent overfitting. Overall, scaling the data misfit and PDE misfit terms to the same magnitude allows an intuitive selection of an optimal $\alpha$ value (namely, $\alpha = 1$). Moreover, near-optimal $\alpha$ values, the corresponding optimal testing errors are similar in size across the majority of the neural network sizes considered. Thus, the optimal selection of $\alpha$ largely eliminates the need to tune additional hyperparameters, such as the width and depth of the neural network.

## 6.2. Inversion for hydraulic conductivity

From the synthetic data, we also invert for the hydraulic conductivity, $K$. This is done by including $K$ as an optimization variable during the training of the neural network. To avoid biasing the solution, we initialize $K$ to be three times that of the ground truth. The inversion results are summarized in Table 1 for the range of $\alpha \in [10^{-4}, 10^2]$, where we report the means and standard deviations of $K$

| | Mean (m/s) | Error (%) | Std. Dev. (m/s) |
|---|---|---|---|
| Truth | $2 \times 10^{-3}$ | – | |
| | Dupuit model | | |
| $\alpha = 10^{-4}$ | $1.42 \times 10^{-3}$ | 29.22 | $1.77 \times 10^{-4}$ |
| $\alpha = 10^{-2}$ | $1.99 \times 10^{-3}$ | 0.41 | $1.87 \times 10^{-5}$ |
| $\alpha = 10^{-1}$ | $2.00 \times 10^{-3}$ | 0.53 | $1.94 \times 10^{-5}$ |
| $\alpha = 1$ | $2.02 \times 10^{-3}$ | 1.05 | $1.90 \times 10^{-5}$ |
| $\alpha = 10$ | $2.03 \times 10^{-3}$ | 1.65 | $2.57 \times 10^{-5}$ |
| $\alpha = 10^{2}$ | $2.05 \times 10^{-3}$ | 2.62 | $3.01 \times 10^{-5}$ |
| $\alpha = 10^{3}$ | $3.51 \times 10^{-3}$ | 75.5 | $2.04 \times 10^{-3}$ |
| | Di Nucci model | | |
| $\alpha = 10^{-4}$ | $1.70 \times 10^{-3}$ | 14.83 | $1.07 \times 10^{-4}$ |
| $\alpha = 10^{-2}$ | $2.00 \times 10^{-3}$ | 0.03 | $2.10 \times 10^{-5}$ |
| $\alpha = 10^{-1}$ | $2.01 \times 10^{-3}$ | 0.48 | $3.42 \times 10^{-5}$ |
| $\alpha = 1$ | $2.03 \times 10^{-3}$ | 1.71 | $1.60 \times 10^{-5}$ |
| $\alpha = 10$ | $2.05 \times 10^{-3}$ | 2.55 | $1.91 \times 10^{-5}$ |
| $\alpha = 10^{2}$ | $2.06 \times 10^{-3}$ | 3.19 | $2.02 \times 10^{-5}$ |
| $\alpha = 10^{3}$ | $4.07 \times 10^{-3}$ | 103.38 | $2.01 \times 10^{-3}$ |

**Table 1**
Recovered hydraulic conductivity, $K$, values from synthetic training data generated from PINNs with Di Nucci equation (top) and Dupuit equation (bottom). Inversion results are mean and standard deviations (std. dev.) across 10 different sets of initial neural network parameters during training.

across 10 initial weights and biases in the training of the neural networks.

In general, the inversion yields accurate values of $K$ for both the Di Nucci and Dupuit equation based PINNs. Of the tabulated results, with the exception of $\alpha = 10^{-4}$, the errors of the inverted $K$ values are on the order of 1%, and can likely be attributed to noise in the data. This gives us confidence that for the range of $\alpha \in [10^{-2}, 10]$ that gives optimal testing error, we can recover accurate estimates of $K$ while simultaneously training to predict the free surface profiles. Moreover, the PDE misfits are small $< 1 \times 10^{-2}$ throughout the domain for $\alpha > 0.1$, as shown in Figures 11 (S.F. S8) and 13 (S.F. S10) for the Dupuit and Di Nucci models, respectively. This allows us to meaningfully interpret the inverted hydraulic conductivity as a parameter of the underlying PDE model.

# 7. Steady-state results using experimental data

We next train neural networks on the experimental data, considering data from 1 mm and 2 mm beads separately. In the 1 mm data set, we have flow profiles for 10 different flow rates while for the 2 mm data set we have 12 different flow rates. Each flow profile consists of 200 data points. For simplicity, the height of the tail water, $h_l$, is set to zero for all experiments, but this model can be easily applied to non-zero tail water level. For each bead size, we take flow profiles from six of the flow rates as training data, and use the remaining datasets as test sets. We train PINNs using the Di Nucci and Dupuit-Boussinesq equations, using $\alpha = 1$ unless otherwise specified, and evaluate the PDE misfit using the same locations $(x_i, q_i)$ as the training data. Theoretical estimates of the hydraulic conductivity are pre-calculated using Cozeny-Karman relation for permeability (Bear, 1972). We consider both using the fixed theoretical estimates of $K$ as well as inverting for $K$ during training of the PINNs. We also train plain neural networks without physics informed regularization as a reference.

## 7.1. Flow data prediction

Examples of predictions by the trained neural networks, with and without physics informed regularization, are shown in Figures 14 (also see S.F. 11, 12) and 16 (S.F. 15, 16) for 1 mm and 2 mm beads respectively. The plots show the best and the worst cases among all the flow rates considered. The plots for all other cases are provided in the supplementary file. The mean squared error losses for both data ($\mathrm{MSE}_h$) and PDE ($\mathrm{MSE}_f$) while training are at least three orders of magnitudes less than the original scales (equal to unity) indicating that the NN have converged (see Table 2).

The plain neural networks are able to fit the training data, but tend to perform poorly in testing due to over-fitting. This is particularly noticeable for the 2 mm bead data, where the plain neural network suffers from spurious oscillations (see 14 and 16). This transcends to the PDE and data misfits in training where the PDE misfit for plain NN is at least one order of magnitude higher than that from PINNs (see Table 2). We also observe that the PINNs trained on Dupuit and Di Nucci models yield similar predictions to each other, and are almost indistinguishable from each other when both use the inverted $K$ values. However, the PINNs predictions using both the Dupuit and Di Nucci models with the fixed theoretical $K$ values tend to differ from the testing data near the boundaries. In particular, they over-predict the seepage face height for 1 mm beads and significantly underpredict the seepage face height for the 2 mm bead data. The deviations suggest that the theoretical estimates of $K$ may be inaccurate. Instead, PINNs with inverted $K$ values, yield the best results among all the techniques used. Please note that the standard Dupuit-Boussinesq model would have estimated a zero seepage height, but the information obtained from training data helps maintain a non-zero seepage height for the PINNs predictions.

The corresponding PDE residuals, across the domain, are shown in Figures 15 (also see S.F. 13, 14) and 17 (S.F. 17, 18) for Dupuit and Di Nucci cases. Unsurprisingly, the PDE residual for the plain neural network is the highest, close to 1 in regions of the domain, even for the training regimes. In contrast, the PDE residuals are below 0.05 for the training flow rates and below 0.5 in testing. PINNs that are trained using the Dupuit and Di Nucci models show similar PDE residuals. In particular, both cases show relatively small residuals ($< 0.001$) across the training data when using the inverted $K$ values. This suggests that the Dupuit equation describes the flow behavior sufficiently well within the domain, and that the higher-order terms in the Di Nucci

| Model | Type | $\alpha$ | Width | Depth | $\mathrm{MSE}_h$ | $\mathrm{MSE}_f$ |
|---|---|---|---|---|---|---|
| 1 mm bead size | | | | | | |
| N/A | Plain NN | 0 | 20 | 4 | $4.40 \times 10^{-4}$ | $9.91 \times 10^{-2}$ |
| Dupuit | Fixed K | 1 | 20 | 4 | $1.15 \times 10^{-3}$ | $3.66 \times 10^{-4}$ |
| Di Nucci | Fixed K | 1 | 20 | 4 | $1.15 \times 10^{-3}$ | $3.65 \times 10^{-4}$ |
| Dupuit | Inverted K | 1 | 20 | 4 | $5.66 \times 10^{-4}$ | $7.22 \times 10^{-6}$ |
| Di Nucci | Inverted K | 1 | 20 | 4 | $5.73 \times 10^{-4}$ | $8.20 \times 10^{-6}$ |
| 2 mm bead size | | | | | | |
| N/A | Plain NN | 0 | 20 | 4 | $1.51 \times 10^{-4}$ | $1.64 \times 10^{-1}$ |
| Dupuit | Fixed K | 1 | 20 | 4 | $1.84 \times 10^{-3}$ | $1.34 \times 10^{-4}$ |
| Di Nucci | Fixed K | 1 | 20 | 4 | $2.03 \times 10^{-3}$ | $1.83 \times 10^{-4}$ |
| Dupuit | Inverted K | 1 | 20 | 4 | $2.71 \times 10^{-4}$ | $7.83 \times 10^{-6}$ |
| Di Nucci | Inverted K | 1 | 20 | 4 | $2.72 \times 10^{-4}$ | $8.02 \times 10^{-6}$ |

**Table 2**
A summary of hyperparameters along with the training losses for the prediction (fixed $K$) and inversion tests performed on the experimental data corresponding to different bead sizes. MSE values correspond to neural networks presented in Figures 14 to 17

equation have only small effects in the experimental regimes considered.

To assess our selection of the regularization parameter, we again train PINNs using $\alpha \in [0, 10^3]$. We focus on the case of using the inverted $K$ values, as they are observed to provide more accurate predictions. We plot the resulting testing errors in Figure 9 as a function of the regularization parameter $\alpha$. These are averaged across 10 runs, each with different initial neural network weights and permutations of training and testing data sets. The figure shows predictions made by PINNs with fixed $K$ for 1 mm on the left and 2 mm beads on the right. Again, we see that the PINNs based on the two different models produce similar testing errors. We also observe that the optimal choice for the regularization parameter is around $\alpha = 1$, where the testing errors are below $2 \times 10^{-3}$ for 1 mm beads and $6 \times 10^{-4}$ for 2 mm beads. The optimal range of $\alpha$ is similar to that found in our study using synthetic data, and highlights the benefit of appropriately scaling the data and PDE model.

### 7.2. Inversion of Hydraulic Conductivity

We also present the inverted values of hydraulic conductivity, $K$, for both the 1 mm and 2 mm cases in Table 3. The table reports the mean and standard deviations of the inverted values across 10 different sets of initial weights, biases, and $K$ during neural network training. The recovered values of $K$ compare well with their corresponding theoretical estimates computed by the Cozeny-Karman relation. As we have noted, the corresponding PDE residuals are on the order of $10^{-3}$ for the training data, suggesting that the PDE model is well satisfied by the trained network. This allows us to interpret the recovered $K$ values as meaningful PDE parameters. The small deviations in the $K$ values are likely due to a discrepancy between the theoretical relationships and the heterogeneity in packing of the beads of the experimental setup.

| Model | Mean $K$ (m/s) | Std. Dev. $K$ (m/s) |
|---|---|---|
| 1 mm beads | | |
| Calculated | $9.10 \times 10^{-3}$ | – |
| Dupuit | $8.30 \times 10^{-3}$ | $3.25 \times 10^{-5}$ |
| Di Nucci | $8.30 \times 10^{-3}$ | $2.64 \times 10^{-5}$ |
| 2 mm beads | | |
| Calculated | $2.85 \times 10^{-2}$ | – |
| Dupuit | $3.48 \times 10^{-2}$ | $1.24 \times 10^{-4}$ |
| Di Nucci | $3.48 \times 10^{-2}$ | $1.09 \times 10^{-4}$ |

**Table 3**
Comparison of inverted and a-priori estimates of hydraulic conductivity, $K$, from experimental data for $\alpha = 1$. Inversion results are mean and standard deviations (std. dev.) across 10 different sets of initial neural network parameters during training.

## 8. Discussion

PINNs are able to improve upon the predictions given by solving the simplified PDE models alone. Admittedly, the Dupuit-Boussinesq approximation and Di Nucci equations do not fully represent the physics in the system. However, PINNs improve upon the predictions by supplementing incomplete PDE information with experimental training data, without resorting to high-fidelity, computationally expensive, multi-dimensional two-phase flow models. Furthermore, when considering the difficulty in prescribing appropriate boundary conditions for flows with seepage faces, we cannot make predictions directly using the PDEs. Even with the relatively simple Dupuit-Boussinesq equation that neglects vertical flow effects, seepage face development, and incomplete boundary specifications, we are able to use PINNs to make meaningful predictions regarding phreatic surface height and hydraulic conductivity values merely by using experimental data. By considering the information from the data, the seepage face height and even lake-level dynamics can be considered.

As a method of learning flow profiles from experimental data, we observe that PINNs tend to have greater generalization capabilities compared to a conventional neural network when trained on limited amounts of training data. The PDE based regularization makes it less sensitive to the noise in data and helps to prevent overfitting in a physics informed manner. Critical to this is the choice of regularization parameter $\alpha$. In our formulation, $\alpha = 1$ naturally represents a balance between the data and PDE misfit terms when the training data and PDE terms are normalized. This appears to yield optimal testing errors across many of our experiments, and highlights the benefits of scaling not only the input and output variables in the neural network, but also the PDE in physics informed machine learning. Furthermore, the PDE based regularization also reduces the burden of tuning other hyperparameters such as the neural network size.

Moreover, PINN is able to accurately recover hydraulic conductivity from the data. The deviations in inverted values of hydraulic conductivity versus the theoretical estimates can be due to many reasons; a combination of experimental error and the empirical nature of the theory. We believe the inverted values of $K$ are more accurate than those calculated. Also, this is a simple, novel way of estimating the hydraulic conductivity through in-situ measurements of free surface heights as opposed to lab-based permeameters tests. As an extension to this work, instead of constant permeability, a 2D permeability field $K(\mathbf{x})$ and boundary conditions can be inverted for separately or jointly.

## 9. Conclusions

In this paper, we have investigated steady groundwater flow using Physics Informed Neural Networks. The free surface profile data comes from analytical results of Dupuit-Boussinesq and Di Nucci models and moreover, laboratory experiments. PINNs make accurate predictions of the free surface profiles on both training and test data and are less sensitive to noise. The conventional neural network gives oscillatory and non-physical behavior on the same data set due to lack of physics information.

In our adopted framework, the regularization parameter for the PDE misfit plays the role in balancing the information from the data and the PDE model. The optimal value of PDE misfit regularization parameter, selected on the basis of minimizing generalization error, has been found close to unity which performs very well on both synthetic and experimental data. This value bolsters the importance of scaling the data and PDE misfit in order to balance the amount of information while training the PINNs. Note that when the PDE represents the physics completely, methods like augmented Lagrangian could be used to strongly enforce the boundary and initial conditions on the PDE loss (Basir and Senocak, 2022), eliminating the need for tuning the regularization parameter.

Note that when the PDE represents the physics completely, methods like augmented Lagrangian could be used to strongly enforce the PDE while eliminating the need for tuning the regularization parameter. Further, hydraulic conductivity has been inverted for the training data which gives fairly accurate predictions of free surface profiles and is close to the theoretical estimates. In the future, we plan to extend this PINNs model to study transient groundwater flow dynamics.

## Data availability

All related codes are available on Github: `https://github.com/dc-luo/seepagePINN` (Shadab, Luo, Shen, Hiatt and Hesse, 2021). Additionally, we have developed a simple toolbox that can be used to investigate steady groundwater flow dynamics. A manual is provided in the Github repository.

## CRediT authorship contribution statement

**Mohammad Afzal Shadab:** Conceptualization of this study, Methodology, Software, Data curation, Writing - Original draft preparation, Supervising. **Dingcheng Luo:** Conceptualization of this study, Methodology, Software, Data curation, Writing - Original draft preparation. **Eric Hiatt:** Experimentation, Data curation, Writing - Original draft preparation. **Yiran Shen:** Software, Data curation, Writing - Original draft preparation. **Marc Andre Hesse:** Conceptualization of this study, Writing - Editing, Supervising.

## References


Baird, A.J., Mason, T., Horn, D.P., 1998. Validation of a boussinesq model of beach ground water behaviour. Marine Geology 148, 55–69.

Bandai, T., Ghezzehei, T.A., 2020. Physics-informed neural networks with monotonicity constraints for richardson-richards equation: Estimation of constitutive relationships and soil water flux density from volumetric water content measurements. Water Resources Research , e2020WR027642.

Basir, S., Senocak, I., 2022. Physics and equality constrained artificial neural networks: application to forward and inverse problems with multi-fidelity data fusion. Journal of Computational Physics 463, 111301.

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

Bear, J., 1972. Dynamics of Fluids in Porous Media. Dover, New York.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Boussinesq, J., 1904. Recherches the èorique sur le èecoulement des nappes d'eau infiltre èes dans le sol et sur le de èbit des sources. J. Math. Pures Appl. 10, 5–75.

Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. Annual Review of Fluid Mechanics 52, 477–508.

Chen, Y., Lu, L., Karniadakis, G.E., Dal Negro, L., 2020. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. Optics express 28, 11618–11633.

Depina, I., Jain, S., Mar Valsson, S., Gotovac, H., 2021. Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow. Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards , 1–16.

Di Nucci, C., 2018. Unsteady free surface flow in porous media: One-dimensional model equations including vertical effects and seepage face. Comptes Rendus Mécanique 346, 366–383.

Dissanayake, M., Phan-Thien, N., 1994. Neural-network-based approximations for solving partial differential equations. communications in Numerical Methods in Engineering 10, 195–201.

Dupuit, J., 1863. Etudes théoriques et pratiques sur le mouvement des eaux dans les canaux découverts et àtravers les terrains perméable. 2nd ed., Dunod, Paris.

Forchheimer, P., 1901. Wasserbewegung durch Boden. Zeitschrift des Vereins Deutscher Ingenieure 45, 1782–1788.

van Genuchten, M., 1980. A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils1. Soil Science Society of America Journal 44, 892. doi:10.2136/sssaj1980.03615995004400050002x.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

Hantush, M.S., 1962. On the validity of the dupuit-forchheimer well-discharge formula. Journal of Geophysical Research 67, 2417–2420.

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.

van Herten, R.L., Chiribiri, A., Breeuwer, M., Veta, M., Scannell, C.M., 2020. Physics-informed neural networks for myocardial perfusion mri quantification. arXiv preprint arXiv:2011.12844 .

Hesse, M., Woods, A., 2010. Buoyant dispersal of CO2 during geological storage. Geophysical Research Letters 37, n/a–n/a. URL: http://doi.wiley.com/10.1029/2009GL041128, doi:10.1029/2009GL041128.

Hiatt, E., Shadab, M.A., Hesse, M.A., Gulick, S.P., 2021. An experimental and numerical investigation of seepage face dynamics, in: 2021 AGU Fall Meeting, (H35R-1240).

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural networks 2, 359–366.

Jagtap, A.D., Mitsotakis, D., Karniadakis, G.E., 2022. Deep learning of inverse water waves problems using multi-fidelity data: Application to serre–green–naghdi equations. Ocean Engineering 248, 110775.

Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. Journal of Computational Physics 426, 109951.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kirkham, D., 1967. Explanation of paradoxes in dupuit-forchheimer seepage theory. Water Resources Research 3, 609–622.

Lagaris, I.E., Likas, A., Fotiadis, D.I., 1998. Artificial neural networks for solving ordinary and partial differential equations. IEEE transactions on neural networks 9, 987–1000.

Liu, D.C., Nocedal, J., 1989. On the limited memory bfgs method for large scale optimization. Mathematical programming 45, 503–528.

Lu, L., Meng, X., Mao, Z., Karniadakis, G.E., 2021a. Deepxde: A deep learning library for solving differential equations. SIAM Review 63, 208–228.

Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., Johnson, S.G., 2021b. Physics-informed neural networks with hard constraints for inverse design. SIAM Journal on Scientific Computing 43, B1105–B1132. URL: https://doi.org/10.1137/21M1397908, doi:10.1137/21M1397908, arXiv:https://doi.org/10.1137/21M1397908.

Ma, L., Huang, C., Liu, Z.S., Morin, K.A., Aziz, M., Meints, C., 2020. Artificial neural network for prediction of full-scale seepage flow rate at the equity silver mine. Water, Air, & Soil Pollution 231, 1–15.

McClenny, L., Braga-Neto, U., 2020. Self-adaptive physics-informed neural networks using a soft attention mechanism. arXiv preprint arXiv:2009.04544 .

Meng, X., Karniadakis, G.E., 2020. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. Journal of Computational Physics 401, 109020.

Nourani, V., Babakhani, A., 2013. Integration of artificial neural networks with radial basis function interpolation in earthfill dam seepage modeling. Journal of Computing in Civil Engineering 27, 183–195.

Priddy, K.L., Keller, P.E., 2005. Artificial neural networks: an introduction. volume 68. SPIE press.

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

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

Rehamnia, I., Benlaoukli, B., Jamei, M., Karbasi, M., Malik, A., 2021. Simulation of seepage flow through embankment dam by using a novel extended kalman filter based neural network paradigm: Case study of fontaine gazelles dam, algeria. Measurement 176, 109219.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. nature 323, 533–536.

Rushton, K.R., Youngs, E.G., 2010. Drainage of recharge to symmetrically located downstream boundaries with special reference to seepage faces. Journal of Hydrology 380, 94–103. URL: http://dx.doi.org/10.1016/j.jhydrol.2009.10.026, doi:10.1016/j.jhydrol.2009.10.026.

Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D.E., Kuhl, E., 2020. Physics-informed neural networks for cardiac activation mapping. Frontiers in Physics 8, 42.

Scudeler, C., Paniconi, C., Pasetto, D., Putti, M., 2017. Examination of the seepage face boundary condition in subsurface and coupled surface/subsurface hydrological models. Water Resources Research 53, 1799–1819.

Shadab, M.A., Luo, D., Shen, Y., Hiatt, E., Hesse, M.A., 2021. Pinns for unconfined groundwater flow. URL: https://doi.org/10.5281/zenodo.5803542, doi:10.5281/zenodo.5803542.

Shukla, K., Di Leoni, P.C., Blackshire, J., Sparkman, D., Karniadakis, G.E., 2020. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. Journal of Nondestructive Evaluation 39, 1–20.

Shukla, K., Jagtap, A.D., Blackshire, J.L., Sparkman, D., Karniadakis, G.E., 2021. A physics-informed neural network for quantifying the microstructural properties of polycrystalline nickel using ultrasound data: A promising approach for solving inverse problems. IEEE Signal Processing Magazine 39, 68–77.

Simpson, M., Clement, T., Gallop, T., 2003. Laboratory and numerical investigation of flow and transport near a seepage-face boundary. Groundwater 41, 690–700.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

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

Tayfur, G., 2014. Soft computing in water resources engineering: Artificial neural networks, fuzzy logic and genetic algorithms. WIT Press.

Wang, S., Teng, Y., Perdikaris, P., 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing 43, A3055–A3081.

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 , 104243.

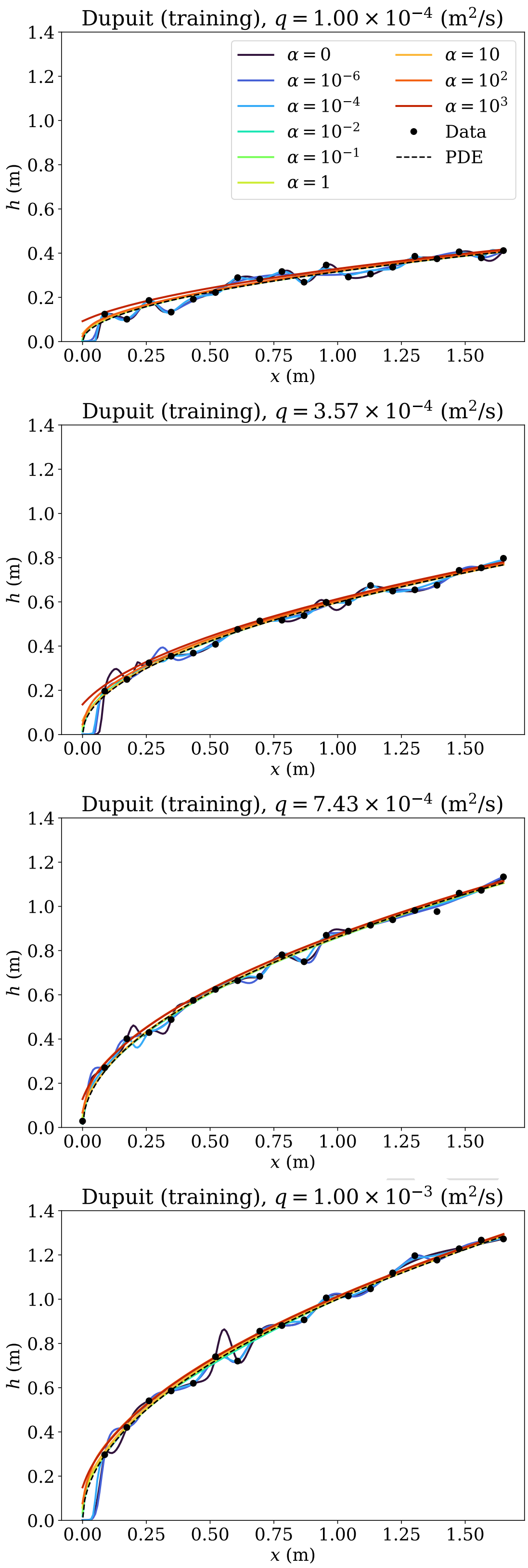



**Figure 4:** Neural network predictions of free surface profiles with varying $\alpha$, using the Dupuit equation as the regularizing PDE. The plots show the effect of changing the specific discharge $q = 10^{-4} - 10^{-3}$ m$^3$/m.s (shown in titles) and PDE regularization parameter $\alpha = 0 - 10^{3}$. Data and PDE refer to the noisy and noiseless data, respectively.

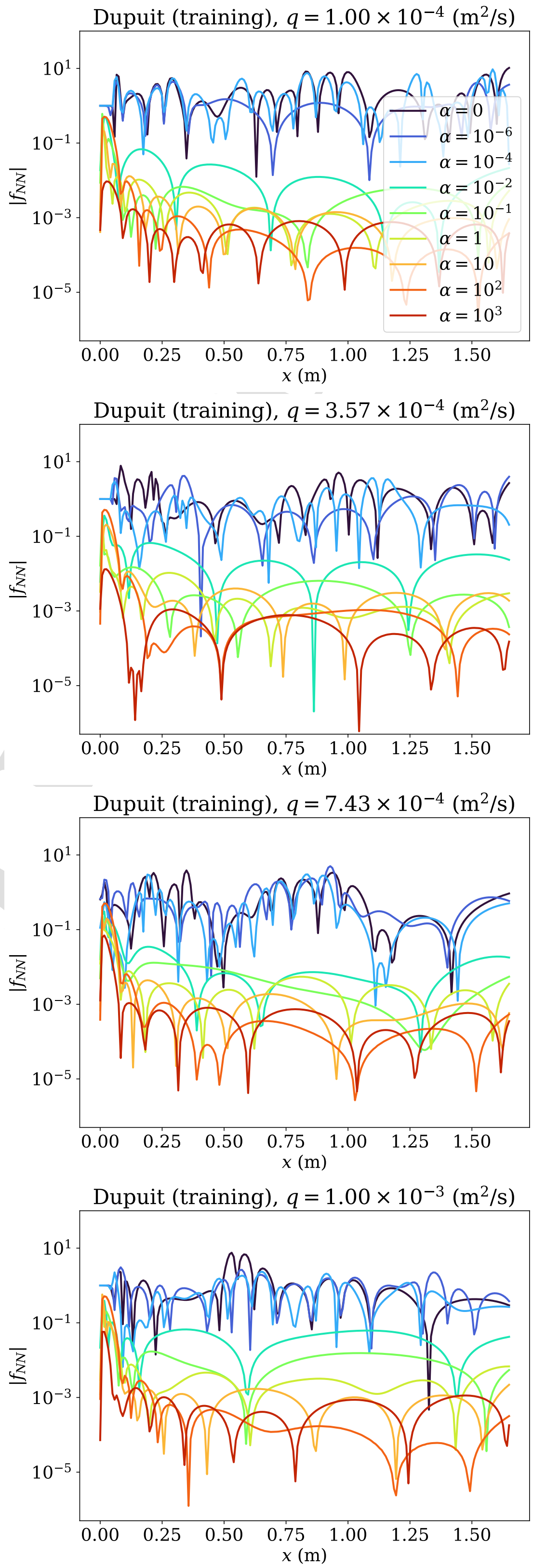



**Figure 5:** PDE residuals inside the domain corresponding to free surface profiles shown in Figure 4 for Dupuit model based PINNs predictions.

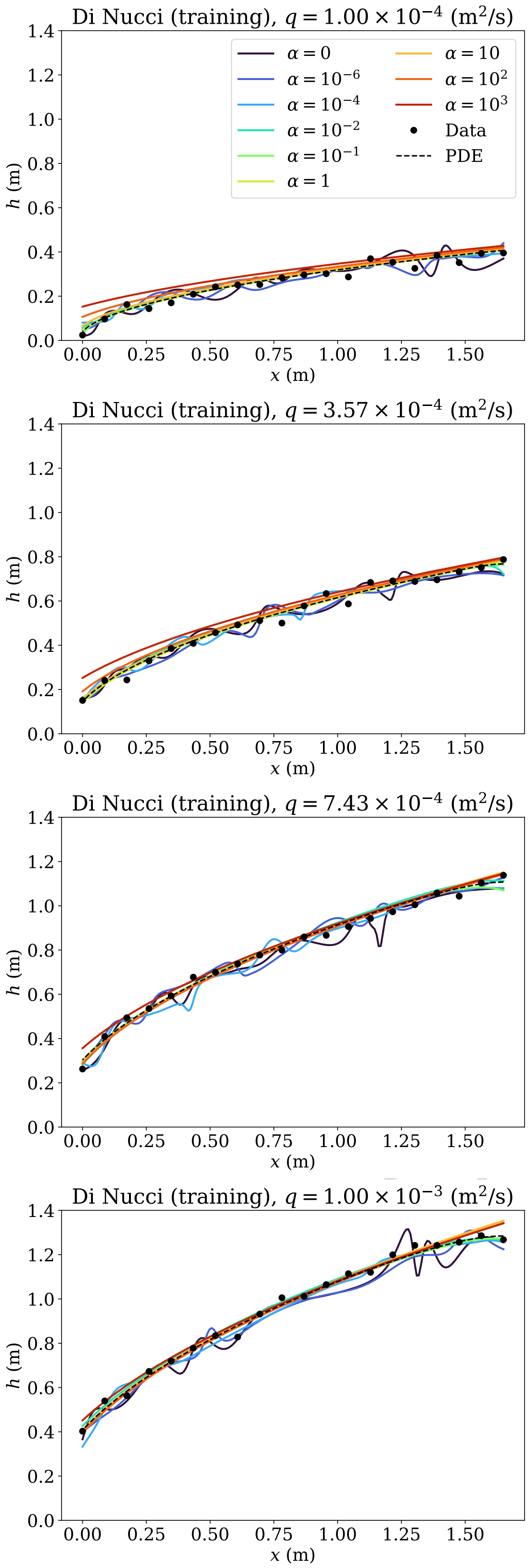



**Figure 6:** Neural network predictions with varying $\alpha$ and specific discharge $q$, using the Di Nucci equation as the regularizing PDE.

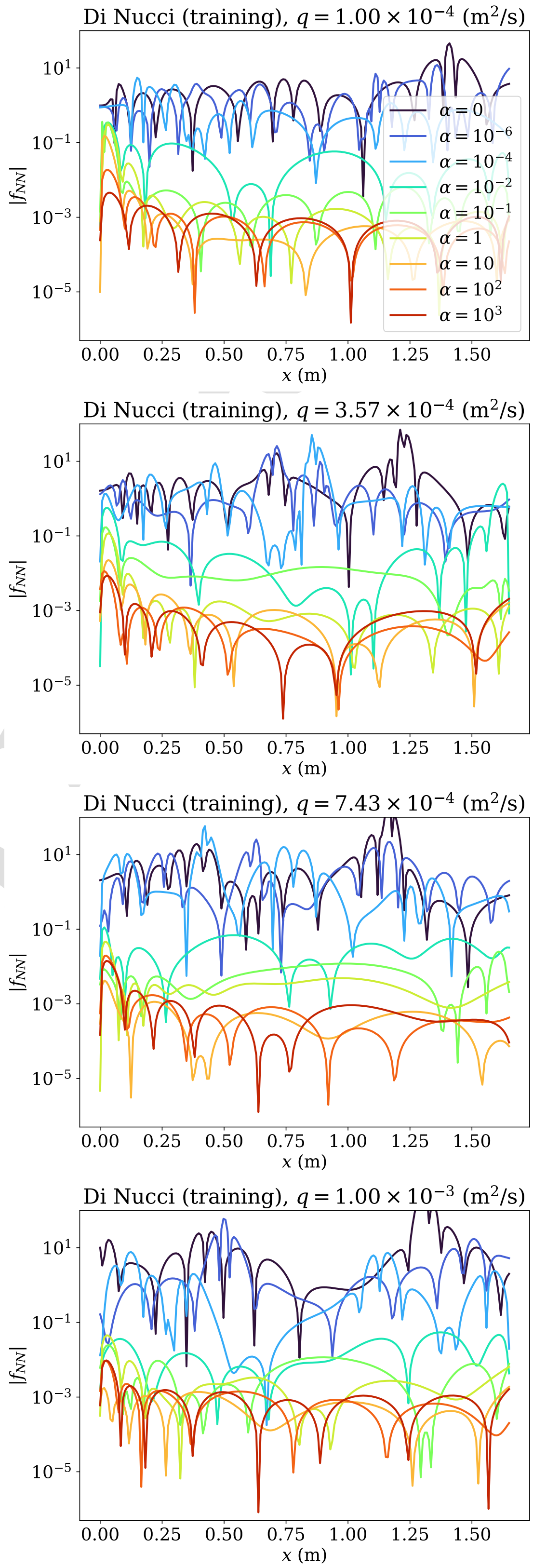



**Figure 7:** The PDE residuals inside the domain corresponding to free surface profile predictions, shown in Figure 6, using PINNs regularized by Di Nucci equation.

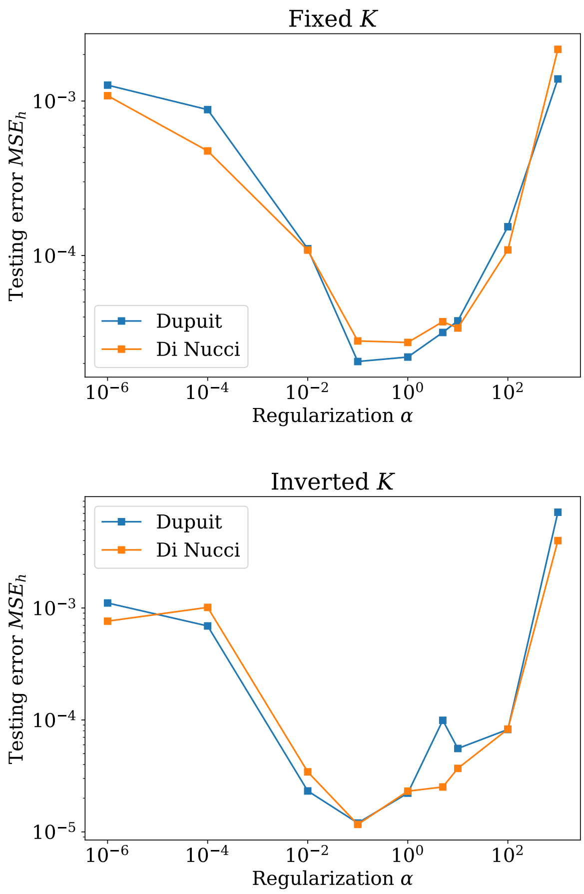



**Figure 8:** Average testing error across 10 runs, each with different initial neural network weights, as a function of the regularization parameter $\alpha$. The top figure corresponds to neural networks with fixed $K$, whereas the bottom figure corresponds to neural networks with inverted $K$. PINNs are trained on synthetic data.

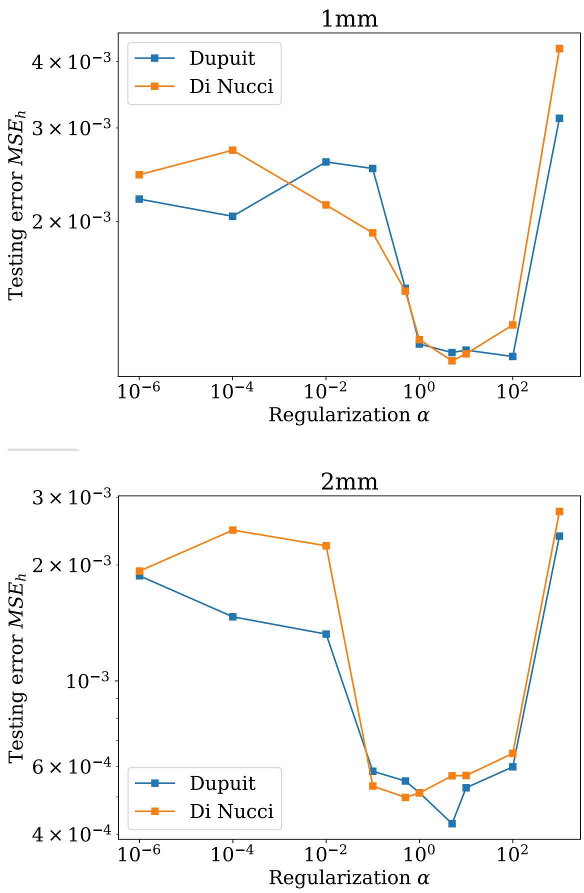



**Figure 9:** Average testing error of neural network predictions with inverted hydraulic conductivity $K$. Average is taken across 10 different training sets arrangements and initial weights as a function of the regularization parameter $\alpha$. The top figure is for the 1 mm bead data whereas the bottom figure is for the 2 mm bead data.

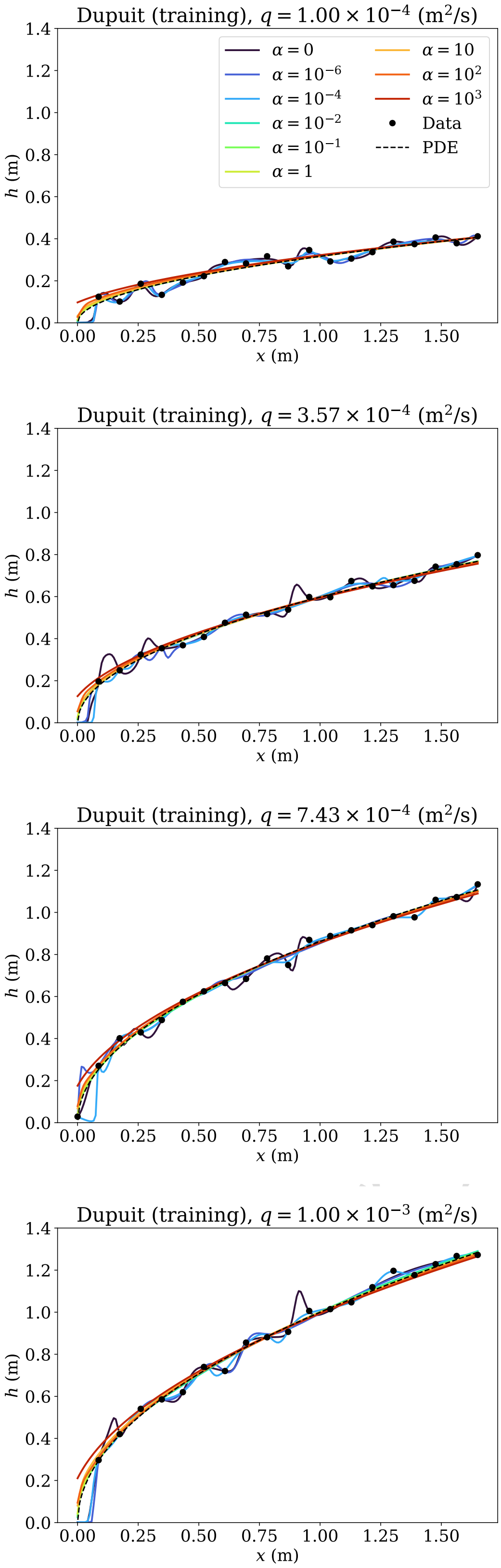



**Figure 10:** Training data and neural network predictions for free surface height while inverting for $K$, using the Dupuit equation as the regularizing PDE.

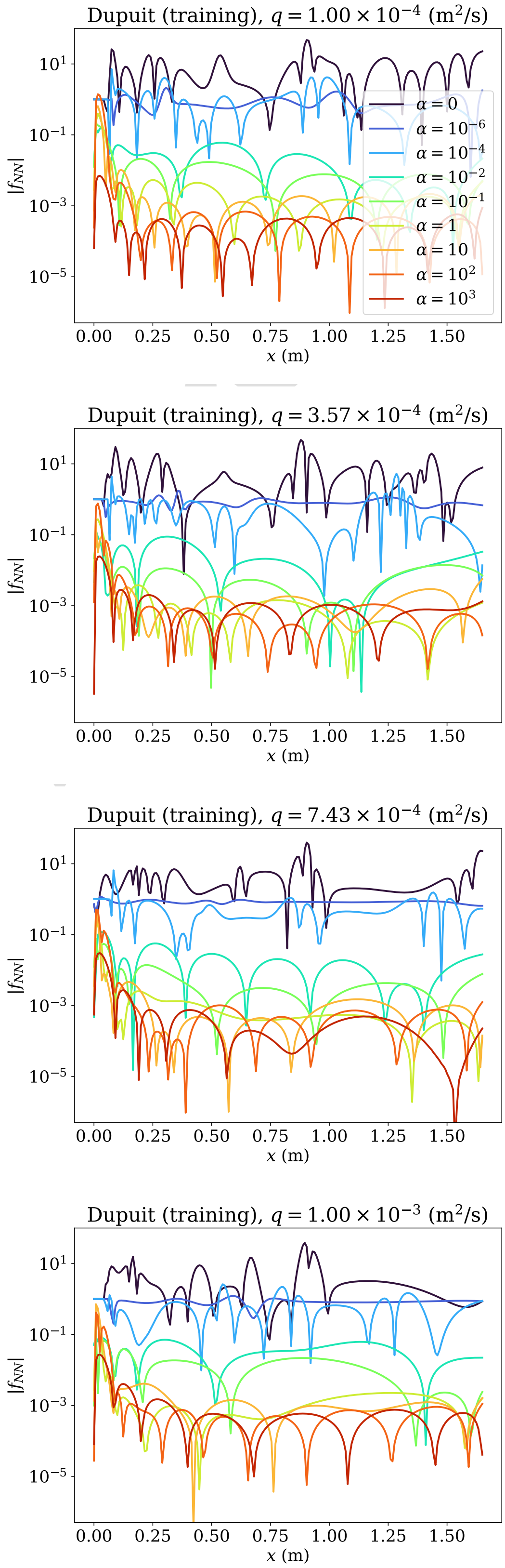



**Figure 11:** The PDE misfit terms inside the domain while inverting for $K$, corresponding to Figure 10, using the Dupuit equation as the regularizing PDE.

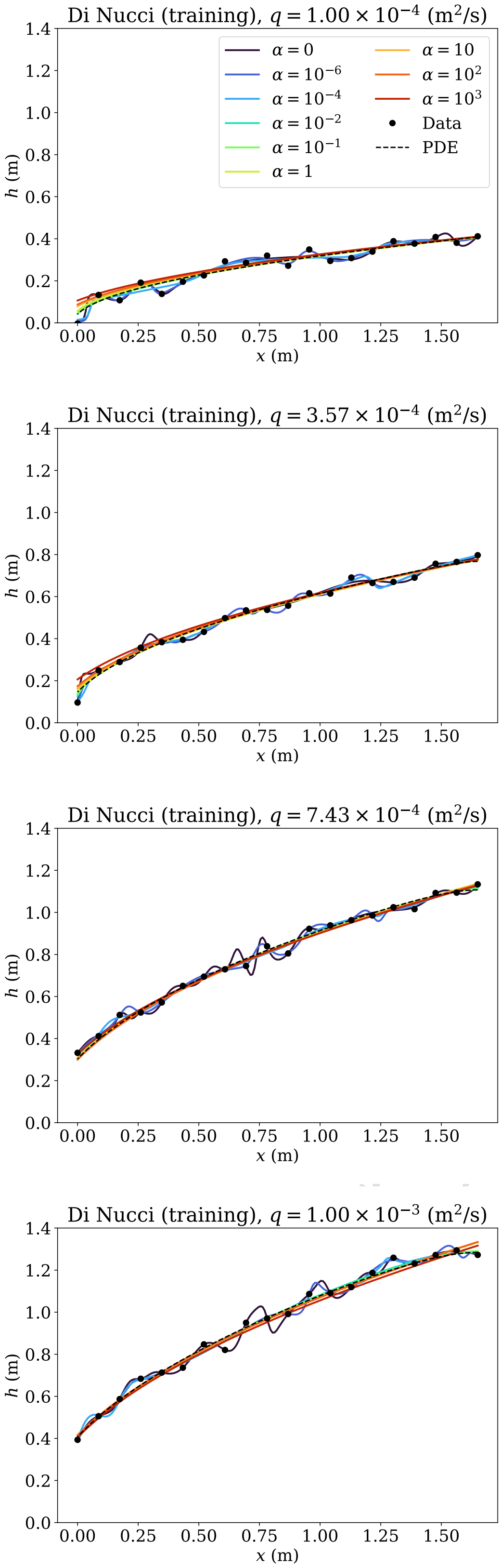



**Figure 12:** Training data and neural network predictions for free surface height while inverting for $K$, using the Di Nucci equation as the regularizing PDE.

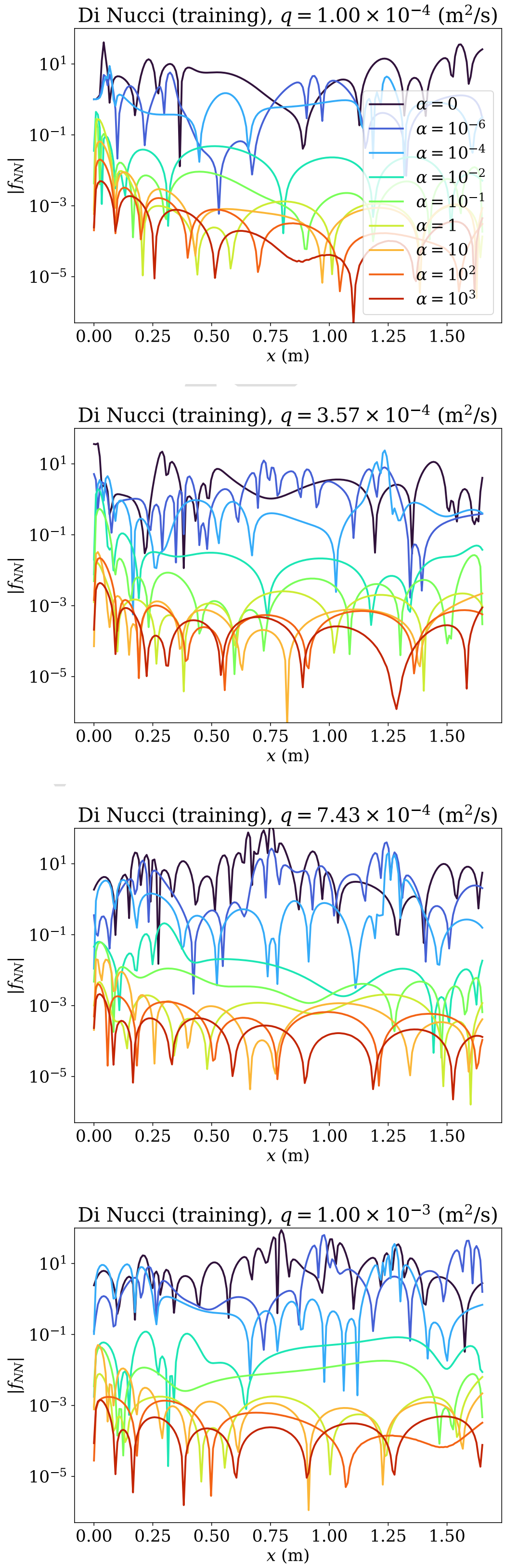



**Figure 13:** PDE residual for Di Nucci model based PINNs predictions corresponding to Figure 12 for different regularization parameters.

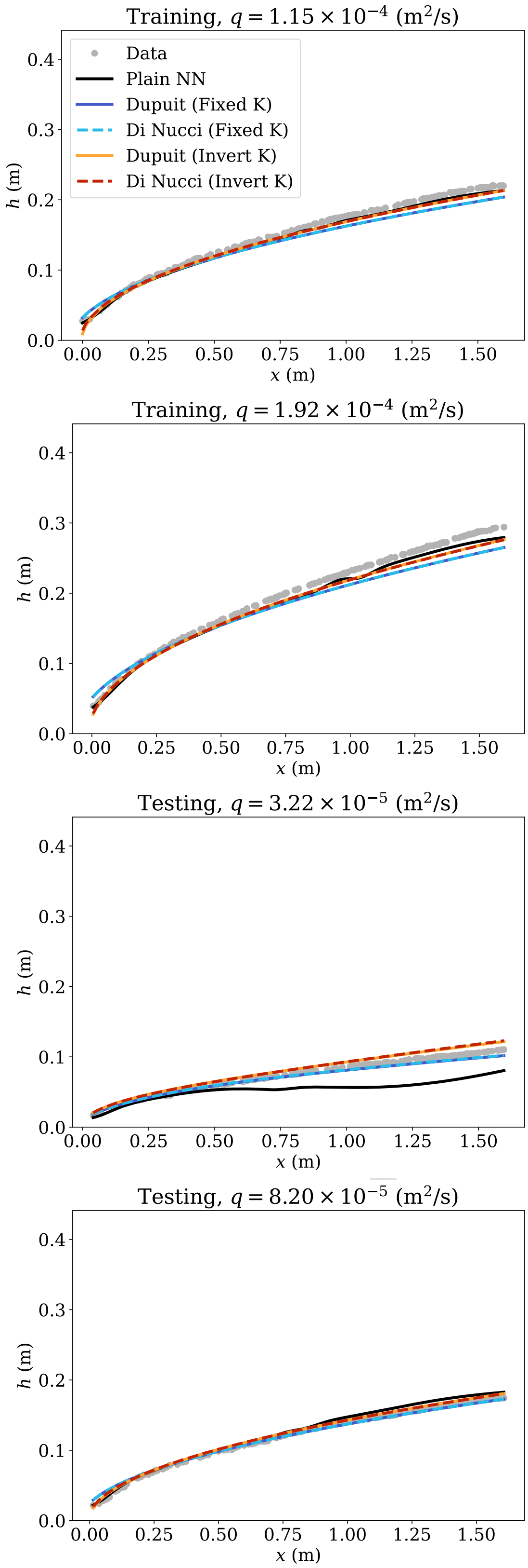



**Figure 14:** Neural network predictions of free surface profiles for the experimental data using 1 mm beads.

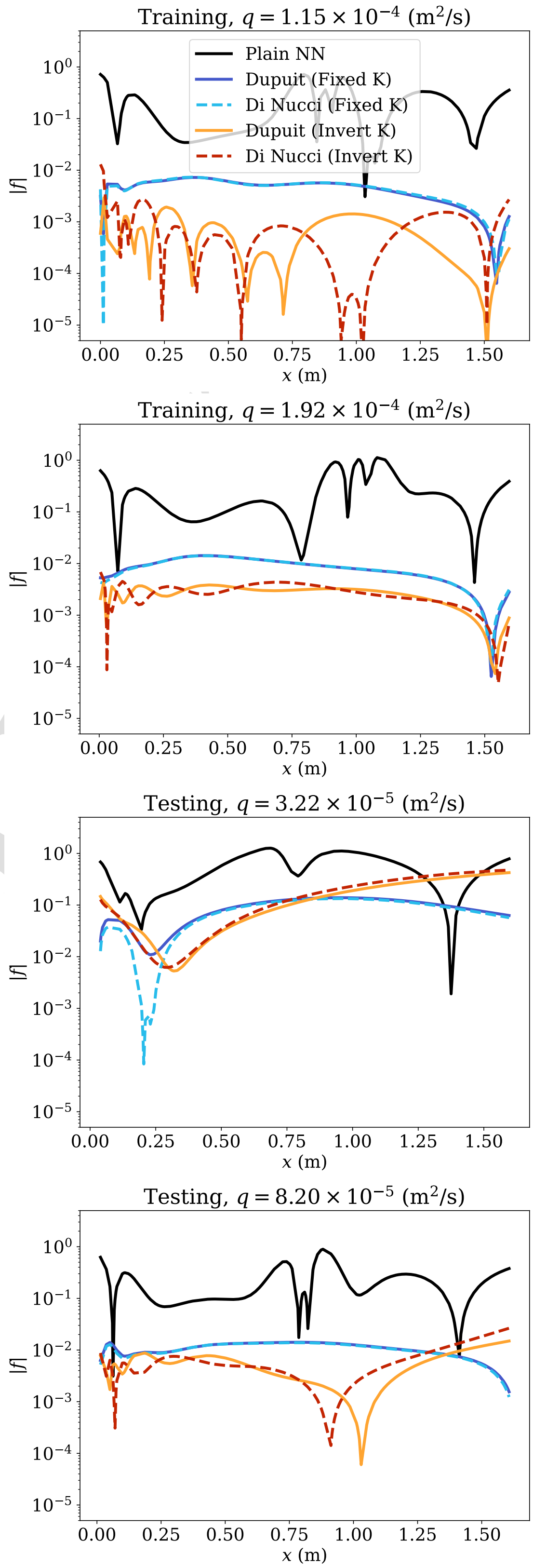



**Figure 15:** The PDE residuals inside the domain corresponding to free surface profile predictions for 1 mm bead size, shown in Figure 14.

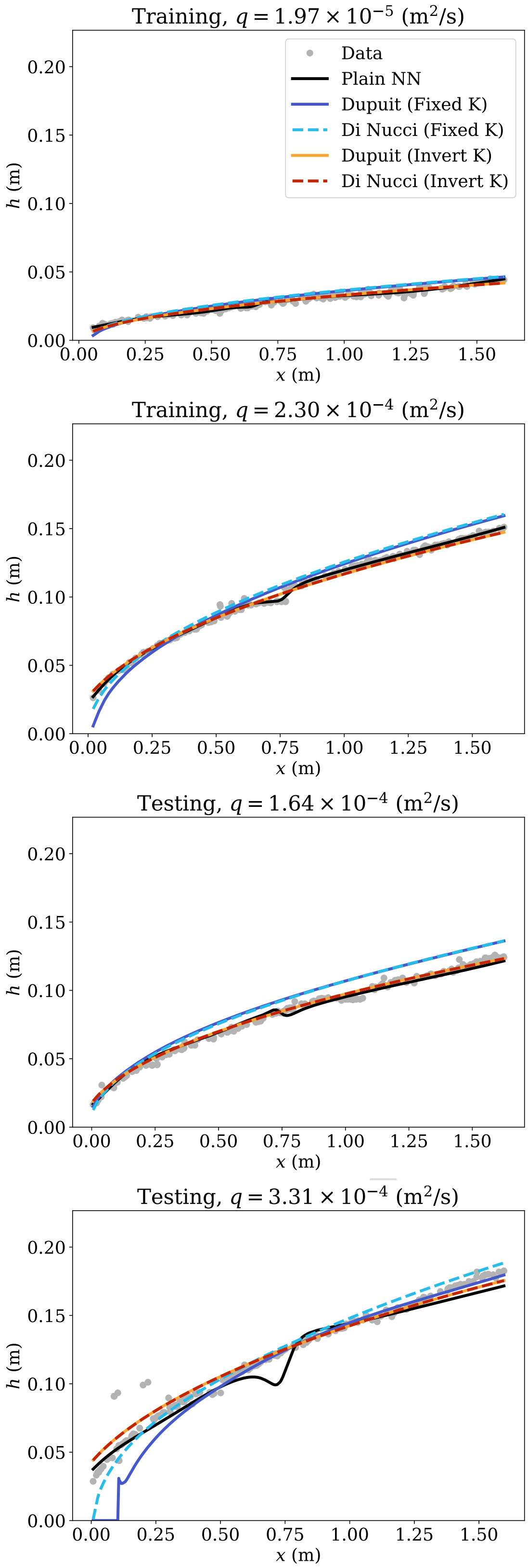



**Figure 16:** Neural network predictions of free surface profiles for the experimental data using 2 mm beads.

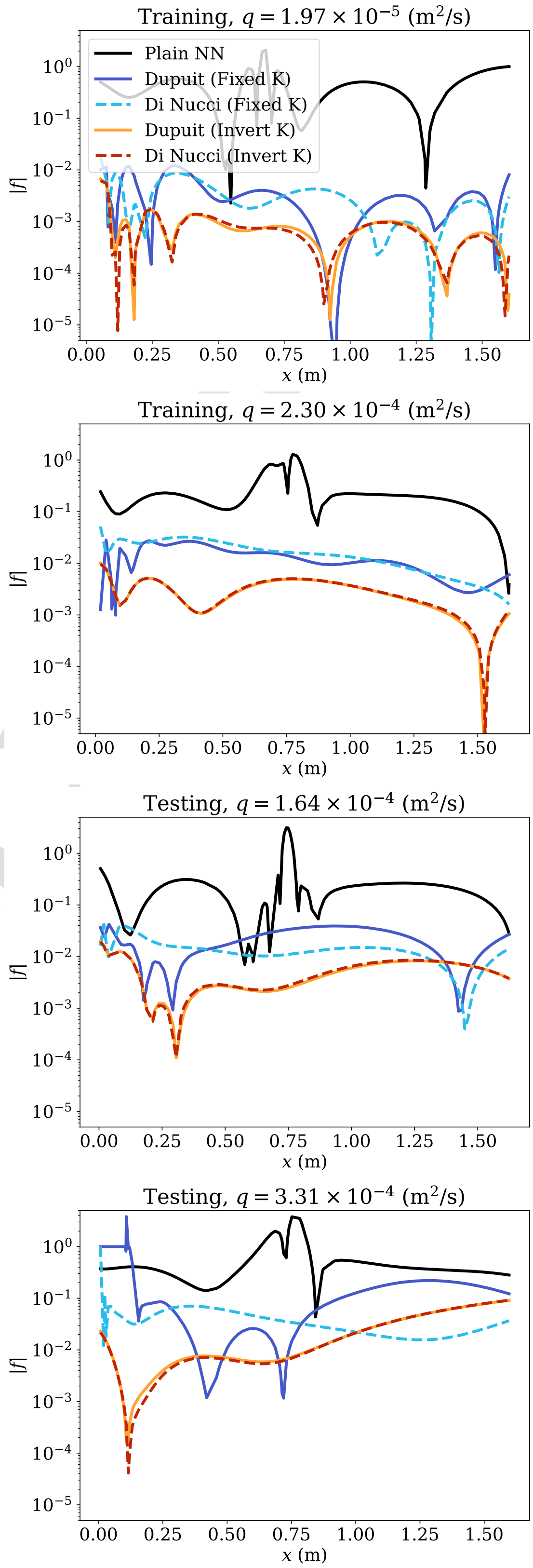



**Figure 17:** The PDE residuals inside the domain corresponding to free surface profile predictions for 2 mm bead size, shown in Figure 16.

# CRediT authorship contribution statement

**Mohammad Afzal Shadab:** Conceptualization of this study, Methodology, Software, Data curation, Writing - Original draft preparation, Supervising.

**Dingcheng Luo:** Conceptualization of this study, Methodology, Software, Data curation, Writing - Original draft preparation.

**Eric Hiatt:** Experimentation, Data curation, Writing - Original draft preparation.

**Yiran Shen:** Software, Data curation, Writing - Original draft preparation.

**Marc Andre Hesse:** Conceptualization of this study, Writing - Editing, Supervising.

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: