AGU ADVANCING EARTH AND SPACE SCIENCES

Water Resources Research

RESEARCH ARTICLE
10.1029/2023WR036589

Physics-Informed Neural Networks for the Augmented System of Shallow Water Equations With Topography

Susanna Dazzi[1]

[1]Department of Engineering and Architecture, University of Parma, Parma, Italy

Key Points:

- Physics-informed neural networks (PINNs) are applied to solve the augmented shallow water equations with topography
- Applications to one-dimensional cases of free-surface flows over non-flat bottom show a good solution accuracy
- Solving the augmented system is an alternative way to deal with non-flat topography

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

S. Dazzi,
susanna.dazzi@unipr.it

Citation:

Dazzi, S. (2024). Physics-informed neural networks for the augmented system of shallow water equations with topography. *Water Resources Research*, *60*, e2023WR036589. https://doi.org/10.1029/2023WR036589

Received 27 OCT 2023
Accepted 24 SEP 2024

Author Contribution:

Conceptualization: Susanna Dazzi
Data curation: Susanna Dazzi
Formal analysis: Susanna Dazzi
Funding acquisition: Susanna Dazzi
Investigation: Susanna Dazzi
Methodology: Susanna Dazzi
Project administration: Susanna Dazzi
Resources: Susanna Dazzi
Software: Susanna Dazzi
Supervision: Susanna Dazzi
Validation: Susanna Dazzi
Visualization: Susanna Dazzi
Writing – original draft: Susanna Dazzi
Writing – review & editing: Susanna Dazzi

Abstract Physics-informed neural networks (PINNs) are gaining attention as an alternative approach to solve scientific problems governed by differential equations. This work aims at assessing the effectiveness of PINNs to solve a set of partial differential equations for which this method has never been considered, namely the augmented shallow water equations (SWEs) with topography. Differently from traditional SWEs, the bed elevation is considered as an additional conserved variable, and therefore one more equation expressing the fixed-bed condition is included in the system. This approach allows the PINN model to leverage automatic differentiation to compute the bed slopes by learning the topographical information during training. PINNs are here tested for different one-dimensional cases with non-flat topography, and results are compared with analytical solutions. Though some limitations can be highlighted, PINNs show a good accuracy for the depth and velocity predictions even in the presence of non-horizontal bottom. The solution of the augmented system of SWEs can therefore be regarded as a suitable alternative strategy to deal with flows over complex topography using PINNs, also in view of future extensions to realistic problems.

1. Introduction

The simulation of flood inundation events is an essential tool for practical purposes, such as hazard assessments, design of mitigation measures, and flood forecasting. The different solution strategies developed in the past decades are essentially based on the Shallow Water Equations (SWEs), which express the conservation of mass and momentum of free-surface flows (Teng et al., 2017). The long-standing development of physics-based numerical methods have led to robust and accurate models that allow large-scale applications with affordable computational time thanks to high performance computing (e.g., Dazzi et al., 2021; Ming et al., 2020; Morales-Hernández et al., 2021; Vacondio et al., 2014) or advanced numerical techniques (e.g., Ferrari et al., 2019; Özgen et al., 2016; Shamkhalchian & De Almeida, 2021; Zokagoa & Soulaïmani, 2018). However, the application of physics-based numerical model to specific problems such as data assimilation and inverse modeling requires using complex schemes and entails a substantial growth of computational time (e.g., Annis et al., 2022; Ferrari et al., 2018; Pujol et al., 2022; Todaro et al., 2019).

A completely different approach is the use of data-driven methods for hydraulic applications (Bomers & Hulscher, 2023; Mosavi et al., 2018), such as machine-learning algorithms (e.g., neural networks). In this case, a "black box" model can be developed to predict one or more output variables based on some input variables; a data set of input-output observations is required to "train" the model, that is, to optimize its internal parameters for minimizing the mismatch between observed and predicted output data. Once the model is trained, new predictions are very fast, hence these models are often proposed as real-time forecasting tools. However, in the training process, the physical laws governing the phenomenon are neglected, which entails the risk of obtaining unphysical predictions.

Recently, research efforts have been dedicated to the enhancement of data-driven methods by enforcing physical constraints during the training process (e.g., Magiera et al., 2020), which helps avoiding unrealistic solutions and improving the predictive accuracy. One of the most promising approaches that combines physics-based and machine-learning strategies is the use of Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) to solve Partial Differential Equations (PDEs) directly. In PINNs, a neural network takes the independent variables (space, time) as input and provides the dependent variables of the PDEs (e.g., pressure, depth, velocity, etc.) as output. For a specific problem, instead of using only already-known labeled input and output data, the neural network is trained to compute a solution that fulfills the governing equations and is coherent with given initial and boundary conditions. Practically, the approximated solution is found by minimizing a loss function that embeds the PDEs residuals computed in few collocation points in the space-time domain, in addition to the residuals computed for

the given initial and boundary conditions. The model can then provide the solution at any other point of the domain. The main advantages of this method include the needlessness of a computational grid (making PINNs a "meshless" method, Bihlo & Popovych, 2022), the possibility of integrating sparse observations in the training process (e.g., Secci et al., 2024), and the suitability for both direct and inverse modeling (Raissi et al., 2019). Drawbacks include the large computational cost (which however can be relieved resorting to GPU computations) and the difficulty in achieving high accuracy (Jagtap et al., 2020). Many versions of PINNs (including advanced or modified schemes) are currently investigated for systems of PDEs in different scientific applications (e.g., Cuomo et al., 2022; Karniadakis et al., 2021; Vadyala et al., 2022), but special attention is dedicated to the solution of Euler and Navier-Stokes equations for fluid dynamics problems (e.g., Jagtap et al., 2022; Mao et al., 2020; Strelow et al., 2023).

From a physical point of view, the SWEs are the depth-averaged version of Navier-Stokes equations assuming a hydrostatic pressure distribution and, from a mathematical point of view, they form a hyperbolic system of PDEs. Therefore, PINNs are expected to solve SWEs as well, and preliminary attempts in this regard can be found in recent works (e.g., Leiteritz et al., 2021; Mahesh et al., 2022). Cedillo et al. (2022) focused their analysis on one-dimensional (1D) steady state flows, which lead to simplified governing equations, and presented an interesting application of PINNs to the inverse problem of estimating the roughness coefficient. Li et al. (2023) solved two-dimensional (2D) dam-break problems with PINNs, but partial analytical solutions were used as prior-known data to help training the network, and topography was not considered. Feng et al. (2023) proposed PINNs as a strategy to solve the 1D SWEs and underlined the feasibility of data assimilation using in situ and remote-sensing observations. Although some simplifications were assumed to account for complex river geometries (Cedillo et al., 2022), these works have shown promising results. Even if it is recognized that PINNs cannot currently replace classical numerical methods due to their lower efficiency and accuracy, some practical applications can be envisaged, such as their use as sub-grid models for the local refinement of the physics-based solution in large-scale river models (Feng et al., 2023). Moreover, very recently, Qi et al. (2024) used PINNs to solve the 2D-SWEs and showed a promising application to a realistic river flood case with topography.

Compared to other types of PDEs in fluid dynamics, some specific problems should be handled when considering SWEs. One is the possible presence of dry regions and the ensuing necessity of enforcing the depth-positivity condition to avoid unphysical solutions (depth ≥ 0). Another one is the presence of source terms in the PDEs (mainly related to topography and friction) in case of real applications. While the friction source term is usually expressed with empirical laws that depend on hydraulic variables (e.g., Chézy-Manning formula) and is not expected to be particularly challenging for PINNs, bed slope source terms should be treated more carefully. In case of real topographies, the bottom elevation can be highly variable in space and include discontinuities that largely affect the flow field. This latter issue has not been investigated in previous works, except for the recent study by Qi et al. (2024).

In mesh-based methods for SWEs (e.g., finite volumes (FV)), bed elevations are provided for each computational cell in the domain among the input data; then, usually, various discretization techniques are applied to evaluate the bed slope source term during the computations (e.g., Aureli et al., 2008; Liang & Marche, 2009; Murillo & García-Navarro, 2010). Another approach sometimes adopted by FV methods is the re-writing of SWEs as an augmented or generalized PDE system (i.e., with the addition of a fictitious equation stating that the bottom elevation, treated as an additional variable, is constant in time) and the development of ad-hoc augmented solvers (e.g., Bernetti et al., 2008; Cozzolino et al., 2011; Dumbser & Balsara, 2016; Rosatti & Begnudelli, 2010). This method is less common than standard source term discretization techniques because, even if it is more accurate and robust near bottom discontinuities, it entails greater mathematical complexity and longer computational times. In PINNs, the strategies to account for source terms can be quite different. In general, constants and other data appearing in the equations (e.g., roughness coefficients) are usually provided as a set of additional input parameters, which can be accessed during PINN training to evaluate PDE residuals (Qi et al., 2024). Focusing on the bed slope source term, however, it should be considered that topography is usually composed of spatially variable bed elevation values; therefore, it is necessary to implement an efficient way to evaluate the local bed slope in the source term for each collocation point where PDE residuals are computed, starting from the corresponding bed elevation at the given spatial coordinate. A possible strategy is the pre-processing of bed elevation gradients with traditional numerical methods (e.g., finite differences, as in Qi et al. (2024)). In this work, a completely alternative approach is explored, which is inspired by the idea of system augmentation of FV methods. In short, the bed elevation is here treated as an additional output variable that the PINN model is trained to predict

and that is provided together with other initial conditions; moreover, a physical constraint stating its stationarity (time derivative equal to zero) is added to the loss function. In this way, PINNs can leverage automatic differentiation to avoid pre-processing steps and compute the bed slope source term directly within the same network that handles the other variables, and somehow "learn" the topographical information during training. Practically, this approach corresponds to setting up a PINN model that solves the augmented system of SWEs instead of the standard SWEs.

Accordingly, this work aims at assessing the effectiveness of PINNs in solving the SWEs with topography adopting the augmented formulation. As a first attempt of applying PINNs in this context, the analysis is restricted to the one-dimensional (1D) case (one spatial variable only) and to synthetic topographies with no friction. Although this may appear a simplification, a few test cases that are quite challenging for classical numerical methods fall in this category and are included in the paper (e.g., Riemann problem on a bottom step; oscillating planar surface on a parabolic basin). The availability of analytical solutions for these tests is fundamental to perform a rigorous validation. Finally, the potentiality of PINNs to embed sparse observations in addition to physical constraints is shown for one of the tests without analytical solution.

The paper is structured as follows. The governing equations, the PINN method, the test cases, and the model setup are described in Section 2. Results are presented in Section 3, while Sections 4 and 5 are dedicated to discussion and conclusions, respectively.

2. Methods

2.1. Augmented System of Shallow Water Equations (SWEs)

The 1D augmented system of SWEs can be expressed as follows:

$$\frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} = 0 \tag{1a}$$

$$\frac{\partial uh}{\partial t} + \frac{\partial}{\partial x}\left(u^2 h + \frac{1}{2} g h^2\right) + gh\frac{\partial z}{\partial x} + ghS_f = 0 \tag{1b}$$

$$\frac{\partial z}{\partial t} = 0 \tag{1c}$$

where x is the spatial coordinate, t is the time, h is the water depth, u is the velocity, z is the bed elevation (above datum), g is the acceleration due to gravity, and S_f is the friction slope (expressed for example with the Chézy-Manning formula). In the following test cases, a frictionless bottom is assumed ($S_f = 0$). Equations 1a and 1b represent the standard SWEs where z appears only in the bed slope source term (third term in Equation 1b), while Equation 1c is the "fictitious" equation expressing the fixed-bed condition that is added to the system to include z as an additional conserved variable. For a detailed treatment of the mathematical properties of the augmented system and of the solution strategies with traditional numerical techniques, which are beyond the scope of this paper, the reader is referred to the literature (e.g., Bernetti et al., 2008; Cozzolino et al., 2011; Dumbser & Balsara, 2016; Rosatti & Begnudelli, 2010).

2.2. Physics-Informed Neural Networks (PINNs)

2.2.1. Introduction to PINNs

In this Section, the general idea of PINNs is explained. Let us consider a generic physical problem, defined for time coordinate $t \in [t_0, t_f]$ and space coordinate $x \in [x_0, x_f]$, which is governed by a system of n_{eq} PDEs with n_{var} dependent variables (indicated by $\mathbf{U}$), and with given initial conditions ICs (i.e., known variables for $t = t_0$ and $x \in [x_0, x_f]$) and boundary conditions BCs (i.e., prescribed variables/functions at $x = x_0$ and at $x = x_f$ for $t \in [t_0, t_f]$). Each PDE in the system can be expressed as $f_j(x, t, \mathbf{U}) = 0$ with $j = 1, \ldots, n_{\text{eq}}$, where f_i contains differential operators, and all terms are on the LHS. Notice that a 1D problem (only x as spatial coordinate) is considered here for the sake of simplicity, but PINNs are also applicable to multi-dimensional problems.

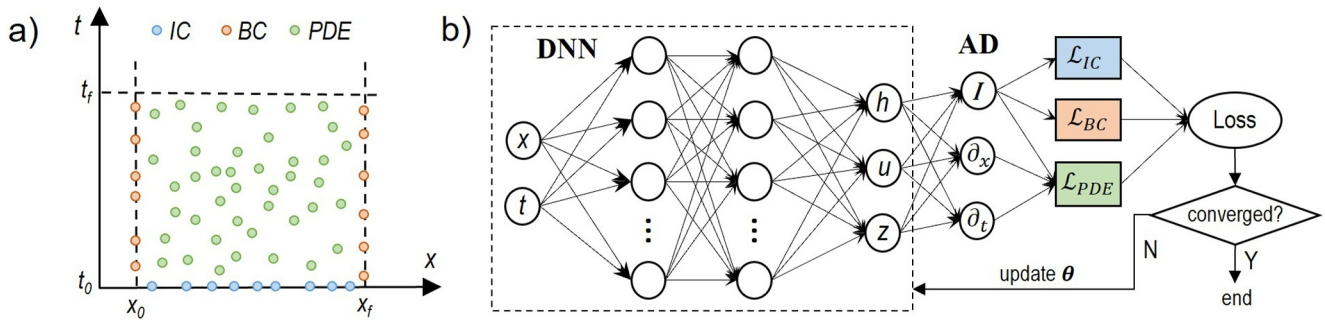

Figure 1. Sketch of (a) collocation points in the space-time domain, and (b) PINN model.

The PINN model is essentially a feed-forward deep neural network (DNN) that is built and trained to approximate the solution of the above problem. The independent variables (x, t) are fed to the input layer of the network, processed by neurons in the hidden layers, and finally the output layer provides a prediction $\hat{\mathbf{U}}$ of the dependent variables. The trainable parameters of the model (i.e., the weights and biases of each neuron) can be collectively indicated by $\boldsymbol{\theta}$. In summary, the PINN model (generically indicated by $\mathcal{F}_{\text{PINN}}$) can be expressed as $\hat{\mathbf{U}} = \mathcal{F}_{\text{PINN}}(x, t, \boldsymbol{\theta})$. For more detailed explanations about the architecture of DNNs, the reader is referred to the wide literature on the topic (e.g., Haykin, 2009).

The training phase of a DNN consists of the optimization of parameters $\boldsymbol{\theta}$ by minimizing a suitable loss function that penalizes deviations of the predicted values $\hat{\mathbf{U}}$ from the true solution $\mathbf{U}$. Training of DNNs is based on the back-propagation algorithm (Haykin, 2009) and can be performed using different optimizers, such as stochastic gradient descent or Adam (Kingma & Ba, 2014). The main difference that distinguishes PINNs from standard DNNs is the loss function $\mathcal{L}$, which here embeds the physical constraints coming from the PDEs, ICs, and BCs, and can be expressed as follows:

$$\mathcal{L} = \sum_{j=1}^{n_{\text{eq}}} (w_{\text{PDE},j}\mathcal{L}_{\text{PDE},j}) + w_{\text{IC}}\mathcal{L}_{\text{IC}} + w_{\text{BC}}\mathcal{L}_{\text{BC}} \tag{2}$$

where w_{PDE}, w_{IC}, and w_{BC} are weights used to balance the relative importance of the different loss terms $\mathcal{L}_{\text{PDE},j}$, $\mathcal{L}_{\text{IC}}$, and $\mathcal{L}_{\text{BC}}$:

$$\mathcal{L}_{\text{PDE},j} = \frac{1}{N_{\text{int}}}\sum_{i=1}^{N_{\text{int}}}\left[f_j\left(x_i, t_i, \hat{\mathbf{U}}(x_i, t_i, \boldsymbol{\theta})\right)\right]^2 \tag{3a}$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{ic}}\sum_{k=1}^{n_{\text{var}}}\sum_{i=1}^{N_{ic}}\left[\hat{\text{U}}_k\left(x_i, t_0, \boldsymbol{\theta}\right) - \text{U}_k(x_i, t_0)\right]^2 \tag{3b}$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_{bc}}\sum_{k=1}^{n_{\text{var}}}\left\{\sum_{i=1}^{N_{bc}}\left[\hat{\text{U}}_k(x_0, t_i, \boldsymbol{\theta}) - \text{U}_k(x_0, t_i)\right]^2 + \sum_{i=1}^{N_{bc}}\left[\hat{\text{U}}_k\left(x_f, t_i, \boldsymbol{\theta}\right) - \text{U}_k\left(x_f, t_i\right)\right]^2\right\} \tag{3c}$$

In particular, the loss $\mathcal{L}_{\text{PDE},j}$ penalizes the residuals of the jth governing equation (i.e., deviations from zero, which indicate that the physical law is not respected), computed in N_{int} collocation points inside the space-time domain, while the loss terms $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{BC}}$ penalize deviations from the prescribed values of initial and boundary conditions, computed on a discrete set of points N_{ic} and N_{bc}. Notice that Equation 3c can be easily modified to consider prescribed functions for the boundary values (possibly including differential operators), instead of prescribed variables. As regards the collocation points, their location is usually random, although recent works suggest that the solution accuracy might improve if points are clustered close to high-gradient regions (e.g., Mao et al., 2020). A schematic illustration of collocation points in the space-time domain is shown in Figure 1a, while Figure 1b schematizes the PINN model with the input/output variables considered in this work (see Section 2.2.2).

The PDE residuals evaluation requires partial derivatives to be computed. For this task, PINNs leverage the automatic differentiation (AD) technique (Baydin et al., 2018), which basically applies the chain rule to evaluate the derivatives of composite functions, interpreted as a combination of elementary operations for which the derivative is known. In PINNs, AD is exploited to compute the temporal and spatial derivatives of variables and functions that appear within the PDEs (i.e., the derivatives with respect to the input values x, t) directly in the computational graph. Compared to numerical differentiation, AD is not affected by truncation or discretization errors and produces exact derivatives up to finite-precision error.

One of the main advantages of PINNs is the possibility of adding different constraints to the loss function, depending on the problem at hand. The most notable example is the inclusion of sparse observations, if available, to condition the training process and obtain more reliable predictions. To this purpose, Equation 2 can be enriched with a loss term $\mathcal{L}_{\text{obs}}$, characterized by its own loss weight w_{obs}, which penalizes deviations from observed data at specific points (N_{obs}) in the space-time domain for which measurements of the output variables are available.

It should be stressed that, whenever initial and/or boundary conditions are modified (or the spatio-temporal domain is extended), the PINN model needs to be trained again since it is not transferrable. For this reason, the PINN training can be considered similar to running a simulation with an alternative numerical method for the solution of the physics-based PDEs.

2.2.2. The PINN Model Used for the Augmented SWEs

The previous Section provided an overview of a generic PINN model, while this Section describes the specific features of the PINN model here adopted to solve the augmented 1D-SWEs (Equation 1).

In this work, the model is created in Python using NVIDIA Modulus framework (NVIDIA, 2022), previously known as SimNET (Hennigh et al., 2021), which builds on the Pytorch package. It has the advantage of being optimized for GPUs, therefore reducing the computational time for training. Compared to other PINN frameworks that adopt the summation of residuals (Equation 3), Modulus adopts an integral formulation of losses, which is then discretized using Monte Carlo integration (Hennigh et al., 2021).

In this PINN model for augmented SWEs, the input variables are space x and time t, while the output variables are water depth h, velocity u, and bed elevation z (Figure 1b). The domain is $t \in [t_0, t_f]$ and $x \in [x_0, x_f]$. Given that z is a function of x in the computational graph, the bed slope in the source term ($\partial z/\partial x$) can be automatically evaluated with AD during training.

The initial conditions are set by providing the known values of the three dependent variables along x (in a subset of N_{IC} points) at $t = t_0$, thus including the topographical information in the z variable. Upstream and downstream boundary conditions are set along t (in a subset of N_{BC} points) at $x = x_0$ and at $x = x_f$. Notice that time-varying BCs are not considered in the current test cases (listed in Section 2.3), and that all test cases adopt prescribed values on the boundaries except one (steady-state flow), for which the value of the derivatives is also imposed (see Section 2.4).

In addition to the loss terms related to ICs, BCs, and the three PDEs, two additional constraints are added to enforce depth positivity ($h \geq 0$) everywhere, and null velocity where the bed is dry. To this end, two new equations that include the Heaviside step function are used:

$$f_{h\_\text{pos}} = h - h \cdot \text{Heaviside}(h + \epsilon) \tag{4a}$$

$$f_{u\_\text{dry}} = u - u \cdot \text{Heaviside}(h + \epsilon) \tag{4b}$$

where ϵ is a small threshold set equal to 10^{-12}. These conditions are automatically fulfilled at wet locations. These constraints are evaluated at the same collocation points used for the PDEs residuals, thus generating two additional loss terms:

$$\mathcal{L}_{h\_\text{pos}} = \frac{1}{N_{\text{int}}} \sum_{i=1}^{N_{\text{int}}} \left[f_{h\_\text{pos}}\left(x_i, t_i, \hat{\mathbf{U}}(x_i, t_i, \boldsymbol{\theta})\right) \right]^2 \tag{5a}$$

Table 1
List of Test Cases

Test name	Bottom	Flow	Exact solution	Spatial domain (m)	Temporal domain (s)	Description
SW1	Flat	Static	Yes	$-6 \le x \le 6$	$0 \le t \le 1$	Still water on a flat bed
SW2	Bump	Static	Yes	$-10 \le x \le 10$	$0 \le t \le 10$	Still water on a bump
SW3	Step	Static	Yes	$-6 \le x \le 6$	$0 \le t \le 1$	Still water on a bottom step
SW4	Parabola	Static	Yes	$-2 \le x \le 2$	$0 \le t \le 1$	Still water on a parabolic bed
SF1	Bump	Steady	Yes	$-10 \le x \le 10$	$0 \le t \le 100$	Subcritical flow over bump
UF1	Flat	Unsteady	No*	$-6 \le x \le 6$	$0 \le t \le 1$	Small perturbation case
UF2	Flat	Unsteady	Yes	$-6 \le x \le 6$	$0 \le t \le 1$	Dam break
UF3	Step	Unsteady	Yes	$-6 \le x \le 6$	$0 \le t \le 1$	Dam break with step
UF4	Parabola	Unsteady	Yes	$-2 \le x \le 2$	$0 \le t \le 2.006$	Thacker 1D

Note. *A reference solution was obtained from a finite-volume solver with very high resolution.

$$\mathcal{L}_{u\_\mathrm{dry}} = \frac{1}{N_{\mathrm{int}}}\sum_{i=1}^{N_{\mathrm{int}}}\left[f_{u\_\mathrm{dry}}\left(x_i, t_i, \hat{\mathbf{U}}(x_i, t_i, \boldsymbol{\theta})\right)\right]^2 \tag{5b}$$

This strategy has no influence in wet regions, but makes the PINN model significantly penalize negative depths in dry regions, even if it may slightly complicate the training process because it also requires mass and momentum conservation to be fulfilled in these regions. However, in this way, the PINN model is allowed to learn the distribution of wet/dry regions (which are not known a priori) during training. Note that alternative approaches, such as setting the loss to zero when depth is negative (Qi et al., 2024) or using the logarithm as an output variable (Strelow et al., 2023), have been recently proposed to deal with non-negative variables such as depth and pressure.

As regards the loss weights, different values, indicated by w_{mm} and w_{z_fixed}, are assigned to the loss of the first two equations of the SWE system (mass and momentum conservation) and to the loss of the "fixed bed" equation, respectively. Moreover, a loss weight w_{h_pos} is assigned to the depth-positivity conditions (Equation 5). Finally, as regards the initial and boundary conditions, a loss weight w_z is dedicated to the residuals of z, higher than the loss weights w_{IC} and w_{BC} adopted for the residuals of h and u on ICs and BCs, respectively. The assignment of different weights to different loss terms guarantees more flexibility in the model setup depending on the test case. In particular, for augmented SWEs, the deviations from the fixed-bed condition and from the initial bed elevations can be penalized harder than other residuals, in order to ensure that the bed elevation is correctly learned from ICs and does not evolve in time. In summary, the loss is computed as follows:

$$\begin{aligned}\mathcal{L} = {} & w_{\mathrm{mm}}\left(\mathcal{L}_{\mathrm{PDE,\,mass}} + \mathcal{L}_{\mathrm{PDE,\,mom}}\right) + w_{z\_\mathrm{fixed}}\mathcal{L}_{\mathrm{PDE},z\_\mathrm{fixed}} + w_{\mathrm{IC}}\mathcal{L}_{\mathrm{IC},h,u} + w_{\mathrm{BC}}\mathcal{L}_{\mathrm{BC},h,u} + w_z\left(\mathcal{L}_{\mathrm{IC},z} + \mathcal{L}_{\mathrm{BC},z}\right) \\ & + w_{h\_\mathrm{pos}}\left(\mathcal{L}_{h\_\mathrm{pos}} + \mathcal{L}_{u\_\mathrm{dry}}\right)\end{aligned} \tag{6}$$

Finally, for tests with data assimilation, the loss function of Equation 6 includes an additional term $w_{\mathrm{obs}}\mathcal{L}_{\mathrm{obs}}$ that penalizes deviations from observed data available at a discrete number of $x - t$ points.

The setup of model hyperparameters is discussed in Section 2.4.

2.3. Test Cases

Different test cases are considered for assessing the model accuracy, which are summarized in Table 1. All cases are frictionless. In the following equations, water depths h, bed elevations z and spatial coordinates x are expressed in m, temporal coordinates t in s, and velocities u in m/s.

2.3.1. Topography

Four types of topography are analyzed in this work, for which both still-water tests and steady- or unsteady-flow tests are defined. The topographies will be graphically depicted in the figures of Section 3 alongside the results.

The first one is the simple flat bottom with equation $z(x) = 0$.

Then, the flat bottom with a bump in the middle (e.g., Delestre et al., 2013) is considered, where the bed elevation is described by the following equation:

$$z(x) = \begin{cases} 0.2 - 0.05x^2 & \text{if } -2 < x < 2 \\ 0 & \text{elsewhere} \end{cases} \tag{7}$$

Another challenging topography is the case of the positive bottom step, with the following expression for bed elevation:

$$z(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{8}$$

The last topography is for the 1D Thacker problem (Delestre et al., 2013), where the bed elevation is a parabola with equation:

$$z(x) = h_0 \left[\left(\frac{x}{a} \right)^2 - 1 \right] \tag{9}$$

where $h_0 = 0.5$ and $a = 1$.

2.3.2. Still-Water Tests

A set of four still-water tests (SW1-SW4) is first considered in order to check the ability of the PINN-SWE to replicate the maintenance of a static solution in time. Temporal and spatial domains are specified in Table 1. For validation, the results of PINN-SWEs for test cases SW1-SW4 are compared with the analytical solution, that is, the preservation of the still-water condition (set as ICs) in time. Notice that, for all cases, BCs are the constant values of depth and velocity at the upstream and downstream end of the domain.

The first test (SW1) is characterized by a horizontal bed and by a constant water depth $h(x) = 1$ m and a null velocity $u(x) = 0$.

Then, the three types of non-horizontal bottom are analyzed. First (SW2), the bump topography (Equation 7) is selected, with initial conditions $h(x) + z(x) = 0.5$ m and $u(x) = 0$.

Case SW3 concerns still water on a positive bottom step (Equation 8), with initial conditions $h(x) + z(x) = 2$ m and $u(x) = 0$.

Finally, still water in a parabolic basin (Equation 9) is considered for case SW4. The initial conditions are $h(x) + z(x) = 0$ (only where $h > 0$, i.e., $-1 \leq x \leq 1$) and $u(x) = 0$.

2.3.3. Steady and Unsteady Flow Tests

One steady flow example (SF1) is considered next, taken from a very common test case for validating numerical models, that is, the simulation of the steady flow over a bump (Equation 7). A constant specific discharge $q_u = 4.42$ m^2/s at the upstream section and water depth $h_d = 2$ m at the downstream outlet are set as boundary conditions. Initial conditions are $h(x) + z(x) = 2$ m and $u(x) = q_u/h(x)$. The steady state is checked at $t = 100$ s. The analytical solution for the water depth $h(x)$ can be obtained by solving the following equation (Delestre et al., 2013):

$$h(x)^3 + \left[z(x) - \frac{q_u^2}{2gh_d^2} - h_d \right] h(x)^2 + \frac{q_u^2}{2g} = 0 \tag{10}$$

whereas the velocity can be simply computed as $u(x) = q_u/h(x)$.

The four examples of unsteady flows (UF1-UF4) specified in Table 1 are then considered. For all cases, BCs are constant values of depth and velocity, equal to those provided by initial conditions, at the inlet and outlet of the domain.

The first case (UF1) concerns the propagation of a small perturbation on a horizontal frictionless bottom. Initial conditions are:

$$h(x) = 0.8 + 0.2\exp\left(-\frac{x^2}{0.4}\right) \text{ and } u(x) = 0 \tag{11}$$

The solution of this problem cannot be determined analytically; therefore, a reference solution, considered accurate enough for validation purposes, was obtained from a very high-resolution simulation performed using a standard FV numerical scheme with a HLL approximate Riemann solver (Toro, 2001). The domain was discretized with cells of size 0.01 m, and the Courant number was set equal to 0.8. Case UF1 is also selected to provide an example concerning the assimilation of sparse observations into the PINN model training. To this end, a limited number of depth values extracted from the FV solution at random spatial locations and at the final time ($t = 1$ s) are used to represent "observed data" for the analyses. Velocity observations are not included, because measurements of this variable are seldom available in the practice.

Case UF2 is a classic dam-break problem on a horizontal bed with initial conditions:

$$h(x) = \begin{cases} 2 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad \text{and} \quad u(x) = 0 \tag{12}$$

for which the analytical solution can be obtained from an exact Riemann solver (Toro, 2001).

Moving on to non-flat topography, the next test case (UF3) is a dam-break problem on a positive bottom step (Equation 8), whose analytical solution was retrieved from the literature (Rougier, 2022). Initial conditions are:

$$h(x) = \begin{cases} 3 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \text{and } u(x) = 0 \tag{13}$$

Finally, another challenging test case (UF4), often used for the validation of numerical models, is considered, namely the 1D Thacker problem (Thacker, 1981) concerning the oscillation of a planar surface on a parabolic basin (Equation 9) without friction. The analytical solution is:

$$h(x,t) = \max\left(0; -h_0\left[\frac{1}{a^2}(x + 0.5\cos(\omega t))^2 - 1\right]\right) \tag{14a}$$

$$u(x,t) = 0.5\omega\sin(\omega t) \quad \text{where } h(x,t) > 0 \tag{14b}$$

where $\omega = \sqrt{2gh_0}/a$, and h_0 and a are defined as in Equation 9. The initial conditions can be obtained from Equation 14 by setting $t = 0$. The temporal domain is extended up to one period ($T = 2\pi/\omega$), that is, approximately 2.006 s.

2.4. Model Setup

A fully connected neural network is adopted for all tests, including 7 hidden layers with 300 neurons each, while the activation function is tan$h$. The optimization algorithm is Adam, and the learning rate decays exponentially from 10^{-3} to 10^{-6} after 30'000 epochs. The following values are assigned to the loss weights: $w_{\text{mm}} = 1$; $w_{z\_\text{fixed}} = 100$; $w_{\text{IC}} = 10$; $w_{\text{BC}} = 1$; $w_z = 100$; $w_{h\_\text{pos}} = 10$. Moreover, the collocation points are randomly distributed, and their number is as follows: $N_{\text{int}} = 1{,}000$; $N_{ic} = 200$; $N_{bc} = 200$ (100 for each boundary). Notice

that the choice of using a random distribution of collocations points avoids the necessity of pre-defining the locations of points in the space-time domain, making PINN a completely meshless method.

This configuration (defined "base" in the following) was derived from a preliminary grid-search optimization of the hyperparameters based on case UF2, which is described in detail of the Supporting Information S1 (Text S1). In particular, it should be underlined that increasing the order of magnitude of loss weights w_z and w_{z_fixed}, related to the topographical constraints, is useful to improve the accuracy of results. The grid-search procedure is quite burdensome; therefore, the same hyperparameters were at first applied to all the other test cases. Then, if unsatisfactory results were obtained for specific test cases, modifications were made to the hyperparameter configuration by trial-and-error. In particular, the following tests required additional fine-tuning:

- Test SF1. For this case, the ICs are just a starting point to reach the steady state. Therefore, their importance on the total loss is reduced by setting $w_{\text{IC}} = 1$. The weights w_{z_fixed} and w_z are also reduced to 10, and the number of collocation points and epochs is doubled to increase the accuracy ($N_{\text{int}} = 2{,}000$, $N_{ic} = 400$, $N_{bc} = 200$ for each boundary; 60'000 epochs). Moreover, the BCs are modified by setting the conditions $uh = q_u$ and $\frac{\partial uh}{\partial x} = 0$ upstream, and $h = h_d$ and $\frac{\partial h}{\partial x} = 0$ downstream, and the BC losses are modified accordingly.
- Test UF4. A much larger number of collocation points is required ($N_{\text{int}} = 12'000$, $N_{ic} = 2{,}500$, $N_{bc} = 1{,}000$, with 500 for each boundary). Moreover, the w_z and w_{z_fixed} are reduced to 10, and the learning rate is set to decay from 10^{-3} to 10^{-5} after 60'000 epochs.

As anticipated in the previous Section, test UF1 is also used to provide an example of assimilation of sparse observations. Observed water depths are extracted from the FV reference solution at the final time $t = 1$ s and at N_{obs} random spatial coordinates along x. With the purpose of representing a case for which only very few observations are available, the value of N_{obs} is set to 10. The loss weight w_{obs} can be set to higher or lower values according to the uncertainty associated with measured data. These considerations are beyond the scope of this paper. In this example, given the limited number of observations, the value $w_{\text{obs}} = 10$ is assumed. These tests are performed in the following two configurations:

- OBS_A: the hyperparameters are set as in the case UF1 without observations ("base" configuration);
- OBS_B: the number of collocation points and epochs is halved compared to the "base" configuration.

2.5. Error Metrics

For all tests and configurations, the PINN model is trained with the full data set of unlabeled collocation points (N_{int}) along with the labeled points of ICs and BCs (N_{ic}, N_{bc}), and the loss to minimize is reported in Equation 6. Once the model is trained, its predictive accuracy must be evaluated: this analysis is performed by comparing the PINN predictions with the true values obtained from the analytical solution at selected points in the space-time domain (not used for training the PINN models).

To this end, the output variables are inferred from each trained PINN model at points $(\bar{\boldsymbol{x}}, \bar{t})$ where $\bar{\boldsymbol{x}}$ is a vector of N equally spaced spatial coordinates that are used to discretize the spatial domain $[x_0, x_f]$, and $\bar{t}$ is a pre-defined time coordinate. Evaluating the solution at this subset of points corresponds to extracting the longitudinal profile of each output variable at a selected time, thus providing results in a form that can be easily compared with analytical or numerical solutions. For all test cases, N is assumed equal to 1,000. For still water and steady flow cases, the longitudinal profiles are analyzed at the final time ($\bar{t} = t_f$), while for unsteady flow cases the errors are evaluated for the longitudinal profiles at five selected times ($\bar{t} = 0, 0.25t_f, 0.5t_f, 0.75t_f, t_f$) to show the temporal evolution of the phenomenon.

The error metrics here used are the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{h}_i - h_i\right)^2} \tag{15a}$$

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

Table 2
Still-Water and Steady-Flow Tests

Test name	MAE h (m)	RMSE h (m)	MAE u (m)	RMSE u (m)	MAE z (m)	RMSE z (m)
SW1	2.2E−04	2.6E−04	6.1E−04	6.9E−04	3.9E−05	5.2E−05
SW2	3.9E−03	5.7E− 03	5.2E−03	5.9E−03	3.2E−03	5.2E−03
SW3	5.9E−04	2.9E−03	9.2E−04	1.2E−03	3.6E−04	2.8E−03
SW4	3.2E−03	4.7E−03	4.1E−03	4.9E−03	1.4E−03	1.8E−03
SF1	5.1E−03	1.0E−02	6.3E−03	1.3E−02	1.4E−03	1.8E−03

Note. Error metrics (MAE and RMSE) of PINN-augmSWE results compared to the analytical solution, for depth, velocity and bed elevation profiles extracted at the final time.

where N is the discrete number of values used to evaluate the error, $\hat{h}$ and h refer to the predicted and true value of the generic variable for a given (x, t) point (here the depth h, but analogous expressions can be written for the velocity u or the bed elevation z). The metrics are here computed for the longitudinal profile of each output variable.

3. Results

In this Section, results of PINNs for augmented SWEs (PINN-augmSWE) are presented. Sections 3.1–3.3 report the results for all the test cases listed in Table 1. Sections 3.4 and 3.5 show the comparison with the results obtained from PINNs solving non-augmented SWEs and from FV schemes, for one test case. Section 3.6 is dedicated to an example of data assimilation. As anticipated, results are not represented in the space-time domain, but the output variables (depth, velocity, bed elevation) are predicted in terms of longitudinal profiles at a predefined time $\bar{t}$. Notice that the bed elevation z is also inferred by the PINN model.

For all cases, training process was performed with a NVIDIA A100 GPU device. Training times were in the order of 7–8 min for tests with "base" configuration ($N_{\text{int}} = 1{,}000$).

3.1. Still Water Tests

The results of test cases SW1-SW4 are reported in this Section. For all cases, the profiles are extracted at $\bar{t} = t_f$ (final time), and the preservation of the lake-at-rest condition is checked. Values of RMSE and MAE computed for the N points along the profiles are reported in Table 2.

For test SW1 (still water on a horizontal bottom), PINNs proposed in this paper can predict the depth profile (Figure 2a) with MAE and RMSE values as low as 2.2E−4 and 2.6E−4, respectively (Table 2). The velocity errors are up to one order of magnitude greater (1E−3) at some locations (Figure 2b), but the RMSE and MAE remain in the order of 6E−4. Finally, the horizontal bed elevation profile is predicted more accurately (Figure 2c), with RMSE and MAE in the order of 1E−5. In general, the slight deviations from the exact solution do not show any significant trend, that is, neither systematic under- or over-estimation, nor localization of large errors at specific spatial coordinates.

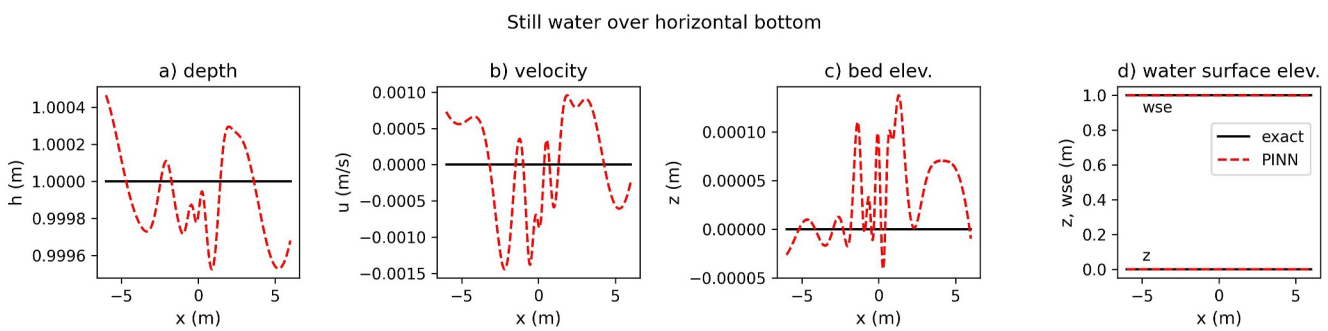

Figure 2. Case SW1: depth, velocity, bed, and water surface elevation profiles at the final time. Comparison between analytical solution and PINN-augmSWE results.

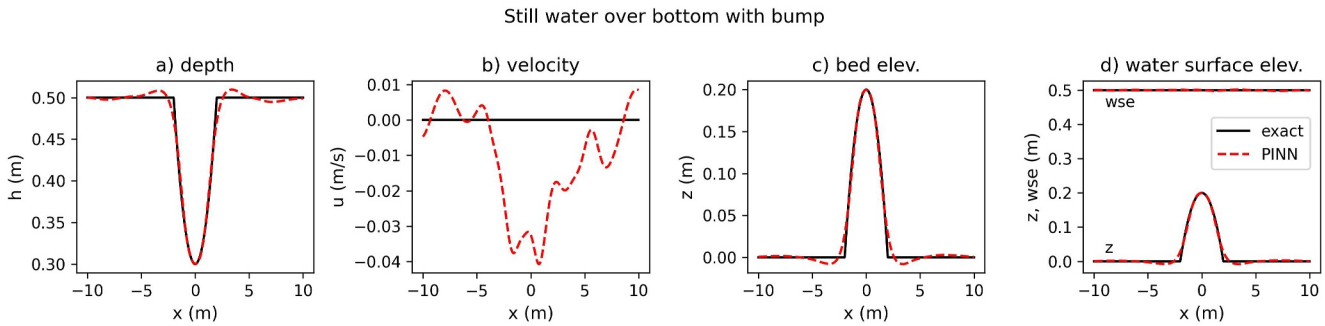

Figure 3. Case SW2: depth, velocity, bed, and water surface elevation profiles at the final time. Comparison between analytical solution and PINN-augmSWE results.

Moving to tests with non-horizontal bottom, Figure 3 shows the profiles for case SW2 (still water on a horizontal bottom with a bump). The error metrics (RMSE and MAE) of depth, velocity and bed elevation are all in the order of 1E−3 (Table 2). However, profiles show that PINN results are affected by larger errors close to the slope changes (just upstream and downstream of the bump), where the depth is locally overestimated (Figure 3a) and the bed elevation is locally underestimated (Figure 3c). However, their summation (water surface elevation) is approximately correct (Figure 3d) and the velocity profile is not locally affected (Figure 3b).

A similar problem can be observed in Figure 4, concerning case SW3 (still water on a bottom step). While no significant deviations can be observed in the portions of the channel with horizontal bottom, depth is slightly overestimated just upstream of the step and underestimated just downstream (Figure 4a), while the bed elevation locally shows an opposite trend (Figure 4c) and, therefore, the water surface elevation is correctly predicted (Figure 4d). Overall, RMSEs and MAEs are still close to 1E−3 (Table 2).

Finally, results for case SW4 (still water on a parabolic bottom), which presents dry regions, are summarized in Figure 5. Despite the constraints on depth positivity, some slightly negative depth values (Figure 5a) and non-null velocity values in dry regions (Figure 5b) are present in the PINN solution. However, the errors computed for the wet and dry regions separately are all comparably low and similar to the errors for the full profile (MAE remains in the order of 1E−3 for depth and velocity). Overall, the error metrics are in line with the other tests (Table 2). In particular, the largest error on the bed elevation is in the order of 1E−3 (Figure 5c).

3.2. Steady Flow Test

Case SF1 (steady flow on a bottom with bump) is treated as an unsteady problem, starting from prescribed initial conditions and checking that, at the final time, the steady state is achieved. Therefore, the profiles are extracted only at $\bar{t} = t_f$ (final time), ignoring the transient phase.

Results are summarized in Figure 6 and Table 2. The depth and bed elevation profiles show the same issue already observed in case SW2 just upstream and downstream of the bump (Figures 6a–6c). Moreover, the depth is captured with a slight inaccuracy on the top of the bump (underestimation), where the velocity is overestimated (Figure 6b). The MAE of all these profiles (Table 2) is characterized by the same order of magnitude of still water

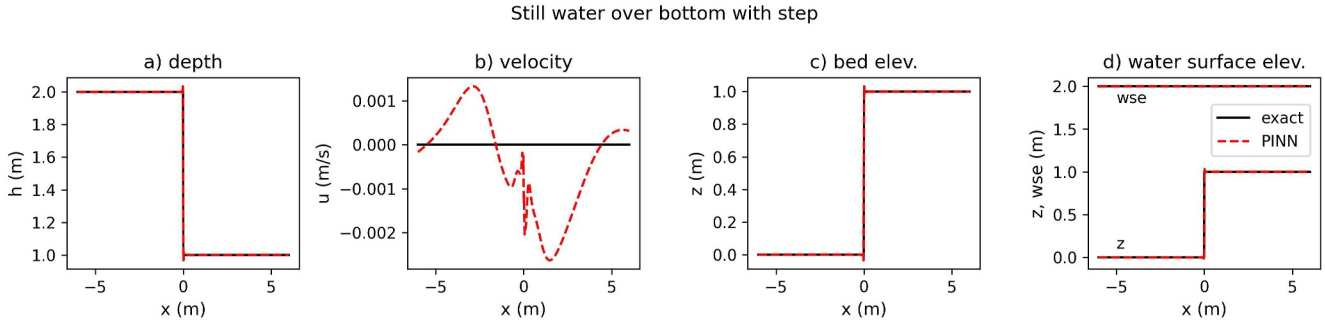

Figure 4. Case SW3: depth, velocity, bed, and water surface elevation profiles at the final time. Comparison between analytical solution and PINN-augmSWE results.

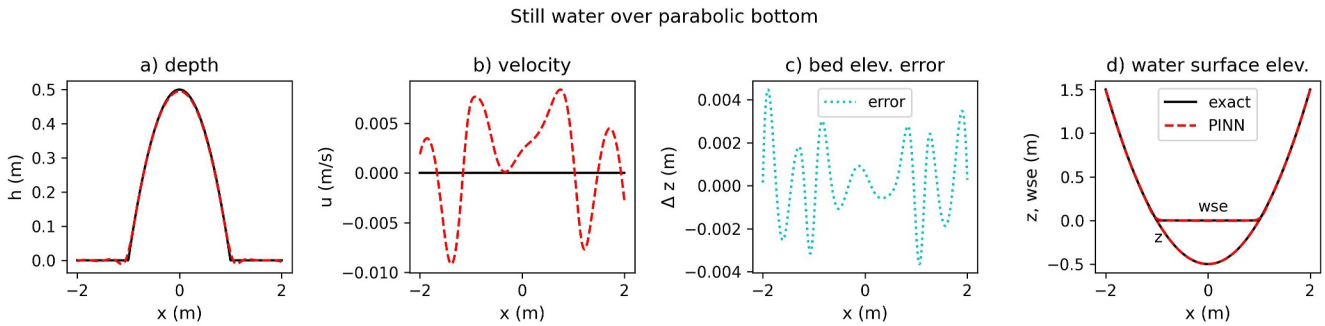

Figure 5. Case SW4: depth, velocity, bed elevation error, and bed and water surface elevation profiles at the final time. Comparison between analytical solution and PINN-augmSWE results.

test SW2 (1E−3), while the RMSE values are slightly larger, but overall the PINN model is able to capture the steady-state condition with RMSE in the order of 1E−2.

3.3. Unsteady Flow Test

For unsteady flow tests, profiles are extracted at different times ($\bar{t} = 0, 0.25t_f, 0.5t_f, 0.75t_f, t_f$) to show the temporal evolution of the phenomenon. Error metrics (MAE and RMSE) computed for the points along these profiles for all tests are reported in Table 3, together with their mean value, while the graphical representation is limited to the initial, middle and final times, except for case UF4 for which all five profiles are depicted.

The first two tests (UF1 and UF2) concern flow on horizontal bottom. Results of case UF1 (propagation of a small amplitude wave) are shown in Figure 7. Initial conditions ($t = 0$) are captured accurately, with RMSEs between the orders of 1E−3 and 1E−4 (Table 3). PINNs capture the subsequent temporal evolution of the phenomenon, characterized by two symmetrical waves propagating upstream and downstream with opposite velocities. The solution accuracy for depth and velocity seems to slightly degrade as time increases (Table 3), but the symmetry of the problem is well predicted.

The dam-break test (case UF2), whose results are depicted in Figure 8, starts from a condition with a sharp discontinuity in the depth profile at $t = 0$, which is slightly smoothed in the PINNs' predictions. Moreover, the largest errors in the initial velocity and bed elevation profiles can be observed at the same location. For $t > 0$, a rarefaction wave moves upstream, and a shock wave propagates downstream. This behavior is predicted correctly by the PINN model, although the wave profiles are smoother than the analytical solution. Differently from case UF1, however, the solution accuracy (MAE, RMSE) does not change in time, remaining in the order of 1E−2 for depth and close to 1E−1 for velocity (Table 3), while the bed elevation is captured more accurately (MAE in the order of 1E−4).

Case UF3 (dam-break over a bottom step) is the first unsteady problem with a non-horizontal bottom, and its results are shown in Figure 9. The PINN-augmSWE model correctly captures the main features of the solution, that is, not only the rarefaction wave propagating upstream and the shock wave moving downstream, but also the

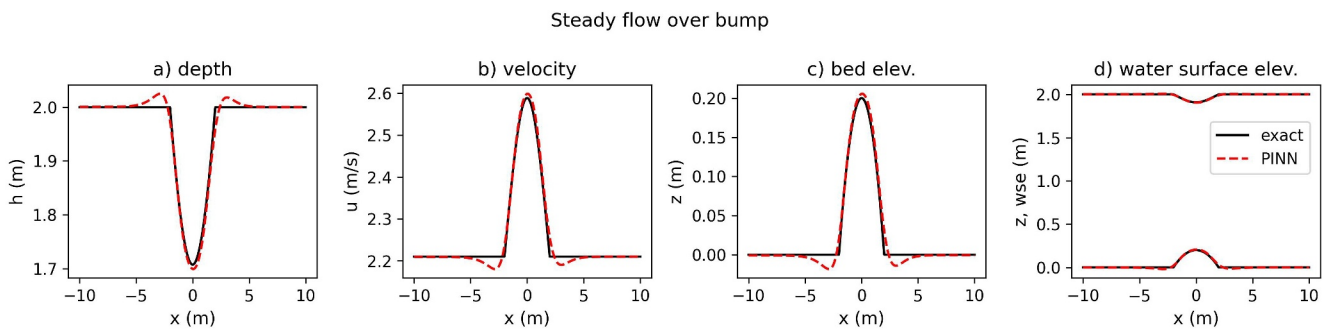

Figure 6. Case SF1: depth, velocity, bed, and water surface elevation profiles at the final time. Comparison between analytical solution and PINN-augmSWE results.

Table 3
Unsteady-Flow Tests

Test name	Time	MAE h (m)	RMSE h (m)	MAE u (m)	RMSE u (m)	MAE z (m)	RMSE z (m)
UF1	0	2.0E−03	3.0E−03	4.6E−04	5.9E−04	9.8E−05	1.4E−04
	0.25	1.8E−03	2.3E−03	8.3E−03	1.2E−02	8.5E−05	1.2E−04
	0.5	3.1E−03	4.4E−03	1.1E−02	1.5E−02	6.8E−05	9.6E−05
	0.75	3.7E−03	5.7E−03	1.4E−02	2.0E−02	6.6E−05	9.2E−05
	1	4.9E−03	7.2E−03	1.5E−02	2.4E−02	6.3E−05	8.8E−05
	Mean	3.1E−03	4.5E−03	9.8E−03	1.4E−02	7.6E−05	1.1E−04
UF2	0	1.2E−02	4.0E−02	4.0E−03	1.1E−02	9.0E−04	2.1E−03
	0.25	1.7E−02	3.4E−02	4.7E−02	9.8E−02	9.4E−04	2.1E−03
	0.5	1.9E−02	3.5E−02	5.3E−02	1.0E−01	9.3E−04	2.0E−03
	0.75	1.9E−02	3.4E−02	5.2E−02	9.6E−02	8.9E−04	2.0E−03
	1	2.0E−02	3.3E−02	5.1E−02	9.0E−02	9.1E−04	2.0E−03
	Mean	1.8E−02	3.5E−02	4.2E−02	7.9E−02	9.1E−04	2.0E−03
UF3	0	1.5E−02	4.7E−02	5.0E−03	1.3E−02	1.6E−03	5.9E−03
	0.25	1.9E−02	4.0E−02	5.2E−02	1.1E−01	1.8E−03	7.2E−03
	0.5	2.2E−02	4.1E−02	5.9E−02	1.1E−01	2.0E−03	7.7E−03
	0.75	2.3E−02	4.1E−02	5.7E−02	1.1E−01	2.0E−03	8.2E−03
	1	2.3E−02	3.9E−02	5.2E−02	1.0E−01	2.1E−03	8.7E−03
	Mean	2.0E−02	4.2E−02	4.5E−02	8.9E−02	1.9E−03	7.6E−03
UF4	0	7.7E−04	1.1E−03	2.6E−04	4.3E−04	1.0E−03	1.2E−03
	0.25 T	3.6E−03	7.6E−03	4.6E−02	1.0E−01	1.7E−03	2.1E−03
	0.5 T	1.0E−02	1.6E−02	3.4E−02	5.3E−02	1.9E−03	2.4E−03
	0.75 T	6.1E−03	1.0E−02	8.5E−02	1.7E−01	2.0E−03	2.5E−03
	T	7.5E−03	1.5E−02	5.7E−02	8.5E−02	2.0E−03	2.6E−03
	Mean	5.7E−03	1.0E−02	4.5E−02	8.1E−02	1.7E−03	2.2E−03

Note. Error metrics (MAE and RMSE) of PINN-augmSWE results compared to the analytical solution, for depth, velocity and bed elevation profiles extracted at five different times and the mean value.

contact discontinuity at the step location, typical of this type of flows. However, near these solution discontinuities, PINN results are either smoothed (shock and rarefaction waves) or display local spikes (contact wave). Error metrics for depth and velocity are comparable to those obtained for case UF2 (Table 3), suggesting that the main source of inaccuracy is the highly transient nature of the phenomena, rather than the presence of a non-horizontal bottom.

Finally, Figure 10 shows the results of case UF4 (Thacker's test of an oscillating planar surface on a parabolic bottom), which also entails wetting-and-drying processes. The overall prediction of this very complex flow is correct for all variables. Depth profiles have a higher accuracy for $t = 0.25T$ and $t = 0.75T$ (when the water surface elevation is horizontal), while velocity profiles are more accurate for $t = 0.5T$ and $t = T$ (at flow reversals when velocity is null), as revealed by RMSE values reported in Table 3. Errors related to negative depths and non-null velocities in dry regions are negligible compared to errors related to the complex oscillating behavior of water in this test. For example, the MAE of water depth profiles computed separately for the dry region remains in the order of 1E−4 for the five considered times, that is, 1-2 orders of magnitude lower than the MAE for depth in the wet region. This indicates that the depth-positivity loss terms in Equation 5 are correctly constraining the solution in the dry region, allowing only minor discrepancies.

3.4. Comparison With PINNs Solving Non-Augmented SWEs

This Section provides an example of comparison between the results of PINN-augmSWEs and of a PINN model solving the (simpler) homogeneous SWEs (without the bed-slope source term). The purpose of this analysis is to

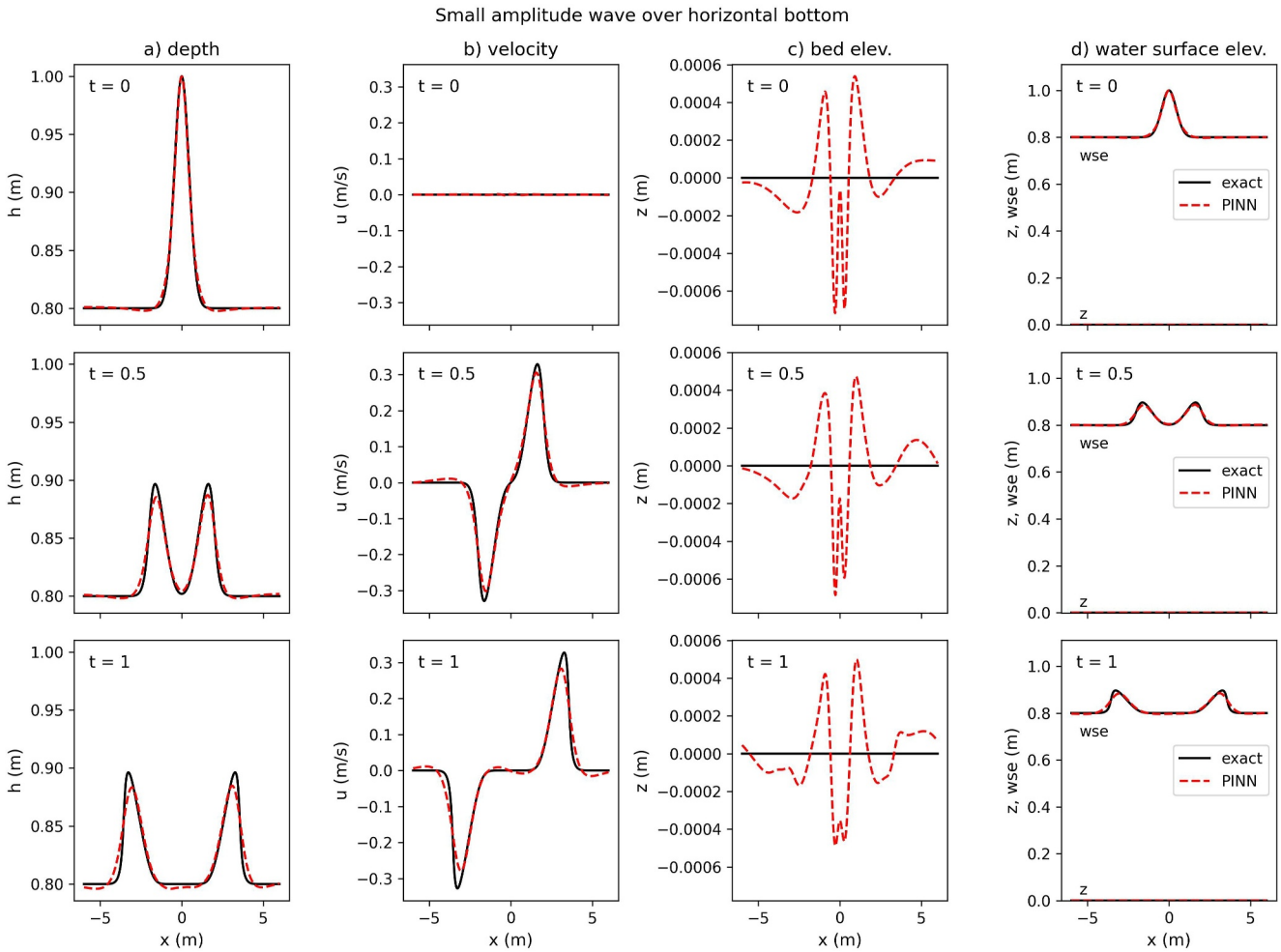

Figure 7. Case UF1: depth, velocity, bed, and water surface elevation profiles at three different times. Comparison between reference solution and PINN-augmSWE results.

evaluate whether the prediction accuracy of flow variables (depth and velocity) decreases due to the necessity of predicting the additional variable (bed elevation) when using augmented SWEs. For the comparison, only the simplest form of topography is considered (flat bottom), because in this case the adoption of the SWEs without topography and without the fictitious fixed-bed equation (Equation 1c) would be sufficient to obtain the solution. Therefore, case UF2 (dam-break over flat bottom) is selected, and for this simple test case a second PINN model to solve non-augmented SWEs is setup, in which only depth and velocity are predicted as output variables, and its solution is compared with the results previously obtained for the augmented SWEs. For the sake of simplicity, the same hyperparameters of Section 2.4 are used (except w_{z_fixed} and w_z).

Figure 11 shows the depth and velocity profiles at the final time obtained from the two PINN models (augmented and non-augmented SWEs), which suggest that they are characterized by a similar accuracy in predicting the flow variables. The RMSE values of the PINN model without topography are 2.7E−2 and 7.3E−2 for depth and velocity profiles, respectively, which are close to the accuracy obtained for the same profiles when the augmented SWEs are solved (3.3E−2 and 9.0E−2, see Table 3).

3.5. Comparison With Finite-Volume Methods

In this Section, the accuracy of PINNs is compared to the one guaranteed by a FV numerical method for one of the unsteady test cases (UF2). To this purpose, the simulation of test UF2 is performed using a first-order FV scheme in which fluxes are computed using the HLL approximate Riemann solver (Toro, 2001), and the spatial domain is discretized with either 100 or 1,000 computational cells.

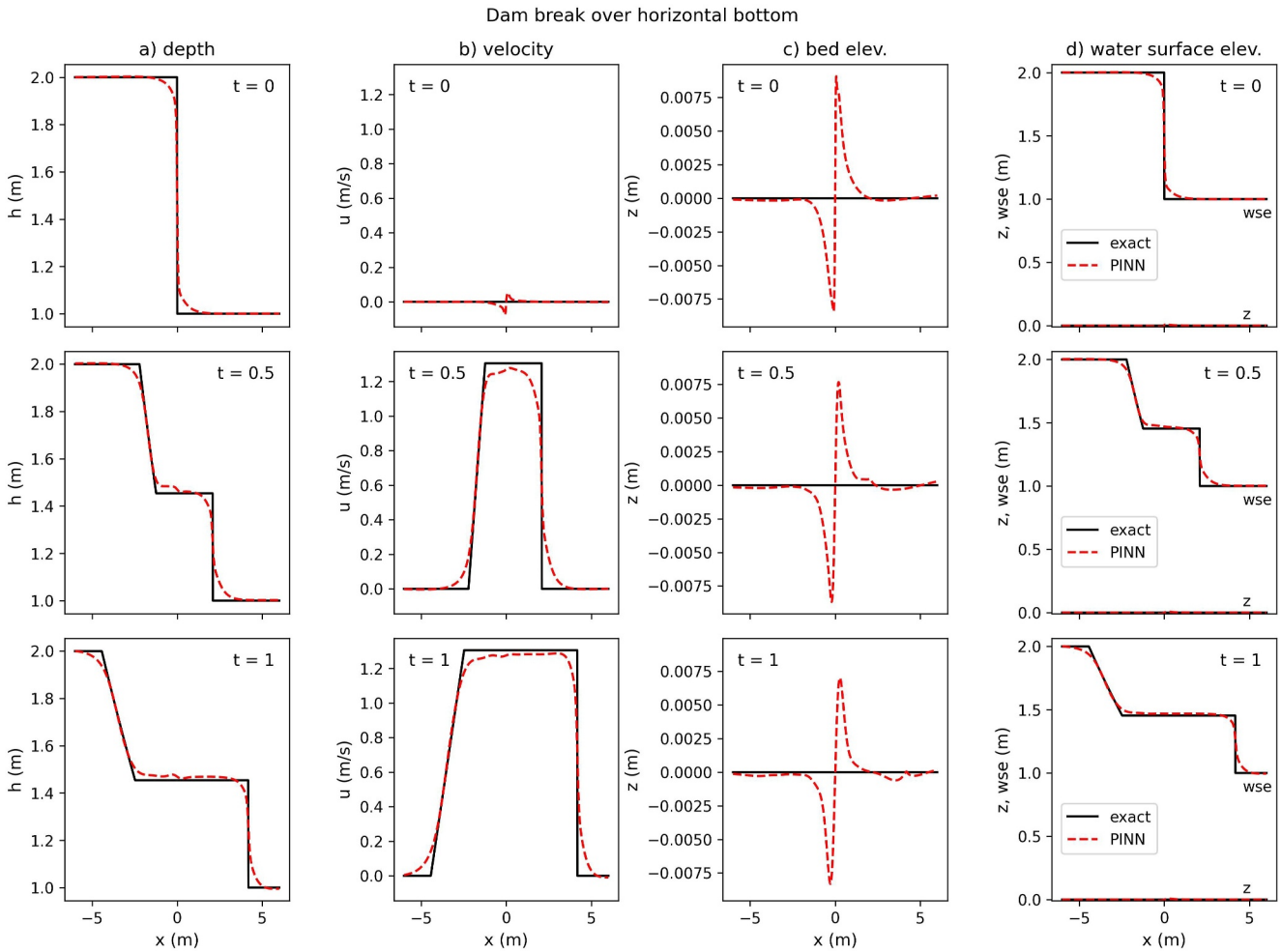

Figure 8. Case UF2: depth, velocity, bed, and water surface elevation profiles at three different times. Comparison between analytical solution and PINN-augmSWE results.

Figure 12 compares the numerical results of these simulations with the analytical solution and with the PINN model's predictions at the final time. With the coarser resolution (100 cells), the RMSE values for depth and velocity (around 3E−2 and 8E−2, respectively) are similar to the PINN ones, although the FV solution appears slightly more accurate at the intermediate state and near the shock. When the resolution is increased, FV results are much closer to the analytical solution (RMSE values around 9E−3 and 2E−2 for depth and velocity, respectively), and the accuracy could have been further improved by using a higher-order scheme. Finally, as regards the order of magnitude of computational times, the FV simulation with 100 cells took only 0.05 s to run, while the one with 1,000 cells took 4.3 s using a serial implementation on CPU (Intel® Xeon® W-2223 Processor).

3.6. Example of Data Assimilation

In this Section, the PINN model's suitability to include sparse observations is tested considering case UF1 (small perturbation test case). Results are compared for two model setups (base "A" and with halved collocation points "B," see Section 2.4) for PINNs trained both without data assimilation (NoOBS_A and NoOBS_B) and with data assimilation (OBS_A and OBS_B). Notice that case NoOBS_A corresponds to case UF1 in Section 3.3.

Table 4 summarizes the accuracy of PINN-augmSWE for these four cases, quantified by means of the RMSE values for the profiles of the three variables at the final time, which are also represented and compared in Figure 13 (only depth and velocity, since bed elevation is characterized by very low errors). Results suggest that the integration of sparse observations is useful to increase the accuracy in both configurations A and B. When the training is performed with a number of collocation points that is already large enough to obtain a fair accuracy, a

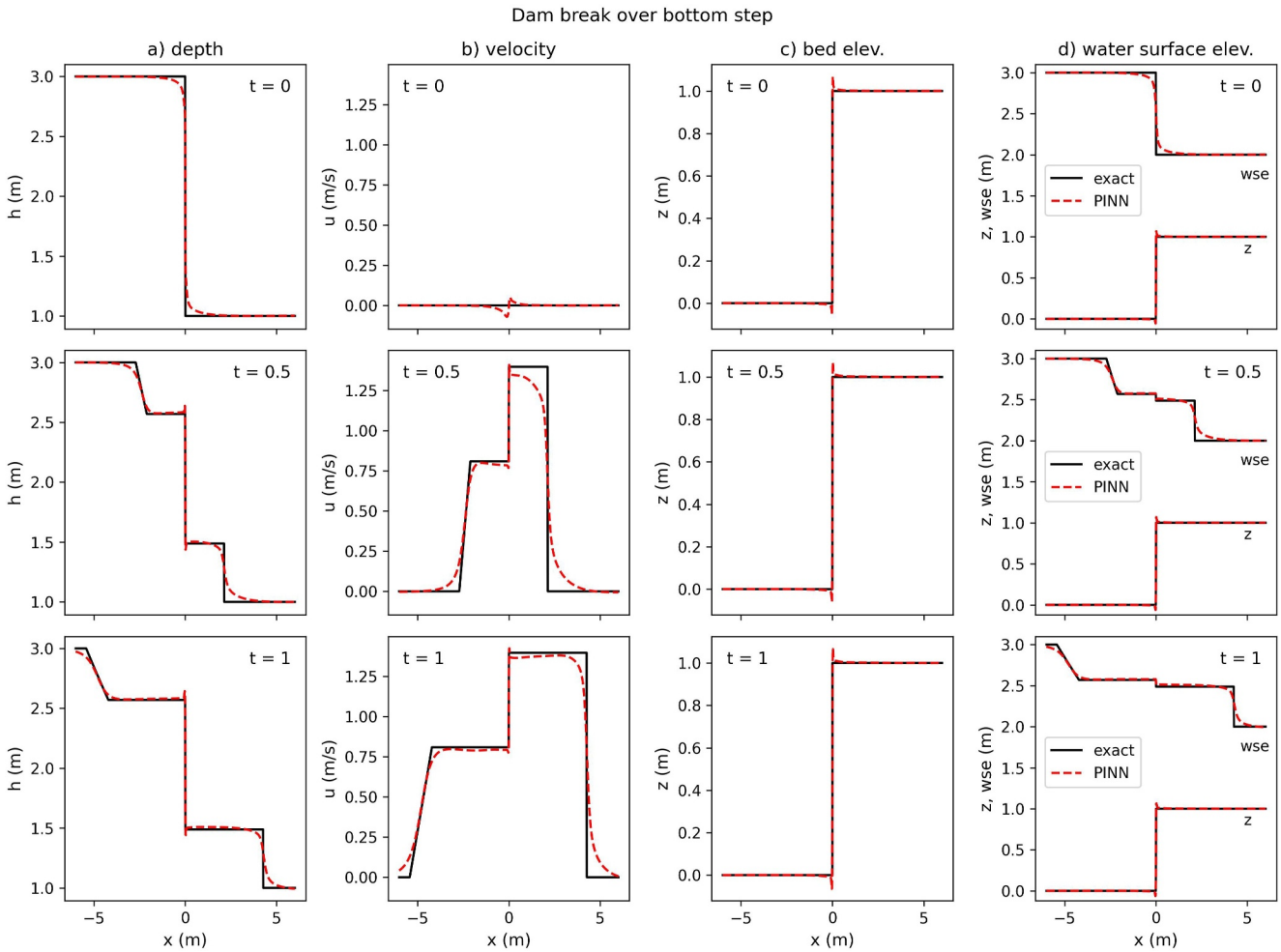

Figure 9. Case UF3: depth, velocity, bed, and water surface elevation profiles at three different times. Comparison between analytical solution and PINN-augmSWE results.

slight decrease of RMSE values is observed (see NoOBS_A vs. OBS_A), while a more evident improvement in accuracy is obtained if training is performed with halved collocation points (see OBS_B vs. NoOBS_B).

Table 4 also reports the dimensionless values of training times for all these cases (w.r.t. NoOBS_A, which takes 8.9 min to train). The integration of sparse observations has almost no influence on the training time compared to the case without data assimilation, when the same hyperparameter configuration is used. Although limited to this specific test case, the inclusion of sparse observations revealed to be useful to obtain more accurate results using a reduced number of collocation points, which entailed a shorter computational time for the training process.

4. Discussion

In this work, a strategy of solving the augmented SWEs to model free-surface flow over non-flat topography is proposed. For PINNs, this choice leads to the estimation of one additional output variable (bed elevation z), which does not change in time and is actually known a priori. Despite introducing seemingly unnecessary complexity to the model and not being strictly required by the problem, this strategy is used as a workaround to leverage automatic differentiation to compute the bed slope ($\partial z/\partial x$), which appears in the source term of the momentum equation and is therefore fundamental to compute the PDE residuals. The proposed approach allows the PINN model to embed its computation in the training process, during which PINNs learn how the bed elevation is spatially distributed from the IC constraint, and that the bed is fixed thanks to the additional constraint of null time

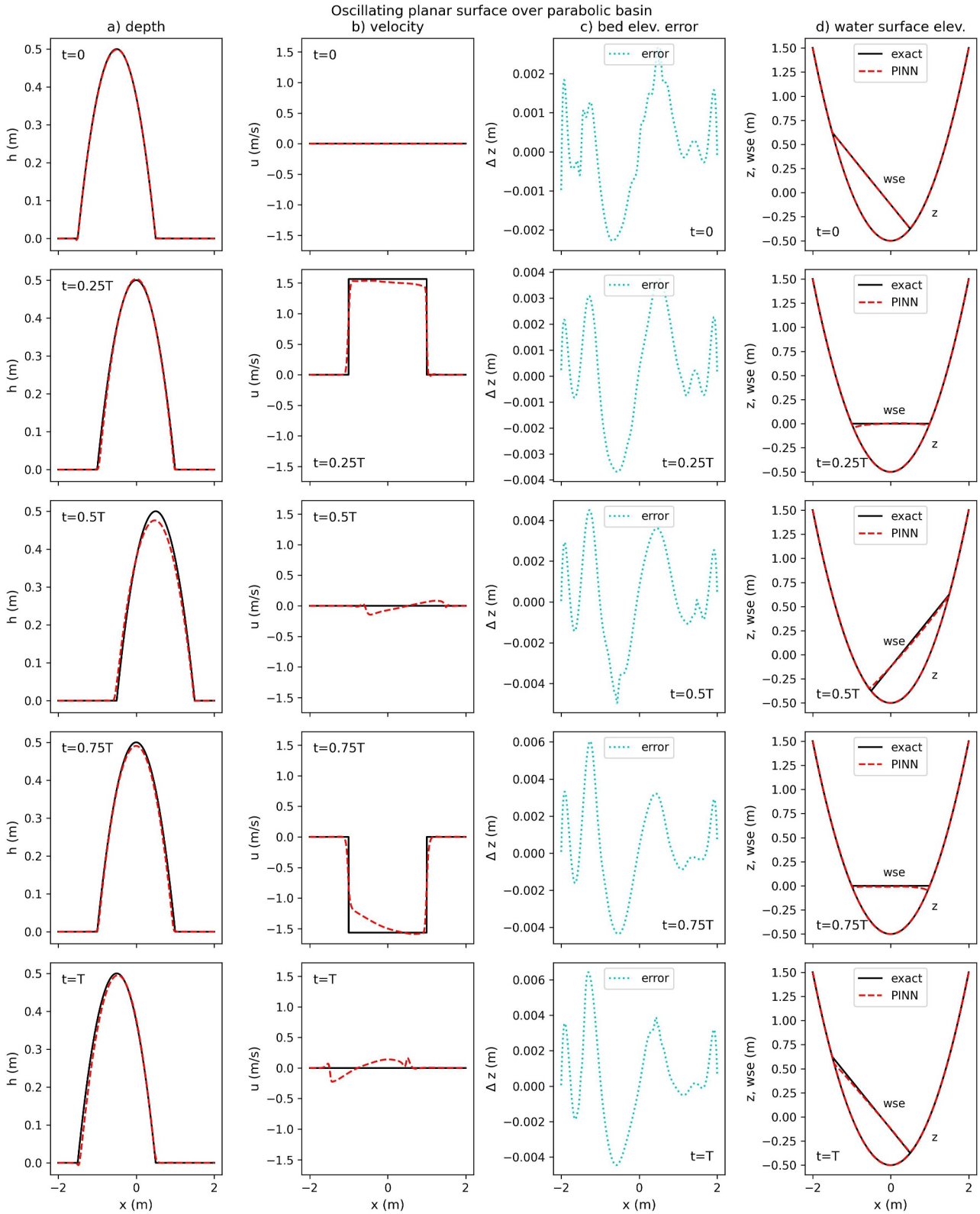

Figure 10. Case UF4: depth, velocity, bed elevation error, and bed and water surface elevation profiles at different times. Comparison between analytical solution and PINN-augmSWE results.

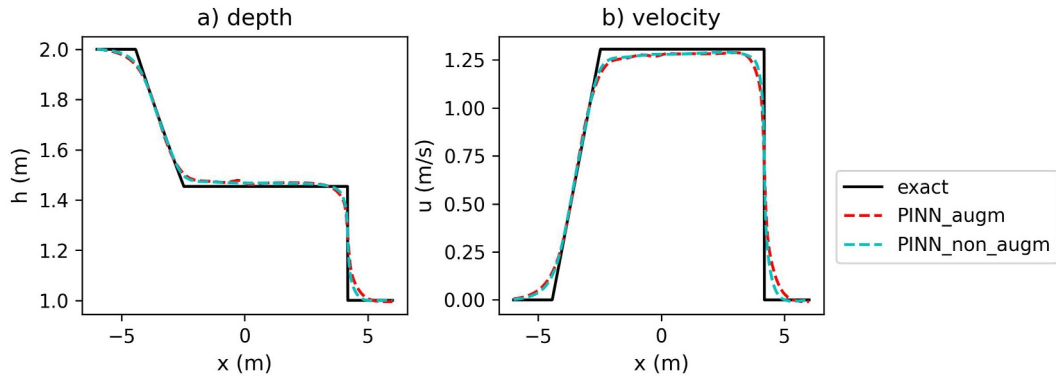

Figure 11. Case UF2: depth and velocity profiles at the final time. Comparison between analytical solution and PINN results with augmented SWEs (with topography) and with standard SWEs (no topography).

derivative of z. Results here presented show that the bed elevation is actually learned with good accuracy by the PINN-augmSWE model for different topographies.

The main advantage of this approach is the use of automatic differentiation to approximate bed slopes within the same network used to compute other variables. In this way, the proposed approach avoids training a separate network to obtain the bed elevation gradients, which would require additional hyperparameters optimization. Moreover, compared to another simple approach to compute gradients, that is, resorting to a pre-processing step with standard numerical techniques (e.g., finite differences, as done by Qi et al. (2024)) and then solving the non-augmented SWEs with PINNs, the strategy here proposed does not require the definition of a mesh resolution to approximate gradients. Both these approaches can be considered alternative strategies to deal with topography in PINN models for the SWEs.

However, some disadvantages of the approach proposed in this paper should also be mentioned. First, the necessity of predicting the topography may induce additional errors in the solution for other variables. Moreover, the definition of an additional hyperparameter (i.e., the loss weight for the fixed-bed condition) is required. These issues are discussed in the following.

In order to check if the addition of the bed elevation as an output variable has an influence on the accuracy of the other flow variables prediction, an additional comparison between two PINN models solving augmented and non-augmented SWEs was performed considering the flat-bottom test UF2 (see Section 3.4). Overall, this analysis shows that the use of augmented SWEs to deal with topography, despite their larger complexity, does not significantly reduce the accuracy obtainable from PINN models compared to standard SWEs. Additionally, in Section 3.3 it was shown that the accuracy obtained for cases UF2 and UF3 as regards the flow variables was similar, indicating that the inclusion of topography using the augmentation of SWEs had a similar effect on the solution of flows over flat beds and over a bottom step. Even if limited to simple test cases, these results suggest

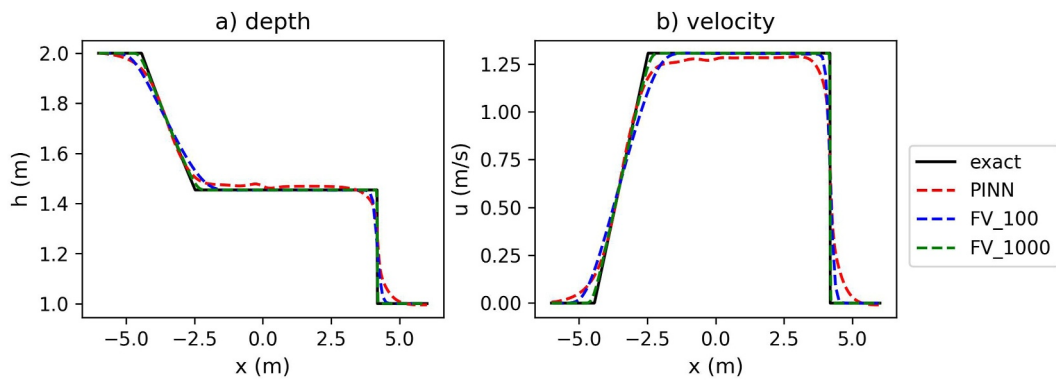

Figure 12. Case UF2: depth and velocity profiles at the final time. Comparison between analytical solution, PINN-augmSWE results, and FV results with 100 and 1,000 computational cells.

Table 4
Case UF1 With/Without Data Assimilation

Test name	RMSE h (m)	RMSE u (m)	RMSE z (m)	Training time (−)
NoOBS_A	7.2E−03	2.4E−02	8.8E−05	1
NoOBS_B	1.5E−02	4.8E−02	9.9E−04	0.48
OBS_A	5.0E−03	1.8E−02	1.5E− 04	1.01
OBS_B	9.6E−03	3.2E−02	8.7E−04	0.45

Note. Error metrics (RMSE) of PINN-augmSWE results compared to the analytical solution, for depth, velocity and bed elevation profiles extracted at the final time (t = 1 s). Training times (made non-dimensional w.r.t. test "NoOBS_A," which takes 8.9 min) are also reported.

that this approach can be considered an alternative viable way to deal with flows over non-flat topography when using PINNs as a solution method. Comparisons with alternative approaches for test cases with non-flat bottom, which could lead to more generalizable conclusions, are left to future developments.

The results presented in the previous Section, however, showed that, for non-horizontal bottom, some discrepancies from the exact solution could be observed especially near abrupt discontinuities in the topography (e.g., just upstream/downstream of the bump for cases SW1 and SF1, and near the step for case SW3). Opposite trends were observed in the prediction of the water depth and bed elevation (one is underestimated and the other is overestimated, or vice versa), while their summation is less affected by these local errors. However, predicting the water surface elevation ($h + z$) as an output variable directly would not lead to better results. Indeed, it was verified that, if the SWEs are re-formulated using the water surface elevation instead of the water depth as an output variable (as in Vacondio et al., 2014), for case UF3 (dam-break over bottom step) the PINN model provides very similar results to those shown in Figure 9 (not shown here for the sake of conciseness). Quantitatively, the RMSE values at the final time for this PINN configuration with $h + z$ as an output variable are very close to what reported in Table 3, with values of 4E−2, 1E−1, and 9E−3 for depth, velocity, and bed elevation profiles, respectively. Therefore, either water surface elevation or depth can be used alternatively as output variables, depending on the type of data that are required for each application and/or that are possibly available for data assimilation.

A challenging issue about setting up a PINN model is the definition of the hyperparameters for training. While for traditional solution methods there is usually a numerical or physical reason behind the setup of model parameters (e.g., spatial resolution, time step size, etc.), for PINNs well-defined criteria are not available (Chuang & Barba, 2022), and users must rely mostly on trial-and-error or grid-search procedures to identify the set of hyperparameters that provide the most accurate results. As mentioned in Section 2.4, the grid-search approach was used in this paper to define the number of neurons and layers in the DNN and the number of collocation points for test UF2. Analyzing the outcomes of this procedure (Figure S1 in Supporting Information S1) reveals that, for this test, the RMSE values for selected profiles of depth and velocity can increase up to 30%–40% compared to the "best" configuration if different hyperparameters are selected, and that increasing the number of layers and collocation points was observed to be slightly more beneficial than increasing the number of neurons. However, these results were case-specific, and additional fine-tuning was necessary for other tests. When dealing specifically with augmented SWEs, it should be considered that additional hyperparameters concerning the loss weight of topographical constraints must be defined, and the accuracy of results can be influenced by their value (Figure S2 in Supporting Information S1); thus, special attention should be dedicated to the selection of these parameters. In particular, it is here suggested that a value 1-2 orders of magnitude larger than the PDE loss weight should be adopted for w_z and w_{z_fixed}, in order to allow PINNs to correctly learn the spatial distribution of bed elevations and improve the prediction accuracy of other variables.

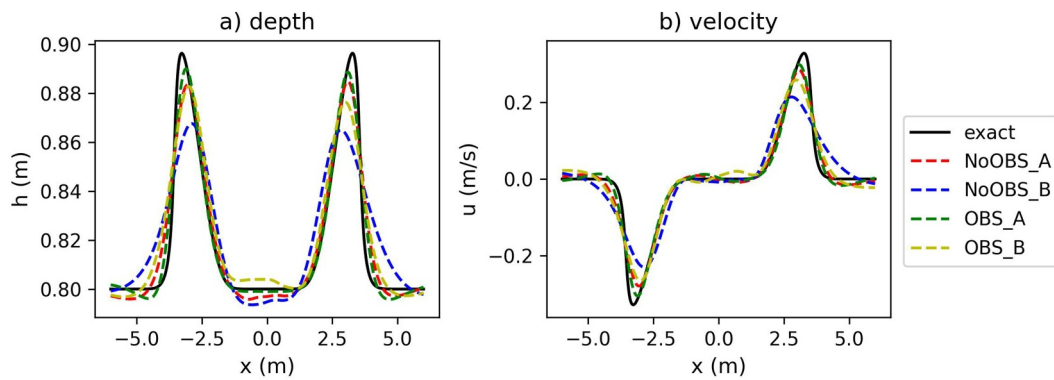

Figure 13. Case UF1: profiles of depth and velocity at t = 1 s. Comparison between reference solution and PINN-augmSWE results with and without data assimilation (for configurations A and B).

Some authors in the recent literature have suggested that PINNs may not be expected to reach the same accuracy of standard numerical methods (Mao et al., 2020). The results shown in this work, even if limited to a specific test case (UF2), confirm this issue as regards the comparison between PINN and FV (Section 3.5). Indeed, for this test, the PINN solution is even slightly less accurate than a FV solution obtained with a rather coarse resolution. Different strategies to improve the accuracy of PINNs are currently investigated in the literature. One notable example is the use of a non-random distribution of collocation points, that is, a distribution in which points are clustered more densely near high-gradient regions in the space-time domain, while a coarser distribution is used in smoother regions (Mao et al., 2020). This strategy may be particularly helpful to increase the accuracy near shock waves. However, in this preliminary study, specific analyses for augmented SWEs were not performed in this regard (e.g., the identification of the most effective type of clustering criterion that should be used near topographical discontinuities) and are left to future works. Note that alternative strategies for defining collocation points, such as adopting a uniform distribution in the space-time domain (Qi et al., 2024), are also possible and may foster the use of different types of DL architectures for PINNs, such as Convolutional Neural Networks (Qi et al., 2024), to obtain more accurate results.

Another drawback of PINNs, also identified in previous works (Chuang & Barba, 2022; Jagtap et al., 2020), is the fact that usually they are not competitive with standard numerical models from a computational point of view. In this work, training times were in the order of a few minutes, despite the exploitation of a powerful GPU device, while the FV simulations took only a handful of seconds even using a serial implementation. Clearly, runtimes depend on many model parameters (e.g., the number of neurons, layers, and collocation points for PINNs; the grid resolution and numerical scheme for FV), not to mention the influence of the hardware used to perform computations and of the different parallelization strategies that can be adopted in the code implementation. A rigorous comparison between PINNs and conventional numerical methods, such as the evaluation of the speed-accuracy tradeoff suggested in some works (e.g., Chuang & Barba, 2022; Qi et al., 2024), is beyond the scope of this work and is left to future analyses. Here, the specific test performed (UF2) shows that, compared to PINNs, the FV method may yield more accurate results in a shorter time and without requiring burdensome hyperparameter optimization. When dealing with standard forward problems, further research is still needed for PINNs to become competitive with state-of-the-art numerical methods such as FV.

Despite these possible limitations, PINNs are actively investigated in different scientific fields due to other advantages, such as their suitability to model inverse and parametrized problems and to exploit data assimilation of sparse observations to enhance the solution accuracy, as discussed in the Introduction. In this work, these types of problems for which PINNs can be more suitable than traditional numerical methods were not treated in detail, because the primary objective was to test the PINNs' potential applicability to solve SWEs with topography, and more in-depth analyses are left to future works. However, in order to show an example of the potential of PINNs in this context, a problem with data assimilation was presented (Section 3.6). PINNs were trained with/without including sparse observations for test UF1 (small perturbation wave), and the comparison of results showed that the PINN model was capable of easily integrating the observations into the training process with almost no increase in the computational time. For this specific test, the results obtained with data assimilation were more accurate than the ones obtained from physical constraints only, especially when a lower number of collocation points was used. The assimilation of only one type of data was analyzed here, that is, the "measured" depth values at 10 random locations and at a specified time (obtained from the FV reference solution for this synthetic test case), which in a real application could correspond to some observations obtained from an aerial/satellite image. Other kinds of observed data can be available in the practice and can be used for data assimilation, for example, time series of recorded water levels at specific locations (gauge stations) or data from post-flood surveys (high watermarks), while in situ velocity measurements are less common, except at the laboratory scale.

It should be underlined that, before moving to realistic test cases, additional analyses concerning problems with friction and time-variable boundary conditions should be performed. Future developments include the extension of PINN-SWEs to two-dimensional flows and its application to realistic topographies, as recently done by Qi et al. (2024) for a river flood case, and to the case of PINNs being used for sub-grid solution refinement. Moreover, the equation expressing the fixed-bed condition could be substituted by the Exner equation, which governs the bed elevation changes of movable beds in combination with some sediment transport closure formula. This could expand the potential applications of the PINN-based models to movable-bed problems. Notice that this latter possible extension also represents an advantage of the proposed approach for dealing with topography compared with alternative strategies (i.e., pre-processing of bed elevation gradients).

5. Conclusions

This work aimed at assessing the applicability of PINNs as a solution method for free-surface flow problems over non-flat bottom. In particular, the strategy of using PINNs to solve the augmented system of SWEs with topography was tested as an alternative way to deal with spatially variable topography during the training process by leveraging automatic differentiation to compute the bed slope source term. Results showed that this approach guarantees sufficiently accurate predictions for different synthetic test cases involving complex topographies. First, by considering still water test cases with different topographies, it was shown that, even if the preservation of the still water condition was not perfectly satisfied, PINNs were able to predict an accurate enough solution for these tests. Then, it was checked that the PINNs solution led to the correct steady-flow state even in the presence of a non-flat bottom. Finally, the results of the unsteady tests revealed that PINNs were capable of simulating highly unsteady flows over horizontal and non-horizontal bottom with an acceptable accuracy. Moreover, the suitability of PINNs for data assimilation problems was shown for one test case.

Despite the fact that some limitations can be identified, addressing the specific challenges of solving SWEs with PINNs (e.g., topography) is worthy of investigation, as it may contribute to enhance PINNs' applicability to realistic cases. This work represent a step in this direction, but further research is needed for PINNs to become competitive with state-of-the-art numerical methods such as FV in terms of efficiency and accuracy.

Data Availability Statement

The source code used to perform all the tests reported in the paper is available on Zenodo (Dazzi, 2024).

Acknowledgments

The Author wishes to thank Prof. Renato Vacondio for his valuable comments on the manuscript and Dr. Niki Andreas Loppi for the introduction to Modulus. The Author is also grateful to the anonymous Reviewers that provided useful suggestions to improve the manuscript. This research was granted by University of Parma through the action "Bando di Ateneo 2022 per la ricerca" co-funded by MUR-Italian Ministry of Universities and Research—D.M. 737/2021—PNR—PNRR—NextGenerationEU. The Author also acknowledges financial support from the PNRR MUR project ECS_00000033_ECOSISTER (Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5—Call for tender No. 3277 of 30/12/2021 of Italian Ministry of University and Research funded by the European Union—NextGenerationEU). This research benefits from the HPC facility of the University of Parma. Finally, the Author acknowledges the CINECA award (project AMNERIS) under the ISCRA initiative, for the availability of high-performance computing resources and support.

References

Annis, A., Nardi, F., & Castelli, F. (2022). Simultaneous assimilation of water levels from river gauges and satellite flood maps for near-real-time flood mapping. *Hydrology and Earth System Sciences*, *26*(4), 1019–1041. https://doi.org/10.5194/hess-26-1019-2022

Aureli, F., Maranzoni, A., Mignosa, P., & Ziveri, C. (2008). A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Advances in Water Resources*, *31*(7), 962–974. https://doi.org/10.1016/j.advwatres.2008.03.005

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, *18*, 1–43.

Bernetti, R., Titarev, V. A., & Toro, E. F. (2008). Exact solution of the Riemann problem for the shallow water equations with discontinuous bottom geometry. *Journal of Computational Physics*, *227*(6), 3212–3243. https://doi.org/10.1016/j.jcp.2007.11.033

Bihlo, A., & Popovych, R. O. (2022). Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, *456*, 111024. https://doi.org/10.1016/j.jcp.2022.111024

Bomers, A., & Hulscher, S. J. (2023). Neural networks for fast fluvial flood predictions: Too good to be true? *River Research and Applications*, *39*(8), 1652–1658. https://doi.org/10.1002/rra.4144

Cedillo, S., Núñez, A. G., Sánchez-Cordero, E., Timbe, L., Samaniego, E., & Alvarado, A. (2022). Physics-Informed Neural Network water surface predictability for 1D steady-state open channel cases with different flow types and complex bed profile shapes. *Advanced Modeling and Simulation in Engineering Sciences*, *9*(1), 1–23. https://doi.org/10.1186/s40323-022-00226-8

Chuang, P. Y., & Barba, L. A. (2022). Experience report of physics-informed neural networks in fluid simulations: Pitfalls and frustration. *arXiv preprint, arXiv:2205.14249*, 28–36. https://doi.org/10.25080/majora-212e5952-005

Cozzolino, L., Della Morte, R., Covelli, C., Del Giudice, G., & Pianese, D. (2011). Numerical solution of the discontinuous-bottom shallow-water equations with hydrostatic pressure distribution at the step. *Advances in Water Resources*, *34*(11), 1413–1426. https://doi.org/10.1016/j.advwatres.2011.07.009

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, *92*(3), 88. https://doi.org/10.1007/s10915-022-01939-z

Dazzi, S. (2024). PINNs for the 1D augmented system of SWEs [Code]. *Zenodo*. https://doi.org/10.5281/zenodo.10625498

Dazzi, S., Shustikova, I., Domeneghetti, A., Castellarin, A., & Vacondio, R. (2021). Comparison of two modelling strategies for 2D large-scale flood simulations. *Environmental Modelling & Software*, *146*, 105225. https://doi.org/10.1016/j.envsoft.2021.105225

Delestre, O., Lucas, C., Ksinant, P. A., Darboux, F., Laguerre, C., Vo, T. N. T., et al. (2013). SWASHES: A compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, *72*(3), 269–300. https://doi.org/10.1002/fld.3741

Dumbser, M., & Balsara, D. S. (2016). A new efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative hyperbolic systems. *Journal of Computational Physics*, *304*, 275–319. https://doi.org/10.1016/j.jcp.2015.10.014

Feng, D., Tan, Z., & He, Q. (2023). Physics-informed neural networks of the Saint-Venant equations for downscaling a large-scale river model. *Water Resources Research*, *59*(2), e2022WR033168. https://doi.org/10.1029/2022wr033168

Ferrari, A., D'Oria, M., Vacondio, R., Dal Palù, A., Mignosa, P., & Tanda, M. G. (2018). Discharge hydrograph estimation at upstream-ungauged sections by coupling a Bayesian methodology and a 2-D GPU shallow water model. *Hydrology and Earth System Sciences*, *22*(10), 5299–5316. https://doi.org/10.5194/hess-22-5299-2018

Ferrari, A., Viero, D. P., Vacondio, R., Defina, A., & Mignosa, P. (2019). Flood inundation modeling in urbanized areas: A mesh-independent porosity approach with anisotropic friction. *Advances in Water Resources*, *125*, 98–113. https://doi.org/10.1016/j.advwatres.2019.01.010

Haykin, S. O. (2009). *Neural networks and learning machines* (3rd ed.). Pearson.

Hennigh, O., Narasimhan, S., Nabian, M. A., Subramaniam, A., Tangsali, K., Fang, Z., et al. (2021). NVIDIA SimNet™: An AI-accelerated multi-physics simulation framework. In *International conference on computational science* (pp. 447–461). Springer International Publishing.

Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, *365*, 113028. https://doi.org/10.1016/j.cma.2020.113028

Jagtap, A. D., Mao, Z., Adams, N., & Karniadakis, G. E. (2022). Physics-informed neural networks for inverse problems in supersonic flows. *Journal of Computational Physics*, *466*, 111402. https://doi.org/10.1016/j.jcp.2022.111402

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

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

Leiteritz, R., Hurler, M., & Pflüger, D. (2021). Learning free-surface flow with physics-informed neural networks. In *2021 20th IEEE international conference on machine learning and applications (ICMLA)* (pp. 1668–1673).

Li, C., Han, Z., Li, Y., Li, M., Wang, W., Dou, J., et al. (2023). Physical information-fused deep learning model ensembled with a subregion-specific sampling method for predicting flood dynamics. *Journal of Hydrology*, *620*, 129465. https://doi.org/10.1016/j.jhydrol.2023.129465

Liang, Q., & Marche, F. (2009). Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources*, *32*(6), 873–884. https://doi.org/10.1016/j.advwatres.2009.02.010

Magiera, J., Ray, D., Hesthaven, J. S., & Rohde, C. (2020). Constraint-aware neural networks for Riemann problems. *Journal of Computational Physics*, *409*, 109345. https://doi.org/10.1016/j.jcp.2020.109345

Mahesh, R. B., Leandro, J., & Lin, Q. (2022). Physics informed neural network for spatial-temporal flood forecasting. In *Climate change and water security: Select proceedings of VCDRR 2021* (pp. 77–91). Springer.

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. https://doi.org/10.1016/j.cma.2019.112789

Ming, X., Liang, Q., Xia, X., Li, D., & Fowler, H. J. (2020). Real-time flood forecasting based on a high-performance 2-D hydrodynamic model and numerical weather predictions. *Water Resources Research*, *56*(7), e2019WR025583. https://doi.org/10.1029/2019wr025583

Morales-Hernández, M., Sharif, M. B., Kalyanapu, A., Ghafoor, S. K., Dullo, T. T., Gangrade, S., et al. (2021). TRITON: A multi-GPU open source 2D hydrodynamic flood model. *Environmental Modelling & Software*, *141*, 105034. https://doi.org/10.1016/j.envsoft.2021.105034

Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. *Water*, *10*(11), 1536. https://doi.org/10.3390/w10111536

Murillo, J., & García-Navarro, P. (2010). Weak solutions for partial differential equations with source terms: Application to the shallow water equations. *Journal of Computational Physics*, *229*(11), 4327–4368. https://doi.org/10.1016/j.jcp.2010.02.016

NVIDIA Corporation. (2022). Modulus user guide, release v22.09. Retrieved from https://docs.nvidia.com/deeplearning/modulus/modulus-v2209/index.html

Özgen, I., Liang, D., & Hinkelmann, R. (2016). Shallow water equations with depth-dependent anisotropic porosity for subgrid-scale topography. *Applied Mathematical Modelling*, *40*(17–18), 7447–7473. https://doi.org/10.1016/j.apm.2015.12.012

Pujol, L., Garambois, P. A., & Monnier, J. (2022). Multi-dimensional hydrological–hydraulic model with variational data assimilation for river networks and floodplains. *Geoscientific Model Development*, *15*(15), 6085–6113. https://doi.org/10.5194/gmd-15-6085-2022

Qi, X., De Almeida, G. A. M., & Maldonado, S. (2024). Physics-informed neural networks for solving flow problems modeled by the 2D Shallow Water Equations without labeled data. *Journal of Hydrology*, *636*, 131263. https://doi.org/10.1016/j.jhydrol.2024.131263

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. https://doi.org/10.1016/j.jcp.2018.10.045

Rosatti, G., & Begnudelli, L. (2010). The Riemann problem for the one-dimensional, free-surface Shallow Water Equations with a bed step: Theoretical analysis and numerical simulations. *Journal of Computational Physics*, *229*(3), 760–787. https://doi.org/10.1016/j.jcp.2009.10.010

Rougier, M. (2022). *Addition of 8 analytical solutions to the SWASHES software. [Research Report]*. Institut Denis Poisson—Université d'Orléans.

Secci, D., Godoy, V. A., & Gómez-Hernández, J. J. (2024). Physics-informed neural networks for solving transient unconfined groundwater flow. *Computers & Geosciences*, *182*, 105494. https://doi.org/10.1016/j.cageo.2023.105494

Shamkhalchian, A., & De Almeida, G. A. M. (2021). Upscaling the shallow water equations for fast flood modelling. *Journal of Hydraulic Research*, *59*(5), 739–756. https://doi.org/10.1080/00221686.2020.1818316

Strelow, E. L., Gerisch, A., Lang, J., & Pfetsch, M. E. (2023). Physics informed neural networks: A case study for gas transport problems. *Journal of Computational Physics*, *481*, 112041. https://doi.org/10.1016/j.jcp.2023.112041

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F., Dutta, D., & Kim, S. J. E. M. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, *90*, 201–216. https://doi.org/10.1016/j.envsoft.2017.01.006

Thacker, W. C. (1981). Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, *107*(-1), 499–508. https://doi.org/10.1017/s0022112081001882

Todaro, V., D'Oria, M., Tanda, M. G., & Gómez-Hernández, J. J. (2019). Ensemble smoother with multiple data assimilation for reverse flow routing. *Computers & Geosciences*, *131*, 32–40. https://doi.org/10.1016/j.cageo.2019.06.002

Toro, E. (2001). *Shock capturing methods for free surface shallow water flows*. Wiley.

Vacondio, R., Dal Palù, A., & Mignosa, P. (2014). GPU-enhanced finite volume shallow water solver for fast flood simulations. *Environmental Modelling & Software*, *57*, 60–75. https://doi.org/10.1016/j.envsoft.2014.02.003

Vadyala, S. R., Betgeri, S. N., Matthews, J. C., & Matthews, E. (2022). A review of physics-based machine learning in civil engineering. *Results in Engineering*, *13*, 100316. https://doi.org/10.1016/j.rineng.2021.100316

Zokagoa, J. M., & Soulaïmani, A. (2018). A POD-based reduced-order model for uncertainty analyses in shallow water flows. *International Journal of Computational Fluid Dynamics*, *32*(6–7), 278–292. https://doi.org/10.1080/10618562.2018.1513496