Treatment of Non-physical Solutions of the Oxygen Diffusion in Soil by Physics-informed Neural Network

MILOŠ IVANOVIĆ[1], SVETISLAV SAVOVIĆ[1], LJUBICA KUZMANOVIĆ[1], and MILAN KOVAČEVIĆ[1]

[1] Faculty of Science, University of Kragujevac, R. Domanovića 12, 34000 Kragujevac, Serbia
E-mails: mivanovic@kg.ac.rs, savovic@kg.ac.rs, ljubica.kuzmanovic@pmf.kg.ac.rs, and kovac@kg.ac.rs

Abstract. We investigate oxygen diffusion in the soil in one dimension by finite differences and the physics-informed neural network. Solving the diffusion equation by either method determines the oxygen concentration profiles inside the soil column at various times. However, while respecting specified Dirichlet and Neumann boundary conditions, the concentration profiles at certain times become negative, which is non-physical per se. We can resolve this situation in finite differences by proclaiming these negative concentration values as zero during the time-stepping scheme. In the case of PINN, we propose an innovative solution with a custom loss function, tailored to avoid such non-physical behavior. Two types of Dirichlet boundary conditions are investigated. The first is constant, and the second one periodically changes, with a period of 24 hours. We demonstrate that the PINN with a customized loss is effective and accurate. The proposed approach to circumvent non-physical solution areas demonstrates promise for application to various analogous problems.

Keywords. Oxygen diffusion; diffusion equation; physics-informed neural networks; finite difference methods

1 Introduction

The diffusion equation is widely used in environmental modeling and agriculture to describe the transport of substances such as pollutants, nutrients, heat, and moisture in soil, water, and air. Thus, in environmental modeling, this equation is used to investigate pollutant dispersion in air and water, groundwater contamination, and heat transfer in climate and ecosystem studies. Agricultural applications of this equation include soil moisture and irrigation management, nutrient transport in soil, pest, and disease spread modeling, greenhouse temperature control, etc.

The aeration of the soil greatly influences its productivity. Plant roots can respire thanks to the air-filled gaps in the soil. During respiration, plant roots take oxygen and emit carbon dioxide. For most plants, the rate at which oxygen is transported from aboveground components (leaves and stems) to the root system is insufficient to fulfill the needs for oxygen [1]. Therefore, the soil must be aerated. Inadequate aeration leads to reduced water absorption and premature withering. A significant number of plants rely on getting oxygen from the soil and releasing carbon dioxide into the surrounding air. In soil air, gases and vapors are distributed by convection and diffusion, with the last-mentioned being the primary mechanism [2]. Several factors influence the oxygen diffusion rate. It rises with decreasing soil water content or increasing suction to a certain point before dropping with additional water depletion. Oxygen diffuses from the atmosphere to plant roots via air-filled soil pores and water films that separate the root surfaces from the gas phase [3]. Compact strata within a soil profile can also affect how oxygen moves through the soil [4]. The rate at which oxygen diffuses through the soil has been suggested as a reliable indicator of soil aeration [5, 6].

The kinetic theory of gases states that soil oxygen molecules constantly move due to thermal forces. Gas moves through diffusion in the soil as a result of a concentration gradient that generates an overall shift of molecules from high to low concentration. Using Fourier transformations, Papendick and Runkles [7] created analytical models to represent transient-state oxygen diffusion in soil. Kanwar [8] devised an analytical solution to describe non-steady-state oxygen transport from the atmosphere to the soil using the Laplace transformation approach. Because of the particular initial and boundary conditions used, this analytical solution's applicability is limited. Generally, finding analytical solutions is a complex task. To accurately predict the state of oxygen diffusion in soil, rather than analytical methods, it is easier to apply numerical methods to solve diffusion equations.

Air is composed of several gases. Without any humidity, it consists of 78% nitrogen and 21% oxygen, with the remaining 1% composed of other gases such as argon, carbon dioxide, neon, and helium. In nature, air is never fully dehydrated and always contains some water vapor, varying from nearly 0 to 5% colorblue in absolute terms. As the water vapor content increases, the proportion of the other gases decreases accordingly. Also, the absolute concentration of oxygen changes with daily variations in barometric pressure. For this reason, it is important to look into how minor periodic changes in O_2 concentration in the air affect the diffusion dynamics of O_2 into the soil.

This paper examines O_2 diffusion dynamics in the soil

*For correspondence

column. Using the explicit finite difference (EFD) approach to solve the diffusion equation, we determined the concentration profiles of O_2 inside the soil for both constant and periodically varying air O_2 concentration. Due to the nature of the differential equation and boundary conditions, in regions not considered by Kalita *et al*[9], non-physical negative concentration profiles are obtained for times beyond a certain critical point.

The central topic of this research is a physics-informed neural network (PINN) for solving the oxygen diffusion problem. In an identical setup, the solution obtained by PINN closely resembles that of the finite difference method, altogether with negative concentration profiles. However, in the PINN approach, the loss function can be used as a better instrument to prevent non-physical negative concentration. We demonstrate that with a choice of custom loss function with a suitable gradient, it is possible to avoid having areas with negative concentration.

The paper is organized as follows. The problem specification is given in Section 2. We present the EFD solution to the problem in Section 3.1. The description of the solution by a PINN is given in Section 3.2. The original approach for preventing non-physical concentration values is detailed in Section 3.3. Setup of the problem, the comparison of PINN solutions with EFD solutions, and the discussion are the subject of Section 4, with the concluding remarks in Section 5.

2 Oxygen diffusion dynamics

The one-dimensional oxygen diffusion in the column of soil depicted in Figure 1 is described by the partial differential equation (PDE) [8]:

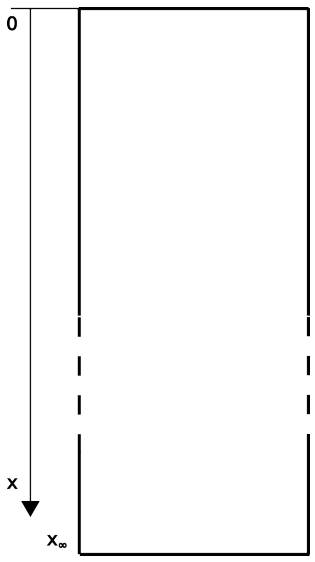

Figure 1. A schematic diagram of the soil column

$$\frac{\partial C\,(x,t)}{\partial t} = D\frac{\partial^2 C\,(x,t)}{\partial x^2} - \alpha \tag{1}$$

where $C(x,t)$ is the concentration of the diffusing substance (oxygen) in the soil at depth x, t is time, D is the diffusion coefficient of oxygen in the soil, and α is activity (the rate at which biological and chemical activities inside the soil consume oxygen). For semi-infinite homogenous soil profile, Kanwar [8] analytically solved Eq. (1) considering these initial and boundary conditions:

$$C(x,t) = C_0, \quad 0 \le x \le L; \quad t = 0 \tag{2}$$

$$\begin{aligned} C\,(x,t) &= C_0, \quad x = 0; \quad t > 0 \\ \frac{\partial C(x,t)}{\partial x} &= 0, \quad x \to \infty; \quad t > 0 \end{aligned} \tag{3}$$

where C_0 is the concentration of oxygen in the atmosphere. For simplicity, Eq. (1) assumes continuous and uniform activity α. If biological action absorbs soil oxygen, adsorption may vary with time and position [8]. In [8], an analytical solution is proposed for this problem:

$$C(x,t) = C_0 + \alpha\left[\left(t + \frac{x^2}{2D}\right)\operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right) - x\sqrt{\frac{t}{\pi D}}\exp\left(\frac{-x^2}{4Dt}\right)\right] \tag{4}$$

Papendick and Runkles [7] measured oxygen concentrations at different depths and times, and Kanwar [8] found a reasonable agreement between measured and predicted oxygen concentrations. Kalita [9] applied a finite element method to solve Eq. (1) numerically for a semi-infinite homogeneous soil profile. Determining the O_2 concentration within the soil is particularly important when considering a situation where the O_2 concentration in the air is a time-varying function, meaning that the boundary condition at $x = 0$ is as follows:

$$C(x = 0, t) = C_0\left[1 + \varepsilon \sin(2\pi t/T)\right], \tag{5}$$

where ε is the O_2 amplitude of concentration oscillation and T is the oscillation period, usually 24h, following daily rhythm. There is no analytical solution for the diffusion equation (1), hence a numerical solution is required.

We should note here that the partial differential equation (1) solved in this work is not the most appropriate model for oxygen diffusion in soil, since the concentration cannot decrease further than zero in reality. It is a good model only for large concentrations. Therefore, for small concentrations, one can use either the PINN approach presented in this work or eventually attempt to model such diffusion processes using a modified Eq. (1), where instead of reaction term $-\alpha$, one could employ a linear reaction term $-\alpha C(x,t)$.

3 Numerical methods

In this section, we will outline the solution to the problem using both the EFD and direct PINN methods. In a separate subsection, we explain the solutions to the issue of negative concentrations.

3.1 *Explicit finite-difference scheme*

Numerical methods do not have limitations of analytical methods and provide versatility, particularly for arbitrary initial and boundary conditions [10, 11]. We found that the EFD method is simpler and computationally more efficient than the implicit finite-difference method [11, 12].

The EFDM is used in this work, despite its conditional stability, since it is the most simple method. We note that there are other efficient unconditionally stable explicit methods, such as the Dufort-Frankel, leapfrog-hopscotch and shifted hopscotch, etc. (see e.g. [13]). Moreover, several numerical methods are constructed to avoid negative concentrations due to numerical errors. The most well-known is the unconditionally positive finite difference scheme [14]. There are even fourth-order methods for diffusion equation [15]. A lot of positivity-preserving algorithms are tested in [16]. Some algorithms can guarantee positivity for more complicated reaction terms, such as the logistic term [17].

The term $\left(\partial^2 C(x,t)/\partial x^2\right)$ of the diffusion equation (1) was represented using the central difference scheme:

$$\left(\frac{\partial^2 C(x,t)}{\partial x^2}\right) = \frac{C_{i+1,j} - 2C_{i,j} + C_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \tag{6}$$

and a forward difference scheme for the derivative term [12]

$$\left(\frac{\partial C(x,t)}{\partial t}\right) = \frac{C_{i,j+1} - C_{i,j}}{\Delta t} + O(\Delta t) \tag{7}$$

Equation (1) now becomes:

$$C_{i,j+1} = C_{i-1,j}\frac{D\Delta t}{\Delta x^2} + C_{i,j}\left(1 - \frac{2D\Delta t}{\Delta x^2}\right) + C_{i+1,j}\frac{D\Delta t}{\Delta x^2} - \alpha\Delta t \tag{8}$$

The indices i and j represent the discrete step lengths Δx and Δt for the coordinate x and time t, respectively. In terms of known values along the j^{th} time row, equation (6) gives $C_{i,j+1}$ at the $(i, j+1)^{\text{th}}$ mesh point. For the difference equation (8), the truncation error is $O(\Delta t, (\Delta x)^2)$. Small enough values of Δt and Δx enable for the truncation error to be minimized until the accuracy is within the error tolerance[12].

The finite difference form of the initial condition (2) for equation (1) is:

$$C_{i,0} = C_0, \quad 0 \le x \le L; \quad t = 0 \tag{9}$$

For a constant boundary condition at the soil surface, it can be expressed in the finite difference form as:

$$C_{0,j} = C_0, \quad x = 0; \quad t > 0 \tag{10}$$

For a periodic boundary condition at the soil surface, it can be written in the finite difference form as:

$$C_{0,j} = C_0\left[1 + \varepsilon \sin(2\pi t_j/T)\right], \quad x = 0; \quad t > 0 \tag{11}$$

The boundary condition at $x \to x_\infty$ in the finite difference form becomes:

$$C_{N,j} = C_{N-1,j}, \quad x \to x_\infty; \quad t > 0 \tag{12}$$

where $N = x_\infty/\Delta x$ is the grid dimension in the x direction and x_∞ is the distance in direction x at which $\partial C/\partial x = 0$. x_∞ replaces $x \to \infty$ in equation (3).

3.2 *Physics-informed neural network*

The PINN is a machine-learning technique that can approximate the solution of PDEs. PDEs with corresponding initial and boundary conditions can be expressed in a general form as:

$$u_t + \mathcal{N}[u] = 0, \quad X \in \Omega, \ t \in [0,T], \tag{13}$$

$$u(X,0) = h(X), \quad X \in \Omega, \tag{14}$$

$$u(X,t) = g(X,t), \quad X \in \Omega_g, \ t \in [0,T]. \tag{15}$$

Here $\mathcal{N}$ is a differential operator, $X \in \Omega \subseteq \mathbb{R}^d$ and $t \in \mathbb{R}$ represent spatial and temporal coordinates respectively, $\Omega \subseteq R$ is a computational domain, $\Omega_g \subseteq \Omega$ is a computational domain of the boundary conditions, $u(X,t)$ is the solution of the PDEs with initial condition $h(X)$ and boundary condition $g(X,t)$. A formulation like this can also be applied to higher-order PDEs since higher-order PDEs can be written in the form of first-order PDEs.

In the original formulation [18], PINN consists of two subnets: an approximator network and a residual network. The *approximator network* receives input (X,t), undergoes the training process, and provides an approximate solution $\widehat{u}(X,t)$ of the PDEs as an output. The approximator network trains on a grid of points, called collocation points, sampled from the simulation domain. Weights and biases of the approximator network are trainable parameters that are trained by minimizing a composite loss function with the following form:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_0 + \mathcal{L}_b, \tag{16}$$

where

$$\mathcal{L}_r = \frac{1}{N_r}\sum_{i=1}^{N_r}\left|u\left(X^i,t^i\right) + \mathcal{N}\left[u\left(X^i,t^i\right)\right]\right|^2, \tag{17}$$

$$\mathcal{L}_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}\left|u\left(X^i,t^i\right) - h^i\right|^2, \tag{18}$$

$$\mathcal{L}_b = \frac{1}{N_b}\sum_{i=1}^{N_b}\left|u\left(X^i,t^i\right) - g^i\right|^2. \tag{19}$$

Here, $\mathcal{L}_r$, $\mathcal{L}_0$, and $\mathcal{L}_b$ represent residuals of governing equations, initial and boundary conditions, respectively. Additionally, N_r, N_0, and N_b are the numbers of collocation points of the computational domain, and the initial and boundary conditions, respectively. These residuals are calculated

by a non-trainable part of the PINN model called the *residual network*. To compute the residual $\mathcal{L}_r$, PINN requires derivatives of the outputs concerning the inputs. Such computation is achieved by automatic differentiation, which relies on combining derivatives of the constituent operations by the chain rule giving the derivative of the overall composition. This technique is a key enabler for the development of PINNs and it is the key element that differentiates PINNs from similar efforts in the early 90s [19, 20], which relied on the manual derivation of back-propagation rules. Nowadays, automatic differentiation capabilities are well-implemented in most deep learning frameworks such as TensorFlow [21] and PyTorch [22], and they allow us to avoid tedious derivations or numerical discretization while computing derivatives of all orders in space-time.

A schematic of the PINN is demonstrated in Figure 2 in which a simple PDE $\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} = 0$ is used as an example. As shown in Figure 2, the approximator network is used to approximate the solution $u(X, t)$ which then goes to the residual network to calculate the residual loss $\mathcal{L}_r$, boundary condition loss $\mathcal{L}_b$, and initial condition loss $\mathcal{L}_0$. The weights and biases of the approximator network are trained using a custom loss function consisting of residuals $\mathcal{L}_r$, $\mathcal{L}_0$, and $\mathcal{L}_b$ through a gradient-descent technique based on the back-propagation.

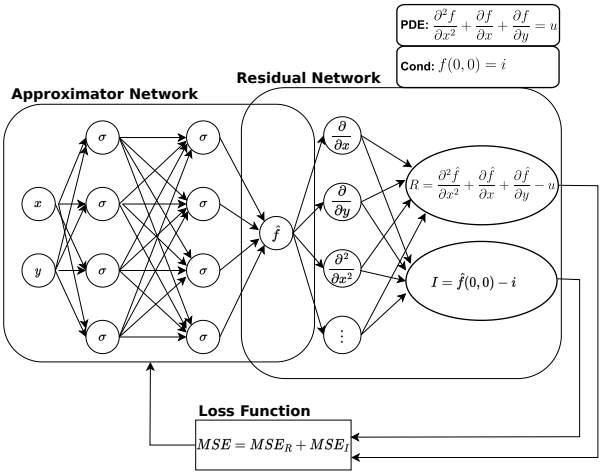

Figure 2. The architecture of the PINN and the standard training loop of PINN that is constructed for solving a simple PDE, where *PDE* and *Cond* denote governing equations, while R and I represent their residuals. The approximator network is subjected to a training process and provides an approximate solution. The residual network is a non-trainable part of PINN capable of computing derivatives of approximator network outputs for the inputs, resulting in the composite loss function, denoted by MSE.

3.2.1 *PINN solution to the diffusion equation*

This problem is solved following the PINN approach by constructing a neural network to approximate the concentration $C(x, t)$. The approximate solution is differentiated concerning their variables, for values defined in the set of collocation points selected from the domain $\mathcal{D} \times [0, T]$, where $\mathcal{D} \subset \mathbb{R}^d$ is a bounded domain, and T denotes the final time. The loss function consists of terms used in (17), (18), and (19) by neural network approximations of C at collocation points, where terms include a given PDE and the initial and boundary conditions along the domain boundary.

The loss function consists of terms used for the assessment of the difference between C and its approximation $\widehat{C}$ obtained by a PINN. Here, terms represent residuals of the governing PDE, the initial and boundary conditions. The total loss $\mathcal{L}$ is determined by the sum of residuals as:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_0 + \mathcal{L}_{b_1} + \mathcal{L}_{b_2}, \tag{20}$$

where

$$\mathcal{L}_r = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| \frac{\partial \widehat{C}(x,t)}{\partial t} - D \frac{\partial^2 \widehat{C}(x,t)}{\partial x^2} + \alpha \right|^2, \tag{21}$$

$$\mathcal{L}_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} \left| \widehat{C}(x,0) - C_0 \right|^2, \tag{22}$$

$$\mathcal{L}_{b_1} = \frac{1}{N_{b_1}} \sum_{i=1}^{N_{b_1}} \left| \widehat{C}(0,t) - C_0 \right|^2, \tag{23}$$

$$\mathcal{L}_{b_2} = \frac{1}{N_{b_2}} \sum_{i=1}^{N_{b_2}} \left| \frac{\partial \widehat{C}(\infty,t)}{\partial x} \right|^2, \tag{24}$$

The first term $\mathcal{L}_r$ penalizes the governing equation (1), N_r being the batch size of collocation points randomly sampled in the training domain consisting of spatial and temporal coordinates, using a uniform distribution as a Hammersley point set. $\widehat{C}(x, t)$ is the neural network approximation of the concentration field $C(x, t)$. The second term $\mathcal{L}_0$ determines the fulfillment of the initial condition (2). The terms $\mathcal{L}_{b_1}$ and $\mathcal{L}_{b_2}$ indicate residuals of Dirichlet and Neumann boundary conditions given in (3). N_0, N_{b_1}, and N_{b_2} denote the numbers of the collocation points in which initial and boundary conditions apply.

In the case of periodic boundary conditions driven by daily changes of oxygen concentration, the $\mathcal{L}_{b_1}$ can be written as:

$$\mathcal{L}_{b_1} = \frac{1}{N_{b_1}} \sum_{i=1}^{N_{b_1}} \left| \widehat{C}(0,t) - C_0 \left[1 + \varepsilon \sin(2\pi t/T) \right] \right|^2. \tag{25}$$

3.3 *Treating non-physical solutions*

It is shown that the models based on the equations given in sections 3.1 and 3.2, when time passes a certain critical point, either PINN or EFD method, start to result in negative concentration profiles. Looking at Eq. (1), the primary cause of such behavior is the factor of activity α. Since $C < 0$ is non-physical per se, the obtained solution requires special treatment. The analytical method by [8] and finite-element scheme by [9] do not consider any special treatment of these

non-physical concentrations, with a valid reason. Since both consider only a range $x \in [0, 100\,\text{cm}]$, it was unnecessary to deal with this problem. However, if we consider the extended range $x \in [0, 700\,\text{cm}]$, as depicted in Figure 3, the areas where $C < 0$ start to appear. We will eliminate them by enforcing two different methods. One will apply to the EFD scheme, and the other will apply to the PINN approach.

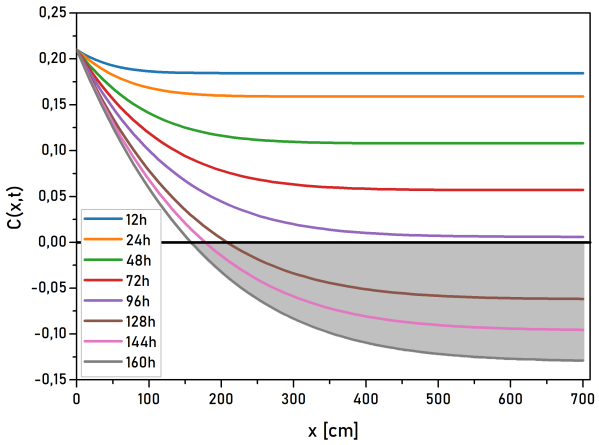

Figure 3. The analytical solution for the problem posed by [9], but in the extended range of $x \in [0, 700\,\text{cm}]$. The last three times exhibit areas where $C < 0$.

3.3.1 *EFD approach*

Let's first consider how to deal with this issue in the proposed EFD scheme. Replacing negative C values with zeros is insufficient as a post-processing action. These boundary values impact the entire transient solution and should be replaced immediately upon appearance. Upon calculating $C_{i,j+1}$ according to Eq. (8), we will replace negatives by zeros like this:

$$C_{i,j+1} = \max(0, C_{i,j+1}), \tag{26}$$

and proceed with time-stepping.

3.3.2 *PINN approach*

However, since PINN does not include any time-stepping, we cannot utilize such a simple scheme. In PINN, it is possible to operate directly with the loss function to provide treatment of the non-physical values. We can "force" the PINN to prevent negative values of the resulting function directly in the training phase, by introducing the following modification to the loss term $\mathcal{L}_r$, specified in Eq. (21):

$$\mathcal{L}_r^* = \mathcal{L}_r \cdot [1 + \text{sgn}(C - C_{tol})] + \lambda \cdot \frac{C}{C_0} \cdot [1 - \text{sgn}(C - C_{tol})] \tag{27}$$

where C_{tol} denotes the tolerance, C_0 the reference concentration, and λ represents the scaling factor. The basic idea is to change the behavior of loss $\mathcal{L}_r$ depending on the value of C. If $C > C_{tol}$, it behaves normally, according to the differential equation. However, in the case that $C < C_{tol}$, we lead C to simply drop to zero, thanks to the gradient. The scaling factor λ is mandatory to keep the second (artificial) loss term in the same order of magnitude as the first one. It can be defined as:

$$\lambda \propto 10^{\lfloor log_{10} |\mathcal{L}_r| \rfloor} \tag{28}$$

The scaling factor represents the order of magnitude of the first loss term $\mathcal{L}_r$ in Eq. (27). Since $\frac{C}{C_{tol}} \in [0, 1]$, the magnitude of the second loss term is λ. In this way, it is possible to ensure the same order of magnitude of $\mathcal{L}_r$ and the custom loss term, for the entire training time. We used:

$$\lambda = 5 \cdot 10^{\lfloor log_{10} |\mathcal{L}_r| \rfloor} \tag{29}$$

in all our experiments. It is likely possible to completely avoid scaling the second loss term, by employing the adaptive weights, as implemented by Haghighhat *et al*[23], but the proposed formulation works sufficiently well.

4 Results & Discussion

To facilitate the comparison of results, we applied the proposed EFD scheme and PINN with customized loss to the diffusion of oxygen in a soil column with the geometry used in the work by Kalita [9], Figure 1. The following values of diffusion coefficient (D) and activity (α), measured by Papendick and Runkles [7], are used: $D = 259.2\,\text{cm}^2 \cdot \text{h}^{-1}$ and $\alpha = 0.002\,125\,\text{cm}^3 \cdot \text{cm}^{-3} \cdot \text{h}^{-1}$. We first investigate a case when the air oxygen concentration at $x = 0$ is $C(0, t) = C_0 = 0.21\,\text{cm}^3 \cdot \text{cm}^{-3}$.

Figure 4 shows the numerical results for relative oxygen concentration at eight different times obtained by solving the diffusion equation (1) by the EFD method and PINN in the case of constant air oxygen concentration. The step lengths $\Delta x = 5\,\text{cm}$ and $\Delta t = 0.000\,05\,\text{h}$ have been used to achieve stability of the EFD scheme. In equation (12), we used $x_\infty = 700\,\text{cm}$ as the distance at which there is no change in the oxygen concentration. Increasing x_∞ had little impact on the solution but greatly increased the grid size and computation time.

The PINN applied to this problem consists of 3 hidden layers with 80 neurons each, activated by the sigmoid activation function. To carry out the back-propagation, we sum up the loss in 8000 collocation points within the domain, including 2000 boundary points and 500 for the initial conditions. Following the PINN approach, we strategically place those collocation points near the domain boundaries to effectively enforce the specified Dirichlet and Neumann boundary conditions, as well as initial conditions. Adam optimization with a learning rate of 1.0e-03 is applied in the training through 500.000 iterations, and finalizing by the L-BFGS method, as suggested by Markidis [24]. Implementing the methods mentioned above and algorithms can be explored within the well-known DeepXDE library designed for scientific machine learning [25].

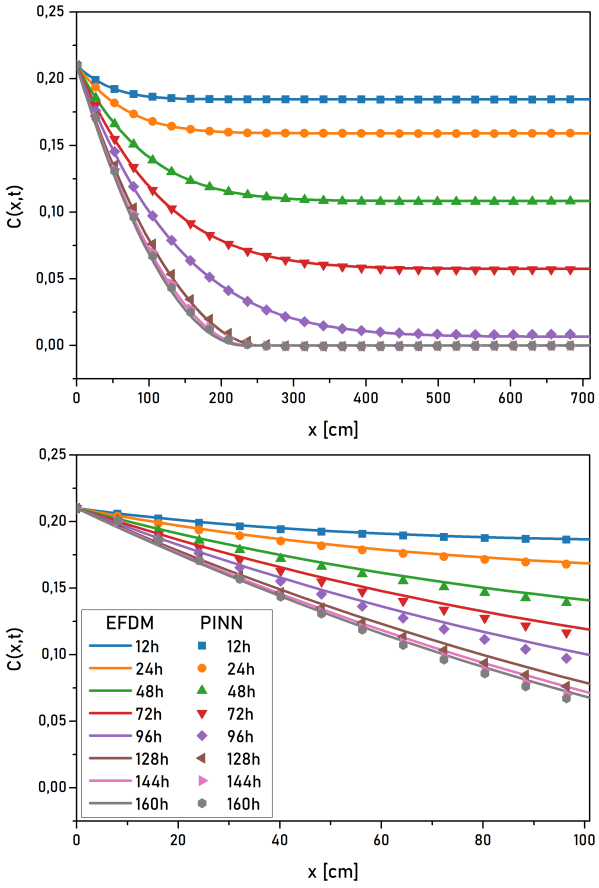

Figure 4. Oxygen concentration vs. depth of the soil column at different diffusion times for constant air oxygen concentration $C_0 = 0.21\,\mathrm{cm}^3{\cdot}\mathrm{cm}^{-3}$. The upper diagram shows the entire domain of 700 cm. The bottom diagram presents the same data focused on the area from 0 to 100 cm.

A good agreement between EFD and PINN solutions is apparent from Figure 4, both applying their specific mechanisms for avoiding negative concentration, described in Section 3.3. The maximum PRMSE between EFD and PINN occurs at 160 hours, reaching 1.03 %, as shown in Table 1.

Figure 5 shows numerical results for the oxygen concentration inside the soil column for various diffusion times when the air oxygen concentration periodically changes as $C(x=0,t) = C_0{\cdot}[1+\varepsilon\,\sin(2\pi\,t/T)]$, where $C_0 = 0.21\,\mathrm{cm}^3{\cdot}\mathrm{cm}^{-3}$, $\varepsilon = 0.05$, and $T = 24\,\mathrm{h}$ are assumed. The PINN applied to this problem is more complex, to catch the periodic phenomena and consists of 4 hidden layers with 180 neurons each, activated by the sigmoid activation function. The normal and boundary collocation point numbers are the same as in the case of constant C_0, as well as the optimization method and learning rate. The maximum PRMSE between PINN and EFD solutions occurs at 160 hours, reaching 0.49 %, as shown in Table 1.

A good match between two different approaches to solving a diffusion PDE demonstrates the correctness of the proposed techniques for avoiding non-physical solutions to the problem of oxygen diffusion in soil. We must emphasize that finding optimal hyper-parameters for the PINN, especially in the case of periodic boundary conditions was challenging. At first, we tried manually but at later stages utilized the technique proposed by Kaplarević *et al* [26]. The PINN/GA (PINN/Genetic Algorithm) strategy provides a fully automatic design of a PINN by an evolutionary strategy with specially tailored GA operators of selection, crossover, and mutation, adapted for deep neural network architecture and hyper-parameter search.

Table 1. RMSE for different times

t [h]	$PRMSE(constant)$	$PRMSE(periodic)$
12	0.06%	0.05%
24	0.11%	0.08%
48	0.18%	0.15%
72	0.28%	0.19%
96	0.47%	0.20%
128	0.67%	0.44%
144	0.82%	0.35%
160	1.03%	0.49%

5 Conclusion

We investigated oxygen diffusion in the soil in one dimension using the EFD method, and a PINN. Due to the impact of the activation parameter α, a concentration may become negative at some point in time. We proposed two techniques to overcome this issue, one trivial for EFDM, and another, more advanced, for PINN. The innovative solution uses a custom loss function, specially tailored to avoid resulting negative concentration profiles. According to the results, the PINN with a customized loss is effective and accurate, and there are no obstacles to application to analogous problems.

On the other hand, there is a lot of room for further improvements. For example, one can work on an automatic determination of the parameter λ from Eq. 28, then the automatic determination of the number of layers, neurons per layer, activation functions, and other hyperparameters. Although PINN has shown solid accuracy, its computation time is much higher than the simple EFD scheme. It is noteworthy that PINN has proven to be an efficient tool for solving nonlinear PDEs. Furthermore, PINN shows its real strength in solving inverse problems, where there are no obstacles to using the proposed approach of preventing non-physical solutions.

Author Contributions

Conceptualization: Svetislav Savović, Miloš Ivanović; methodology, software: Miloš Ivanović, Svetislav Savović; validation: Ljubica Kuzmanović, Miloš Ivanović; writing original draft preparation: Miloš Ivanović; writing-review and editing, Svetislav Savović, Milan Kovačević; visualization: Ljubica Kuzmanović; supervision: Svetislav Savović.

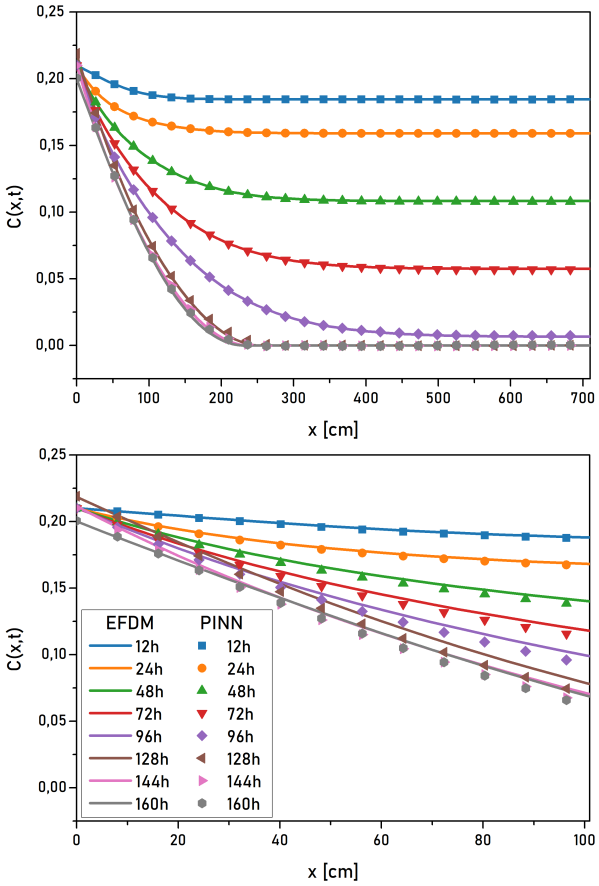

Figure 5. Oxygen concentrations vs depth of the soil column at different diffusion times obtained for periodically changing air oxygen concentration $C(x=0,t) = C_0 \cdot [1 + \varepsilon \sin(2\pi\, t/T)]$, $C_0 = 0.21\,\mathrm{cm^3 \cdot cm^{-3}}$, $\varepsilon = 0.05$, and $T = 24\,\mathrm{h}$. The upper diagram shows the entire domain of 700 cm, and the bottom diagram presents the same data focused on the area from 0 to 100 cm.

Funding

The work described in this paper was supported by a grant from the Serbian Ministry of Science, Technological Development and Innovations (Agreement No. 451-03-65/2024-03/200122).

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Hillel, Daniel J 1980 Fundamentals of soil physics. Academic Press, New York

[2] Refsgaard, JC, Christensen, TH, and Ammentorp, HC 1991 A model for oxygen transport and consumption in the unsaturated zone. *J. Hydrol.* 129: 349–369

[3] Kanwar, RS, Mukhtar, S, and Singh, P 1989 Transient-state oxygen diffusion through undisturbed soil columns. *Trans. ASAE.* 32: 1645–1650

[4] Huang, C-H, Ouyang, Y, and Zhang, J-E 2006 Effects of a compact layer on soil O2 diffusion. *Geoderma*. 135: 224–232

[5] Letey, J, Stolzy, LH, Valoras, N, and Szuszkiewicz, TE 1962 Influence of Soil Oxygen on Growth and Mineral Concentration of Barley 1. *Agron. J.* 54: 538–540

[6] Erickson, AE 1982 Tillage effects on soil aeration. *Predict. tillage effects soil phys. prop. processes*. 44: 91–104

[7] Papendick, RI and Runkles, JR 1965 Transient-state oxygen diffusion in soil: I. The case when rate of oxygen consumption is constant. *Soil Sci.* 100: 251–261

[8] Kanwar, RS 1986 Analytical solutions of the transient state oxygen diffusion equation in soils with a production term. *J. Agron. Crop Sci.* 156: 101–109

[9] Kalita, Prasanta K 1999 Transient finite element method solution of oxygen diffusion in soil. *Ecol. Modell.* 118: 227–236

[10] Savović, S and Caldwell, J 2003 Finite difference solution of one-dimensional Stefan problem with periodic boundary conditions. *Int. J. Heat Mass Transfer.* 46: 2911–2916

[11] Savović, S and Caldwell, J 2009 Numerical solution of Stefan problem with time-dependent boundary conditions by variable space grid method. *Therm. Sci.* 13: 165–174

[12] Anderson, D, Tannehill, JC, Pletcher, RH, Munipalli, R, and Shankar, V 2020 Computational fluid mechanics and heat transfer. CRC press

[13] Nagy, Ádám, Omle, Issa, Kareem, Humam, Kovács, Endre, Barna, Imre Ferenc, and Bognar, Gabriella 2021 Stable, explicit, leapfrog-hopscotch algorithms for the diffusion equation. *Computation*. 9: 92

[14] Savović, Svetislav, Drljača, Branko, and Djordjevich, Alexandar 2022 A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes. *Ricerche di Matematica*: 1–8

[15] Kovács, Endre, Majár, János, and Saleh, Mahmoud 2024 Unconditionally positive, explicit, fourth order method for the diffusion-and Nagumo-type diffusion–reaction equations. *Journal of Scientific Computing*. 98: 39

[16] Khayrullaev, Husniddin and Kovács, Endre 2024 Numerical performance of some positivity preserving methods for the diffusion equation where the diffusion coefficient depends on both time and space. *International Journal of Advanced Natural Sciences and Engineering Researches*. 8: 284–292

[17] Khayrullaev, Husniddin, Omle, Issa, and Kovács, Endre 2024 Systematic Investigation of the Explicit, Dynamically Consistent Methods for Fisher's Equation. *Computation*. 12: 49

[18] Raissi, M, Perdikaris, P, and Karniadakis, GE 2019 Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378: 686–707

[19] Dimitris, CP and Lyle, HU 1992 A hybrid neural network-first principles approach to process modeling. *AIChE J.* 38: 1499–1511

[20] Lagaris, IE, Likas, A, and Fotiadis, DI 1998 Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* 9: 987–1000

[21] Abadi, M, Barham, P, Chen, J, Chen, Z, Davis, A, Dean, J, Devin, M, Ghemawat, S, Irving, G, Isard, M, Kudlur, M, Levenberg, J, Monga, R, Moore, S, Murray, DG, Steiner, B, Tucker, P, Vasudevan, V, Warden, P, Wicke, M, Yu, Y, and Zheng, X 2016 TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. 265–283

[22] Paszke, A, Gross, S, Chintala, S, Chanan, G, Yang, E, DeVito, Z, Lin, Z, Desmaison, A, Antiga, L, and Lerer, A 2017 Automatic differentiation in PyTorch. *31st Conference on Neural Information Processing Systems*

. [23] Haghighat, E and Ruben, J 2021 SciANN: A Keras / TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput. Methods Appl. Mech. Eng.* 373: 113552

[24] Markidis, S 2021 The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Front. Big Data*. 4: 669097

[25] Lu, L, Meng, X, Mao, Z, and Karniadakis, G 2021 DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* 63: 208–228

[26] Kaplarević-Mališić, A, Andrijević, B, Bojović, F, Nikolić, S, Krstić, L, Stojanović, B, and Ivanović, M 2023 Identifying optimal architectures of physics-informed neural networks by evolutionary strategy. *Appl. Soft Comput.* 146: 110646