Gulf Journal of Mathematics ISSN: 2309-4966
Vol 16, Issue 2 (2024) 337-352 https://doi.org/10.56947/gjom.v16i2.1863

NUMERICAL SIMULATION OF POLLUTANTS TRANSPORT IN GROUNDWATER USING DEEP NEURAL NETWORKS INFORMED BY PHYSICS

SOULEYMANE ZIO[1*], BERNARD LAMIEN[2], MOHAMED BEIDARI[1], TOUGRI INOUSSA[2]

Abstract. Real-time monitoring of groundwater pollutants is very challenging to implement due to the difficulty and high economic cost. Groundwater numerical modeling is generally used as an alternative to create protection systems and illustrate their effectiveness. Several groundwater simulation software programs are available, however, they are based on numerical methods that require high spatial and temporal meshing, which considerably increases their computational costs and limits their use for optimization process. To solve this problem, a methodology that combines machine learning techniques based on neural networks, flow, and transport equations in underground reservoirs is proposed. This method is known as physics-informed deep neural networks, abbreviated as PINN. In this work, we will show the performance of PINN in predicting the flow and transport of pollutants in an underground reservoir. To evaluate the effectiveness of the PINN, the finite element method with a fine mesh grid is used as a reference solution to validate the PINN results. The results show that PINNS provide very good result in predicting pollutant transport in the reservoir with a relatively low cost.

1. Introduction and preliminaries

The assessment and management of groundwater pollution risks is crucial for preventing potential sources of pollution. To overcome the difficulties and economic costs associated with real-time monitoring of groundwater pollutants, numerical simulation is utilized to predict water flow and solute transport in groundwater reservoirs. The numerical simulation of this physical phenomenon is based on the coupling of the transient equations of water flow and solute transport in a porous medium which can be homogeneous or heterogeneous. One of the most widely used software programs today is MODFLOW software developed by the United States Geological Survey (USGS). Several other methods exist in the literature, they are generally based on the so-called traditional methods namely the finite element, finite difference, and finite volume methods. These methods have shown their effectiveness in solving several problems of interest in different fields of science and engineering (aerospace, mechanics, climate, etc.).

Date: Received: Feb 14, 2024; Accepted: Apr 4, 2024.
* Corresponding author ziosouleymane27@gmail.com.
2020 *Mathematics Subject Classification.* Primary 46L55; Secondary 44B20.
Key words and phrases. Neural networks, finite element method, groundwater, pollution.

However, they require a fine spatial and temporal mesh which considerably increases their computational costs. The cost of calculation increases significantly when traditional methods are employed to solve optimization or inverse problems, as it requires multiple direct problem assessments due to their iterative nature. In order to reduce the computational costs of simulation software based on traditional numerical methods several alternatives are proposed in the literature [5, 6, 7, 9, 12, 1, 2, 3]. One of the recent alternatives is to substitute the computer code by a model that uses neural networks [1, 2, 3]. This model called surrogate model or digital twin is designed from a large amount of data provided by the computer code. These data obtained in time and space are therefore used to train the neural network. However, this alternative has several limitations: multiple possible solutions, not taking into account the reality of physical phenomena and large learning database. Because neural networks are stochastic, from one learning to another, the results can be different, which can be problematic. Moreover, traditional neural networks do not take into account physical phenomena. In this work, we will implement a software to calculate the transport of pollutants in underground reservoirs using the method of PINNs (Physics Informed Neural networks) [19, 20]. PINNs are a new class of neural networks that combine machine learning and physical laws. This recent approach was born in 2019 through the work of Maziar Raissi, Paris Perdikaris, and George en Karniadakis [21, 22]. PINNs are neural networks that integrate the physical laws represented in the form of differential equation(s) in the construction of a simulation software that is less costly in computing time. Here, we build a PINN based on the physical laws of water flow coupled with pollutant transport phenomena in an underground reservoir.

2. Governing equations

The transport of contaminants in an underground reservoir can be modelled by coupled equations, namely the flow equation and the transport equation. The velocity field obtained by the resolution of the flow equation is used in the advection-dispersion equation that reflects the progress of the pollutant in underground media. Here, we consider the presence of a single type of pollutant emitted from a source of contamination Sr. It is also assumed that the pollutant does not react with the porous medium. The two coupled equations and the boundary conditions can be written as follows:

2.1. **Flow equation.**

$$\frac{\partial}{\partial x_i}[K_{ij}\frac{\partial h}{\partial x_j}] = 0 \tag{2.1}$$

where K_{ij} is the transmissivity tensor $(L2T{-}1)$, h is the water pressure.

$$q = -K\frac{\partial h}{\partial x_i} \tag{2.2}$$

q is the volume of water discharged per unit area and per unit time, or the unit flow in $(L2T{-}1)$. The flow velocity that allows the coupling of the two equations $v = \frac{q}{\theta}$, x_j, x_i are Cartesian coordinates, i=1.2.

2.2. **Pollutant transport equation.**

$$\frac{\partial(\phi C)}{\partial t} + \frac{\partial}{\partial x_i}(\phi v_i C) = \frac{\partial}{\partial x_i}(\theta D_{ij}\frac{\partial C}{\partial x_j}) + Sr(x_i, x_j, C) \tag{2.3}$$

where ϕ is dimensionless porosity, C is the concentration of the pollutant,D_{ij} is the tensor of the hydrodynamic dispersion coefficient, v_i is the linear velocity of the pore water, Sr is the volumetric flow per unit volume of the aquifer representing the sources (positive) and wells (negative).

3. PINNs:Physics-informed neural networks

Physics-informed neural networks, abbreviated PINNS, are a new class of method for finding an approximation of the solution of a differential equation using ANN neural networks. In this section we first introduced the ANN and in second section we introduced the PINNS.

3.1. **ANN Neural Network.** A neuron network [9, 10, 11, 12] is a mathematical function that links a vector **X** (model input) to a vector **Y** (model output). The change from **X** to **Y** is done by a succession of n layers where layer n-1 is used to obtain the response of layer n. The mathematical expression can be written:

$$\mathbf{H}_n = \sigma\big(\mathbf{W}_n\mathbf{H}_{n-1} + \mathbf{B}_n\big), \tag{3.1}$$

where $\mathbf{W}_n$ and $\mathbf{B}_n$ represents the weights and biases associated with layer n and σ a function called activation function which is generally nonlinear. This description is illustrated in Figure :

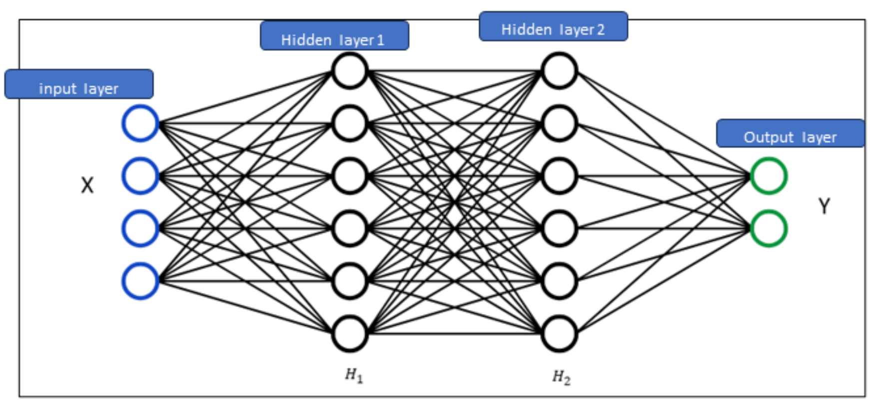

FIGURE 1. Artificial Neuron Network Diagram (ANN)

In Equation 3.2 we present the loss function used to find the ANN parameters.

$$\mathfrak{L} = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{s}}^i - \mathbf{s^i})^\mathbf{2} \tag{3.2}$$

$\hat{\mathbf{s}}$ approximate solutions obtained by ANN and $\mathbf{s}$ real solutions of the problem,$\mathbf{s}$ represents the solution pair $\mathbf{s} = \{\mathbf{h}, \mathbf{C}\}$. This type of network is widely used in several fields, such as marketing or behavioural analysis issues (social networks, etc.), they are widely used to learn the evolution of a physical magnitude such as the velocity of fluid flow, temperature, deformation of structures, etc. Nevertheless, they have several limitations in solving physical problems, which include the fact that, they do not account for physical phenomena in presence, and the need to have a large amount of data for training. Recently, a family of ANN-based methods that take into account the physics of the problem are used for differential equation resolution. This class of method includes DeepONet, fourier Neural Operator (FNO), and PINNS [23].

3.2. **Implementation of a physical informed neural network.** In contrast to ANNs, PINNS only use mathematical models of the problem under strong formulation, boundary conditions, and initial conditions to train their network. Physics-informed neural networks (PINNS) come as an alternative to conventional neural networks. This family of neural networks takes as input the space variables **x** and time t and as output the solution of the differential equation that governs the studied problem. The PINNS have an architecture identical to the classical networks, that is to say constitute several layers and a non-linear activation function. The big difference lies in the calculation of the loss function and the input vectors. The differential equations that represent the physical phenomeno are directly involved in calculating the loss function in a PINN. The idea is to calculate differential equations, boundary and initial conditions at collocation points in the problem domain, then by minimizing loss functions, the best weights and biases associated with physical problems are found. The PINN algorithm can be explained briefly in Figure 2. The Figure illustrates that the spatial and temporal variables of the problem are regarded as network inputs. The governing equations of the physical problem, the boundary and initial conditions are used to minimize the loss function, which depends on the parameters of the neural network.

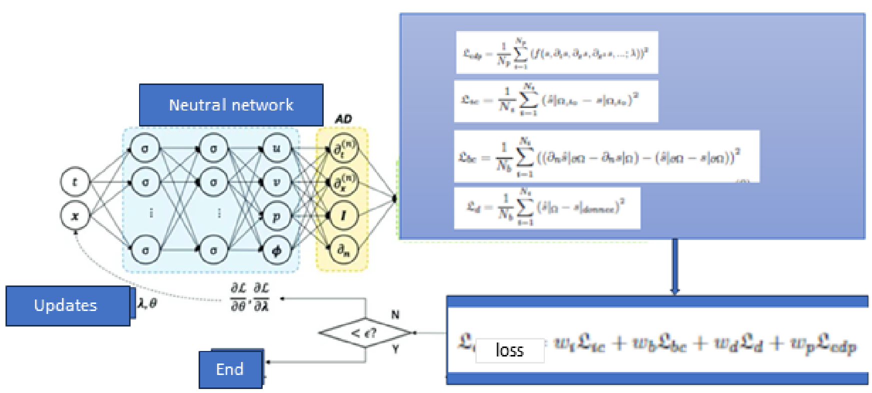

FIGURE 2. Algorithm implementation of the PINN

Considering that the governing equation of the physical problem in the field $\mathbf{\Omega}$ and bounded by $\partial\mathbf{\Omega}$ and set by λ can be written as follows:

$$F(\mathbf{x}, t : \mathbf{\Delta s}; \lambda) \qquad x \,\epsilon\, \mathbf{\Omega}, \tag{3.3}$$

To equation 3.3 can be associated different kind of boundary conditions(Dirichlet, Neumann, Robin, etc.) on $\partial\Omega$ boundaries and initial conditions. The variables $\mathbf{x}$, t represent the locations or positions and the time, $\mathbf{s}$ the solutions of the problem and $\mathbf{\Delta}$ a differential operator that can be linear or non-linear represented by the partial derivatives $\left(\frac{\partial}{\partial t}, \frac{\partial}{\partial \mathbf{x}}, \frac{\partial^2}{\partial \mathbf{x^2}}, etc..\right)$. The loss function to be minimized in the PINN can be written as:

$$\mathfrak{L}_{loss} = w_i \mathfrak{L}_{ic} + w_b \mathfrak{L}_{bc} + w_d \mathfrak{L}_d + w_p \mathfrak{L}_{edp} \tag{3.4}$$

where w_i, w_b, w_d and w_p are the weights of the error function associated respectively with the initial condition, boundary conditions, data and partial differential equations. The different loss functions can be written as:

$$\mathfrak{L}_{ic} = \frac{1}{N_i} \sum_{i=1}^{N_i} \left(\hat{s}|_{\Omega,t_0} - s|_{\Omega,t_0} \right)^2 \tag{3.5}$$

$$\mathfrak{L}_{bc} = \frac{1}{N_b} \sum_{i=1}^{N_i} \left((\partial_n \hat{s}|_{\partial\Omega} - \partial_n s|_{\partial\Omega}) - (\hat{s}|_{\partial\Omega} - s|_{\partial\Omega}) \right)^2 \tag{3.6}$$

$$\mathfrak{L}_{d} = \frac{1}{N_b} \sum_{i=1}^{N_i} \left(\hat{s}|_{\Omega} - s|_{data} \right)^2 \tag{3.7}$$

$$\mathfrak{L}_{edp} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left(f(s, \partial_t s, \partial_x s, \partial_{x^2} s, ...; \lambda) \right)^2 \tag{3.8}$$

where t_0 is the initial time, $\hat{s}$ the true solution, N_i, N_b, and N_p are respectively the number of collocation points used to generate differential equation data, the initial condition, boundary condition, and the number of collocation points used to evaluate the EDP.

In this study, the differential equation $F(\mathbf{x}, t : \mathbf{\Delta s}; \lambda)$ is represented by flow equation and the diffusion advection equation presented respectively by the equations 2.1, 2.3 with the boundary and initial conditions. The parameter λ that appears in the differential equation can be obtained by coupling an inversion approach in the optimization of network parameters. In a classical inversion approach, the direct problem (EDP solution) is solved, then numerical data or experimental are used to find unknown parameters through a inversion method. An avandtage of PINNS when used as a forward solver in an inverse problem is that, the search for the unknown parameters target of the inverse problem, as well as, the biases and weights of the PINNS can be done simultaneaously. This approach will be used in the future to identify sources of pollution from underground reservoirs.

4. Results

In this section, we present the results obtained using the PINN method in the implementation of the software design to simulate the flows of water and solute in the aquifer. This will consist of solving the equations 2.1, 2.3.

Before this step, we proceed to the validation of the PINN results using the analytical solutions for the 1D cases and finite element method for 2D [24]. For 1D, we use only the equation 2.3 subject to Neumann and Dirichlet conditions. For the 2D case, the coupled equations 2.1 and 2.3 are used to solve the problem of flow and transport of pollutant in a aquifer.

4.1. **Validation 1D.** The analytical solution of equation of advection-diffusion transient 1D is obtained from the equation 2.3 without source term, that means $Sr = 0$ and that the solution we look for is $C(x,t)$. The equation 1D can therefore be written as:

$$\frac{\partial C}{\partial t} + v\frac{\partial C}{\partial x} - D\frac{\partial^2 C}{\partial x^2} = 0, \tag{4.1}$$

The domain $\Omega = [0, L]$ and $t > 0$, the boundary conditions and initial are respectively x=0, x=L and t=0. This equation accepts analytical solutions for the following Dirichlet and Neumann boundary conditions: At t=0 the initial solution $C(x, t = 0) = f(x) = \sin(\pi x)$, is the same for both analytical solutions.

- **Dirichlet**

$$C(x = 0, t) = C(L, t) = 0,$$

analytical solution can be expressed as :

$$C(x,t) = e^{-\frac{u^2}{4D}t} e^{\frac{u}{2D}x} \sum_{n=1}^{\infty} b_n sin(\frac{n\pi}{L}x) e^{-D(\frac{n\pi}{L})^2)t} \tag{4.2}$$

with

$$b_n = \frac{2}{L}\int_0^L e^{-\frac{u}{2D}x} f(x) sin(\frac{n\pi}{L}x)dx. \tag{4.3}$$

- **Neumann**

$$\frac{\partial C(x = 0, t)}{\partial x} = 0, \frac{\partial C(x = L, t)}{\partial x} = 0, \tag{4.4}$$

the analytical solution :

$$C(x,t) = a_0 + \sum_{n=1}^{\infty} a_n cos(\frac{n\pi}{L}x) e^{-D(\frac{n\pi}{,}L)^2)t} \tag{4.5}$$

with

$$a_0 = \frac{1}{L}\int_0^L f(x) cos(\frac{n\pi}{L}x)dx \quad ; n = 1, 2, ... \tag{4.6}$$

Analytical solutions for each boundary condition were compared to the PINN results. The training step for obtaining the PINN consisted of minimizing the loss function calculated on 1000 collocation points distributed in the domain Ω, and 200 points on the boundary and initial conditions. After the training step, a PINN consisting of a hyperbolic tangent-like activation function and a network architecture with 10 hidden layers, each containing 60 neurons, was obtained and used to predict the pollutant concentration. This architecture was used to predict the solution for the three boundary conditions presented above. After training step , a PINN emulator obtained by using the neutral network composed to hyperbolic tangent activation function and a network architecture with 10 hidden layers, each containing 60 neurons, was obtained and used to predict the pollutant concentration. This architecture was used to predict the solution for the two Boundaries conditions are given above boundary conditions presented above.

Figures show the comparison between PINN prediction and analytical solutions. The results for the Dirichlet condition are presented on Figure (3) and Neumann Figure (4). These results show that the PINN provide a very good results with mean square error (MSE) in the range of 10^{-4} for both cases. These errors can be further decreased when we increase the number of snack points inside the domain and on the boundary conditions, however this increase will require a high computer cost in the training step.

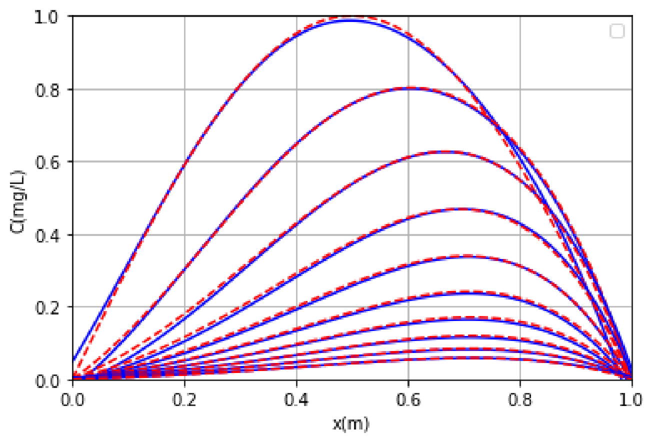

FIGURE 3. Comparison PINN solution and analytical solution with Dirichlet condition

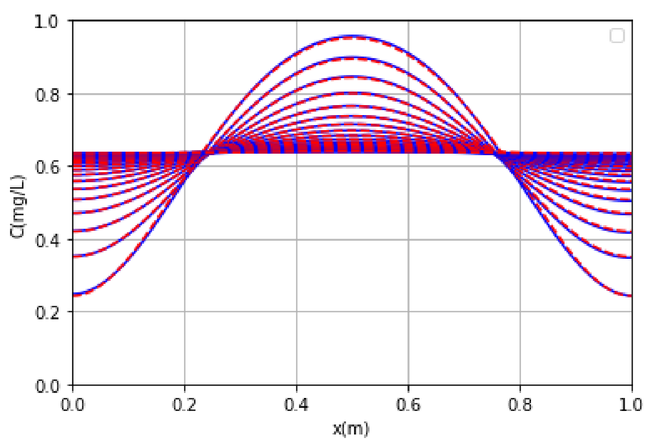

FIGURE 4. Comparison PINN solution and analytical solution with Neumann condition

4.2. **Comparison between PINN solutions and 2D FE solution.** The finite element method has proved its worth in solving differential equations in several engineering fields. It is now one of the most used methods in several simulation software. Unlike the PINNS method which uses the strong form, the finite element method solves the differential equation in its weak form or variational formulation. One may wonder whether it is necessary to use the PINN method while the the traditional ED resolution methods namely FE, VF, DF provide reliable results. Indeed, in several engineering problems, the resolution of ED by conventional FE methods requires a large number of calculations, which can play on rapid decision making. PINNS are presented as an alternative to reduce computational cost and an excellent candidate to develop a multiphysics digital twin. Using the FE method the equation 2.1 and 2.3 can be written in the following weak form: let W the test function; such as $W|_{\Gamma_D} = 0$

$$\int T_{ij}\frac{\partial W}{\partial x_i}\frac{\partial h^n}{\partial x}d\Omega = \int_{\Gamma} W T_{ij}\frac{\partial h^n}{\partial x}d\Gamma + \int Q_S W d\Omega \tag{4.7}$$

$$\int W\frac{C^{n+1}-C^n}{\Delta t}d\Omega+\int W v_i\frac{\partial C^n}{\partial x_i}d\Omega+\int D_{ij}\frac{\partial W}{\partial x_i}\frac{\partial C^n}{\partial x}d\Omega+\int Sr(x_ij,C^n)d\Omega = \int_{\Gamma_N} D_{ij}\frac{\partial C^n}{\partial x}d\Gamma \tag{4.8}$$

The equations 4.7 and 4.8 are coupled through the calculation of the advection velocity v that appears in the equation 4.8. Indeed,

$$v_i = K_{ij}\frac{\partial h}{\partial \mathbf{x}} \tag{4.9}$$

where,h^n,C^n and C^{n+1} refers to the water pressure at time t, the concentrations at times t and t+dt, Γ_D and Γ_N represent the boundary conditions of Dirichlet and Neumann. The performance of PINNS and FEM depends on a number of parameters. In the case of PINNS, performance depends on network parameters and the number of collocation points used for training. While the precision of the finite element method is much related to the mesh size (or number of elements) and the discretization time step dt.

Here we consider an aquifer of length L=1 [L] and height H=0.5 [L] $\Omega = [0, L] \times [0, H]$. This aquifer is subjected to a flow at the position $x_1 = 0$ and a water pressure imposes H_2 at $x_1 = L$. The lower and upper limits are considered impermeable. It is subjected to a pollution source located at the coordinate point $(x_2 = 0.25, x_1 = 0)$ see Figure 5. To the equations 2.1, 2.3 applies the following boundary and initial conditions:

$$-K\frac{\partial h}{\partial x} = q;\ \ x_1 = 0, \forall t \tag{4.10}$$

$$-K\frac{\partial h}{\partial x} = 0;\ \ x_2 = 0\ \ , or\ \ x_2 = L_2, \forall t \tag{4.11}$$

$$h = H_2,\ \ x_1 = L_1, \forall t \tag{4.12}$$

$$h = 0,\ \ x \in \Omega,\ \ t = 0. \tag{4.13}$$

and for the equation of transport 2.3 :

$$C = C_0(x_2);\ \ x_1 = 0,\ \ \forall t \tag{4.14}$$

$$\frac{\partial C}{\partial x} = 0;\ \ x_1 = L_1, \forall t \tag{4.15}$$

$$\frac{\partial C}{\partial x} = 0;\ \ x_2 = 0\ \ , or\ \ x_2 = L_1, \forall t \tag{4.16}$$

$$C = 0,\ \ x \in \Omega,\ \ t = 0. \tag{4.17}$$

$$C_0(x_2) = C_0 e^{-((x_2-\lambda_1)^2/\epsilon^2)}. \tag{4.18}$$

The parameters λ_1 represent the source location, C_0 is the leak concentration. Here, the hydrodynamic dispersion coefficient tensor $\mathbf{D} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}$ such as $D_{12} = D_{21} = 0$ and $D_{11} = Dw\tau_f + \alpha_L \|\mathbf{v}\|$, $D_{22} = Dw\tau_f + \alpha_T \|\mathbf{v}\|$, coefficient of conductivity $\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$

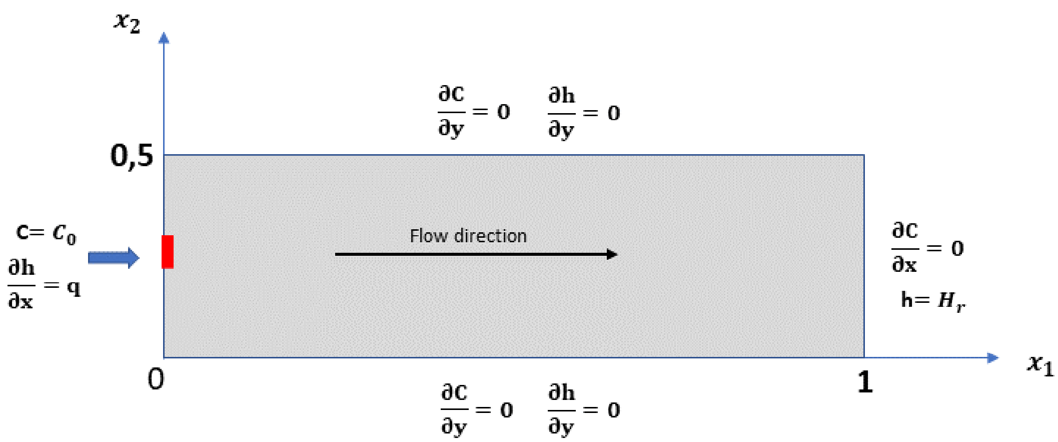

FIGURE 5. Aquifer scheme

For the validation, we consider properties constant in all directions. Aquifer parameters are shown in table below 1:

4.3. **Solution FEM.** The solutions obtained by the finite element method are used as a reference to validate the PINN results, so it is important to make a convergence analysis. The convergence analysis consists of finding the size of the mesh hm for which the MEF solution converges, i.e that the results for different mesh sizes have the converge. We consider triangular elements with four mesh sizes hm = 0.1,0.05,0.02,0.01, for each value of hm the domain Ω is discretized according to the different meshes see Figure 6.

The pollutant concentration measured at the centre of the reservoir at x_2 =0.25 is used for convergence analysis. Figure 7 shows that the solutions converge from

Parameters	values
Longitudinal dispersivity $\alpha_L[L^2T^{-1}]$	0.1
Transverse dispersivity $\alpha_T[L^2T^{-1}]$	0.01
diffusion coefficient D_w	1.2
Coefficient of transmitivity or conductivity $K(LT^{-1})$	1.4
Concentration $C_0[ML^{-1}]$	1
location of the source$\lambda_1[L]$	0.25
tortuosity coefficient tw	0.25
maximal time $T_{max}[T]$	1
aquifer geometry L	$L_1 = 1, L_2 = 0.5$

TABLE 1. aquifer parameters used for simulation

the mesh hm= 0.05, however we consider the solution obtained at the mesh size hm= 0.01 as a reference to analyze the performance of the PINN. In Figure 8, we present the evolution of the pollutant in three different time obtained by the finite element method using a mesh size hm = 0.01. This solution is used as a reference to validate the PINN predictions.

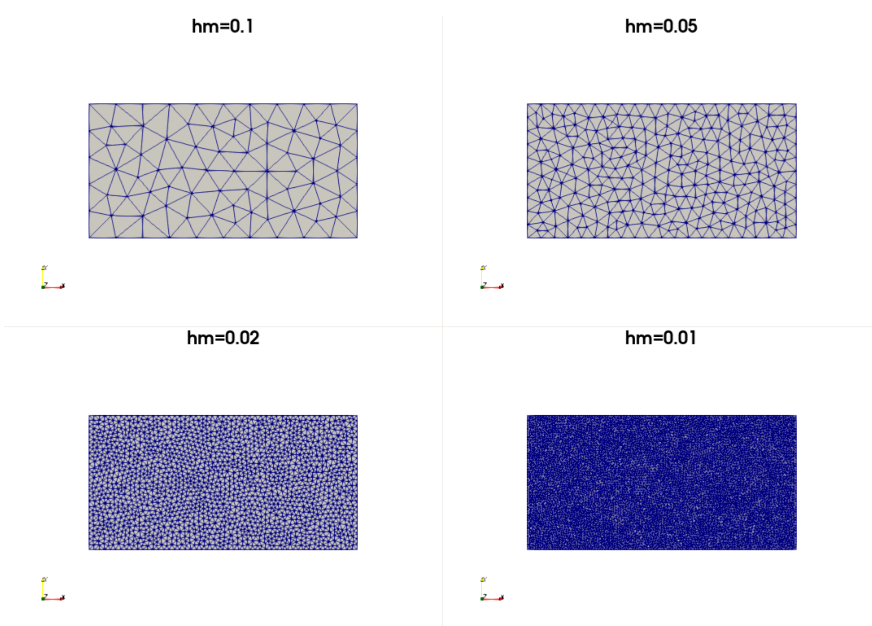

FIGURE 6. Meshing in domain for different mesh size hm

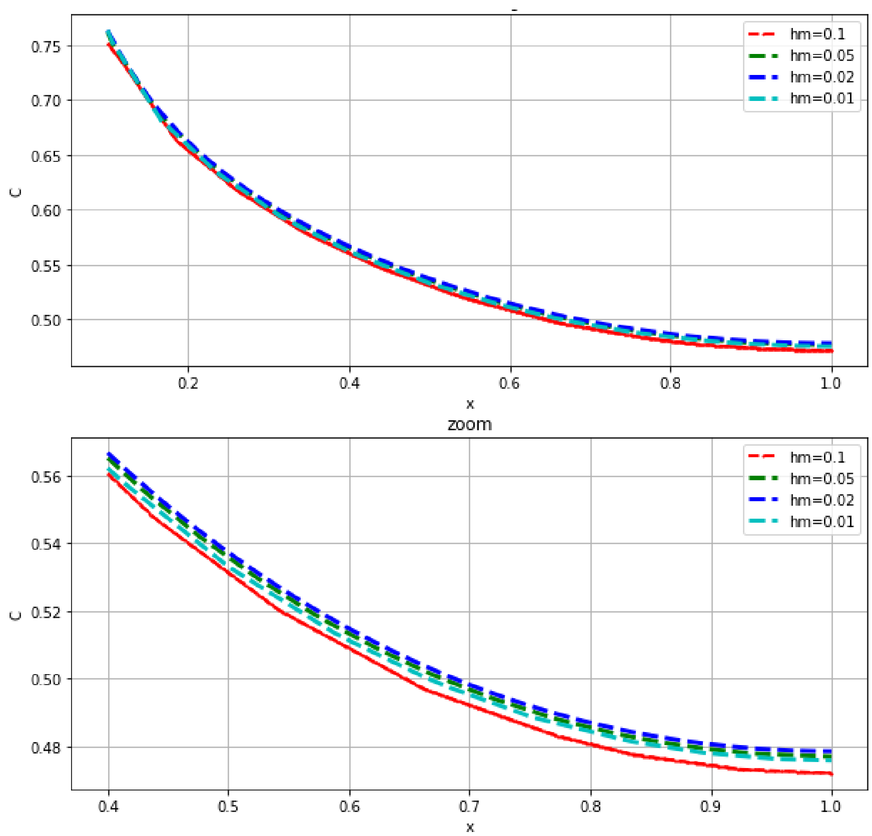

FIGURE 7. Convergence according to the mesh size of finite elements considering contaminant a T=1

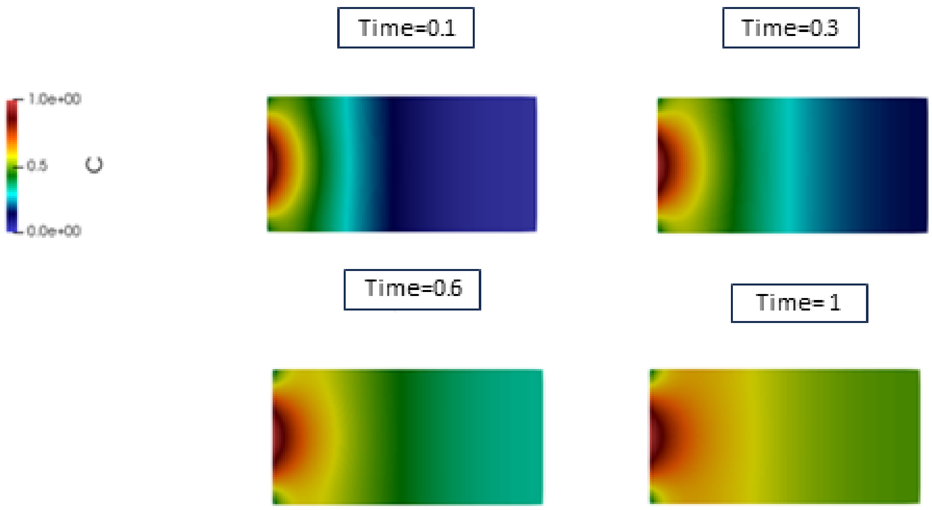

FIGURE 8. Solution obtained using the finite element method mesh size hm=0.01

4.4. **PINN Solution.** In the previous section, we presented the solution of the problem using the finite element method. Here, we analyze the results obtained by the PINNS and the importance of the choice of the number of collocation points at the initial condition (Ni) and at the contour conditions (Nb). The PINN solution depends on the neuron network parameters and the collocation points used to train the model. Its collocation points are located at three levels, namely, within the domain Nd, at the boundary conditions Nb and initial conditions Ni. Here, we analyze the performance of the PINN for different combined values of Nb, Ni. The network architecture being fixed as mentioned above, Nd=2000 collocation points and considering four combinations of the values of Nb, Ni. At each value of Nb and Ni we calculate the absolute error using the PINN solution and the finite element method. In Figure 9, we observe that the error remains constant for both values of Ni and Nb equals to 150. Therefore, we consider both values of Nb and Ni equals to 400 as the ideal number of collocation points to obtained the initial conditions and boundary data. In Figure 10, we present the evolution of the pollutant using the PINNS method. There is a great similarity between the results of the finite element method 8 and the PINN 10.

To better observe the convergence between the results of the two methods, we present in Figure 11 the results of the PINN and MEF obtained at y=0.25. The two results are very close.

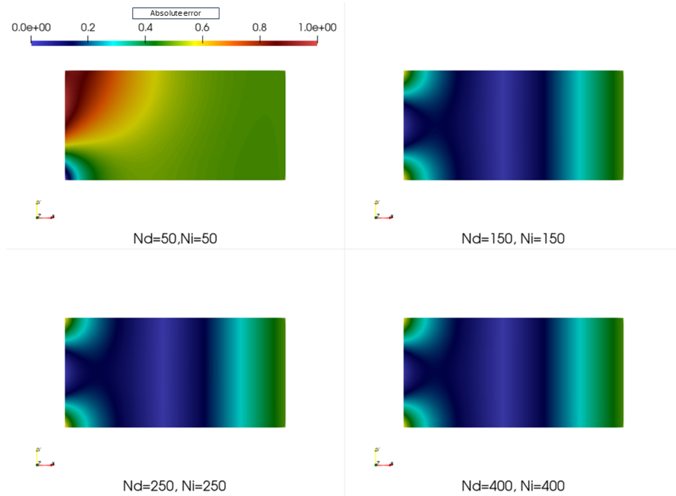

FIGURE 9. Absolute error between PINN solution and FEM

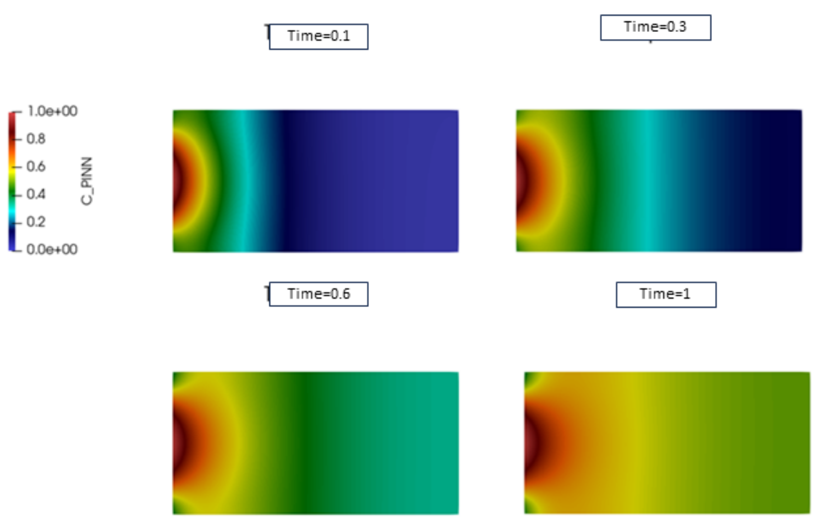

FIGURE 10. Evolution of pollutant obtained using PINN for Nb=400 and Ni=400

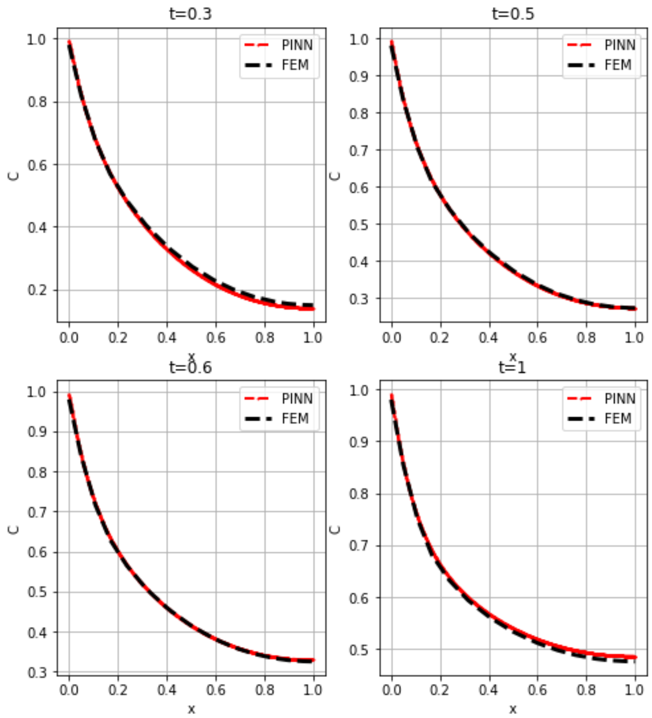

FIGURE 11. PINN and MEF solution at time t=0.3,0.5,0.6 and 1 at coordinate positions y=0.5 and all x taking values between 0 and

5. Conclusion

In this work, we presented the effectiveness of the PINN method to solve the equation of flow and transport of pollutants in an aquifer. To show the effectiveness of the PINN in solving the problem, we confronted it with analytical solutions for one-dimensional cases and the finite element method for the two-dimensional problem. The results showed that the PINN could substitute the finite element method which is time consuming. Nevertheless, to have a good result with the PINNS, the network parameters and number of colocation points at the boundary conditions and inside the domain must be well selected. In future work, we will show how PINN can be used to identify sources of groundwater pollution.

References

1. Raj Singh and Bithin Datta. (2007).Artificial neural network modeling for identification of unknown pollution sources in groundwater with partially missing concentration observation data, Water Resources Management: An International Journal, Published for the European Water Resources Association (EWRA). 21(3), 557-572. https://doi.org/10.1007/s11269-006-9029-z.
2. Raj Mohan Singh, Bithin Datta, and Ashu Jain, A. (2004). Identification of unknown groundwater pollution sources using artificial neural networks. Journal of Water Resources Planning and Management .130(6),506-514. https://doi.org/10.1061/(ASCE)0733-9496(2004)130:6(506).
3. Divya Srivastava and Raj Mohan Singh. (2014). Breakthrough curves characterization and identification of an unknown pollution source in groundwater system using an artificial neural network (ANN). Environmental Forensics .15(2),175-189. https://doi.org/10.1080/15275922.2014.890142
4. Carson, Yolanda and Maria, Anu. (1997).Simulation optimization: methods and applications. WSC '97: Proceedings of the 29th conference on Winter simulation.118–126. https://doi.org/10.1145/268437.268460
5. Ilias Bilionis and Nicholas Zabaras. (2012).Multi-output local Gaussian process regression: Applications to uncertainty quantification. Journal of Computational Physics. 231(17) 0021-9991. www.sciencedirect.com/science/article/pii/S0021999112002513
6. S. Conti and A. O'Hagan.(2010).Bayesian emulation of complex multi-output and dynamic computer models.Journal of Statistical Planning and Inference.140(3) 640-651. https://doi.org/10.1016/j.jspi.2009.08.006
7. Thomas E. Fricker.(2010).Multivariate Gaussian Process Emulators with Nonseparable Covariance Structures.Taylor & Francis 55(1) 47-56. https://doi.org/10.1080/00401706.2012.715835
8. Phillip Boyle.(2007).Gaussian Processes for Regression and Optimisation.Computer Science, Engineering, Mathematics. https://api.semanticscholar.org/CorpusID:118429927
9. Shing Chan and Ahmed H. Elsheikh.(2018). A machine learning approach for efficient uncertainty quantification using multiscale methods.Journal of Computational Physics.354. 493-511. https://doi.org/10.1016/j.jcp.2017.10.034
10. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun .(2015).Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 770-778 https://10.1109/CVPR.2016.90
11. J. S. Hesthaven and S. Ubbiali.(2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. Journal of Computational Physics 363 55-78. https://10.1016/j.jcp.2018.02.037

12. R.K. Tripathy and Ilias Bilionis.(2018).Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification.Journal of Computational Physics 375 0021-9991. 10.1016/j.jcp.2018.08.036
13. Widrow, B. and Lehr, M.A.(1990).30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. Proceedings of the IEEE 78(9) 1415-1442. 10.1109/5.58323
14. Yang Liu and Fan Sun.(2018). A fast differential evolution algorithm using k-Nearest Neighbour predictor. Expert Systems with Applications 38(4) 4254-4258. https://doi.org/10.1016/j.eswa.2010.09.092
15. Augustina Clara Alexander and Julius Musyoka Ndambuki.(2021). Optimal design of groundwater pollution management systems for a decanting contaminated site: A simulation-optimization approach. Groundwater for Sustainable Development 15 10066. https://doi.org/10.1016/j.gsd.2021.100664
16. M. Tamer Ayvaz.(2010). A linked simulation–optimization model for solving the unknown groundwater pollution source identification problems. Journal of Contaminant Hydrology 117(1) 46-59. https://doi.org/10.1016/j.jconhyd.2010.06.004
17. Qiankun Luo and Jianfeng Wu and Yun Yang and Jiazhong Qian and Jichun Wu.(2014). Optimal design of groundwater remediation system using a probabilistic multi-objective fast harmony search algorithm under uncertainty.Journal of Hydrology 519 3305-3315. https://doi.org/10.1016/j.jhydrol.2014.10.023
18. Mitja Janža.(2022). Optimization of well field management to mitigate groundwater contamination using a simulation model and evolutionary algorithm. Science of The Total Environment 807 150811. https://doi.org/10.1016/j.scitotenv.2021.150811
19. Félix Fernández de la Mata and Alfonso Gijón and Miguel Molina-Solana.(2023). Physics-informed neural networks for data-driven simulation: Advantages, limitations, and opportunities. Physica A: Statistical Mechanics and its Applications, 610 128415. https://doi.org/10.1016/j.physa.2022.128415
20. M. Raissi and P. Perdikaris and G.E. Karniadakis.(2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.Journal of Computational Physics, 378 686-707. https://doi.org/10.1016/j.jcp.2018.10.045
21. Raissi, Maziar and Perdikaris, Paris and Karniadakis, George Em, (2017) Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv preprint arXiv:1711.10561. https://arXivpreprintarXiv:1711.10561
22. Raissi, Maziar and Perdikaris, Paris and Karniadakis, (2017) George Em, Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations.arXiv preprint arXiv:1711.10566. https://doi.org/10.48550/arXiv.1711.10566
23. Lu Lu and Xuhui Meng and Shengze Cai and Zhiping Mao , (2022) A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. Computer Methods in Applied Mechanics and Engineering. https://doi.org/10.1016%2Fj.cma.2022.114778
24. Manickam, K., (2015) Optimization techniques for mixed finite element methods of an semilinear elliptic PDEs - superconvergence. Gulf Journal of Mathematics, 3(1). https://doi.org/10.56947/gjom.v3i1.170

[1] Institut du Génie Informatique et telecom, École polytechnique de Ouagadougou, Ouaga 2000, Burkina Faso

Email address: ziosouleymane27@gmail.com

[2] Institut du Génie des Systèmes Industriels et Textiles, École polytechnique de Ouagadougou, Ouaga 2000, Burkina Faso