Physics-informed Neural Networks Based On Sequential Training For CO2 Utilization And Storage In Subsurface Reservoir

Mansour Pour, Kiarash; Voskov, Denis

DOI
10.1615/JMachLearnModelComput.2023048866

Publication date
2023

Document Version
Final published version

Published in
Journal of Machine Learning for Modeling and Computing

Citation (APA)
Mansour Pour, K., & Voskov, D. (2023). Physics-informed Neural Networks Based On Sequential Training For CO2 Utilization And Storage In Subsurface Reservoir. *Journal of Machine Learning for Modeling and Computing*, *4*(4), 27-40. https://doi.org/10.1615/JMachLearnModelComput.2023048866

PHYSICS-INFORMED NEURAL NETWORKS BASED ON SEQUENTIAL TRAINING FOR CO_2 UTILIZATION AND STORAGE IN SUBSURFACE RESERVOIR

***Kiarash Mansour Pour*[1,*] *& Denis Voskov*[1,2]**

[1]*Department of Geoscience and Engineering, TU Delft, Delft, Netherlands*

[2]*Department of Energy Resources Engineering, Stanford University, CA, USA*

*Address all correspondence to: Kiarash Mansour Pour, Department of Geoscience and Engineering, TU Delft, Delft, 2628 CN, Netherlands, E-mail: K.mansourpour@tudelft.nl

Original Manuscript Submitted: 5/15/2023; Final Draft Received: 8/15/2023

CO_2 utilization and storage (CCUS) simulation in subsurface reservoirs with complex heterogeneous structures necessitates a model that can capture multiphase compositional flow and transport. The governing equations are highly nonlinear due to the complex thermodynamic behavior, which involves the appearance and disappearance of multiple phases. Accurate simulation of these processes necessitates the use of stable numerical methods. While machine learning (ML) approaches have been used to solve a variety of nonlinear computational problems, a new approach based on physics-informed neural networks (PINNs) has been proposed for solving partial differential equations (PDEs). Unlike typical ML algorithms that require a large dataset for training, PINNs can train the network with unlabeled data. The applicability of this method has been explored for multiphase flow and transport in porous media. However, for nonlinear hyperbolic transport equations, the solution degrades significantly. This work proposes sequential training PINNs to simulate two-phase transport in porous media. The main concept is to retrain the neural network to solve the PDE over successive time segments rather than train for the entire time domain simultaneously. We observe that sequential training can capture the solution more accurately concerning the standard training for conventional two-phase problems. Furthermore, we extend the sequential training approach for compositional problems in which nonlinearity is more significant due to the complex phase transition. Our approach was tested on miscible and immiscible test cases and showed higher accuracy than the standard training method.

KEY WORDS: *hyperbolic PDE, PINNs, Buckley–Leverett, gas injection, sequential training, compositional simulation, CCUS*

1. INTRODUCTION

The global attempt to mitigate the effects of climate change has increased the urgency of lowering carbon emissions. Carbon dioxide capture, utilization, and storage (CCUS) is one of the existing technologies that has substantial potential for lowering greenhouse gas emissions. The capture of carbon dioxide from industrial sources, compression, transportation, and subsequent utilization for operations such as permanent storage in deep underground geological formations

and increased oil recovery in depleted areas are all part of CCUS. However, proper modeling of multiphase compositional flow and transport in underground reservoirs with complex heterogeneous structures is required to effectively simulate CO_2 use and storage (CCUS).

Compositional simulation deals with the modeling of the flow of multiple phases in a porous medium. The interplay of phase behavior, flow, and transport governs the interactions between hydrocarbon phases. Compositional simulation continues to be a challenging problem. Complexities are mainly due to nonlinear couplings between multiphase multicomponent flow in porous media with thermodynamic phase behavior (Alpak, 2010; Voskov and Tchelepi, 2012). Conventional compositional simulation is based on the solution of the discretized governing equations describing the mass, energy, and momentum transfer in the reservoir either implicitly or explicitly. The fully-implicit method (FIM) is preferred in practice, with the nonlinear system solved by a Newton method due to the fewer restrictions on time-step size. In practical applications, the highly nonlinear nature of problems involving kinks and inflection points poses a significant challenge for Newton's method to achieve convergence, particularly when large time steps are utilized. Consequently, to overcome this limitation, alternative nonlinear solvers have been proposed (Jenny et al., 2009; Jiang and Pan, 2022; Pour et al., 2023) or smaller time steps are selected for simulations.

Machine learning (ML) techniques, particularly deep learning (Lecun et al., 2015), are gaining prominence in the computer science and engineering fields. Notably, physics-informed neural networks (PINNs) are being utilized to solve problems where there is knowledge of engineering conservation equations and constitutive closure relationships, but without labeled data (Raissi et al., 2019). By constructing neural networks with several hidden layers, coupled with nonlinear activation functions, complex nonlinear solutions can be approximated. As a result, PINNs have been employed to tackle diverse applications governed by differential equations, such as the Euler equation (Jagtap et al., 2020, 2022b), gas dynamics (De Florio et al., 2021; Lou et al., 2021), water dynamics (Jagtap et al., 2022b), and chemical kinetics (Ji et al., 2021; Kim et al., 2021). PINNs have demonstrated their versatility in several applications, including data assimilation and observations into numerical models, parameter identification (i.e., solving inverse problems) (Raissi et al., 2019; Yang et al., 2021), and uncertainty quantification (Mao et al., 2020; Tipireddy et al., 2020).

Recently, the application of PINNs has been explored in subsurface flow and transport. Two-phase immiscible transport in porous media is typically described by a nonlinear first-order hyperbolic partial differential equation (PDE), also known as the Buckley–Leverett (B-L) equation. Standard PINNs have been utilized to model this phenomenon (Fuks and Tchelepi, 2020). The authors demonstrated that PINNs cannot find the solution in the case of the steep saturation front with the nonconvex flux function (Fuks and Tchelepi, 2020). Only after an artificial diffusion term was added to the original conservation equation, did the neural networks solution manage to approximate the true solution (Fuks and Tchelepi, 2020). There were multiple attempts to solve this problem by modifying the loss function. Rodriguez-Torrado et al. (2022) proposed a new neural network architecture known as physics-informed attention-based neural networks (PIANNs), which is a blending of recurrent neural networks and attention mechanisms. Fraces and Tchelepi (2021) introduced yet another solution to the B-L problem. Their method entails embedding the entropy and velocity constraints into the neural network residual.

Recently, a sequential training scheme has been proposed for PINNs (Krishnapriyan et al., 2021; Mattey and Ghosh, 2022). In this approach, unlike training for the entire spatiotemporal domain, we discretize the time domain and march in time to reach the end time. Mattey and Ghosh (2022) demonstrate the effectiveness of the sequential training approach for Allen–Cahn

and Cahn–Hilliard equations. Krishnapriyan et al. (2021) demonstrated the method's effectiveness in solving a one-dimensional reaction-diffusion problem. The present study aims to investigate the efficacy of sequential training in the context of hyperbolic transport equations in porous media. This approach is inspired by the conventional practice of using Newton's method to solve PDEs in reservoir simulation, while also taking into consideration the fact that reducing the time step can help improve the effectiveness of Newton's method. Our work suggests a sequential training scheme with a dynamic time step that reduces the number of time steps required for training, instead of uniformly marching in time.

Through the analysis of a single time-step residual, our findings demonstrate that the nonlinearity of the flux becomes increasingly dominant as the time step increases. This nonlinearity can be translated into the residual of the loss function. As a result, the final value of the loss function is higher for larger time steps. Next, we extend the sequential training approach to compositional problems, in which nonlinearity is more pronounced due to complex phase transitions. The presence of discontinuous points that coincide with the phase transition zone can severely hinder the convergence of the loss function for neural networks. To mitigate this issue, we utilize a continuous sigmoid approximation of the fractional flow. Finally, we test the performance of the sequential training approach on pure hyperbolic transport in porous media in the 1D domain, ranging from the classic B-L problem to both immiscible and miscible compositional transport. Our findings demonstrate that the sequential training method outperforms the standard training approach in terms of accuracy, for both immiscible and miscible transport scenarios.

2. GOVERNING EQUATIONS

We consider the general form of the transport equations for an isothermal multiphase compositional problem with n_p phases and n_c components that can be written as

$$\frac{\partial}{\partial t}\left(\phi \sum_{j=1}^{n_p} x_{cj}\rho_j S_j\right) + \operatorname{div} \sum_{j=1}^{n_p} x_{cj}\rho_j \mathbf{v}_j + \sum_{j=1}^{n_p} x_{cj}\rho_j \tilde{q}_j = 0, \quad c = 1, \ldots, n_c. \tag{1}$$

Here, ϕ is porosity, x_{cj} is the mole fraction of component c in phase j, S_j is the phase saturation of phase j, ρ_j is phase molar density, $\mathbf{v}_j$ is phase velocity, and $\tilde{q}_j$ is phase rate per unit volume. Darcy's law is applied to describe how each phase flows:

$$\mathbf{v}_j = -\mathbf{K}\frac{k_{rj}}{\mu_j}(\nabla \mathbf{p_j} - \gamma_j \nabla \mathbf{d}), \tag{2}$$

where $\mathbf{K}$ – permeability tensor, k_{rj} – relative permeability, μ_j – phase viscosity, $\mathbf{p_j}$ – vector of pressures in phase j, γ_j – phase density, $\mathbf{d}$ – vector of depths (positive downwards).

The closure assumption of instantaneous thermodynamic equilibrium further increases the nonlinearity. We used the overall molar formulation suggested by Collins et al. (1992). In this formulation, the following system must be solved at any grid block containing a multiphase (n_p) multicomponent n_c mixture:

$$F_c = z_c - \sum_{j=1}^{n_p} v_j x_{cj} = 0, \tag{3}$$

$$F_{c+n_c} = f_{c1}(p, T, x_1) - f_{cj}(p, T, x_j) = 0, \tag{4}$$

$$F_{j+n_c*n_p} = \sum_{c=1}^{n_c}(x_{c_1} - x_{cj}) = 0, \tag{5}$$

$$F_{n_p+n_c*n_p} = \sum_{j=1}^{np} v_j - 1 = 0. \tag{6}$$

Here $z_c = \Sigma x_{cj}\rho_j s_j/\rho_j s_j$ is the overall composition and $f_{cj}(p, T, x_{cj})$ is the fugacity of component c in phase j. The solution of this system is called a multiphase flash (Michelsen, 1982) and needs to be applied at every nonlinear iteration (Voskov and Tchelepi, 2012). The solution provides molar fractions for each component x_{cj} and phase fraction v_j. The above system of equations provides a complete mathematical statement for multiphase multicomponent transport. Here we concentrate on two-phase, two-component incompressible transport for miscible and immiscible cases.

$$\frac{\phi\partial Z_c}{\partial t} + v_t\frac{\partial F_c}{\partial x} = q_c, \quad x \in \Omega, \quad c \in \{\mathrm{H_2O}, \mathrm{CO_2}\}, \tag{7}$$

where Z_c is the overall composition of component c and v_t is the total velocity. q_c is the source, sink term. F_c is the fractional flow of the component c which is defined as

$$\begin{cases} F = x_c(1 - f(s)) + y_c f(s), & \text{two-phase zone} \\ F = Z, & \text{single-phase zone} \end{cases}, \tag{8}$$

$$f_\alpha = \frac{\lambda_\alpha}{\lambda_{\mathrm{CO_2}} + \lambda_{\mathrm{water}}}, \quad \alpha = \{\mathrm{H_2O}, \mathrm{CO_2}\}, \tag{9}$$

where x_c and y_c are molar fractions of component c in the liquid and gas phases, respectively. $\lambda_\alpha = (kk_{r\alpha})/\mu_\alpha$ stands for the phase mobility, μ_α is the viscosity of the phase, and $k_{r\alpha}(S_\alpha)$ is the relative phase permeability. For the immiscible two-phase B-L transport test case, the fractional flow F_c is equivalent to f_α. The initial and boundary conditions are

$$\begin{cases} Z(x,t) = 0, & \forall x \;\&\; t = 0, & \text{initial condition} \\ Z(x,t) = 1, & x = 0 \;\&\; t > 0, & \text{boundary condition} \end{cases}. \tag{10}$$

3. STD-PINNS SOLUTION

Raissi et al. (2019) proposed that the solution of the PDE could be approximated by a deep neural network through the loss function of the neural network. In a standard PINNs solution, a neural network is trained for the entire spatial-temporal domain. Let's examine the standard format of a PDE,

$$Z_t + \mathcal{N}(Z) = 0, \tag{11}$$

where $Z(x,t)$ denotes the latent (hidden) solution, $\mathcal{N}[.]$ is a nonlinear differential operator. By adopting the methodology of Raissi et al. (2019), the solution $Z(t,x)$ to the PDE is estimated using a deep neural network that relies on a group of parameters denoted as θ. In simpler terms, the PDE solution is expressed as a sequence of function compositions.

Deep neural networks are composed of n_l series of functions,

$$Z_\theta(\mathbf{X}) = z_{n_l}(z_{n_{l-1}}(\ldots(z_2(z_1(\mathbf{X}))))), \tag{12}$$

$$z_i(\mathbf{X}) = \sigma(\mathbf{W_i}\mathbf{X} + \mathbf{b_i}), \quad i = 1, \ldots, n_l, \tag{13}$$

where each hidden layer consists of a stack of artificial neurons that process input feature matrix $\mathbf{X}$ as the weighted sum of weights $\mathbf{W_i}$ and biases $\mathbf{b_i}$ before passing through activation function σ ($\tanh$ in our study); θ is the ensemble of all the model parameters based on weight and biases:

$$\theta = \{W_1, W_2, \ldots, W_{n_l}, b_1, b_2, \ldots, b_{n_l}\}. \tag{14}$$

To provide the neural network with the physics specified by the PDE, we define the residual of the PDE as the left-hand side of Eq. (11) and replace Z_c with $\tilde{Z}_c$.

$$R_\theta(x,t) := (\tilde{Z})_t + \mathcal{N}(\tilde{Z}) = 0. \tag{15}$$

Here, $\tilde{Z}(x,t)$ is the PDE solution which is approximated by a neural network. The loss function of the neural network is made of three error conditions:

$$L_{\text{tot}} = L_i + L_{ii} + L_{iii}. \tag{16}$$

Each error is

$$\begin{cases} L_i = \dfrac{\sum_K^{N_i} \tilde{Z}(x_k^i, 0) - Z_k^i}{N_i}, \\ \qquad x_k^i \in \Omega, \end{cases} \quad \text{mean squared error of the initial condition,}$$

$$\begin{cases} L_{ii} = \dfrac{\sum_K^{N_b} \tilde{Z}(x_k^b, t) - Z(x_k^b, t_k)}{N_b}, \\ \qquad (\mathbf{x_k^b}, t_k^b) \in \Gamma \times (0, T], \end{cases} \quad \text{mean squared error of the boundary condition,} \tag{17}$$

$$\begin{cases} L_{iii} = \dfrac{\sum_K^{N_i} \left(R(x_k^t, t_k^k)\right)}{N}, \\ \qquad (\mathbf{x_k^r}, t_k^b) \in \Gamma \times (0, T], \end{cases} \quad \text{mean squared error of the residual of the PDE.}$$

Here we introduce N_i, N_b, and N_r which are the number of initial, boundary, and residual collocation points, where Γ is the boundary of Ω and Z_k^i is the given initial condition at $(x_k^i, 0)$. The superscripts $(\bullet)^b$, $(\bullet)^i$, $(\bullet)^r$ stand for boundary, initial, and residual conditions of the PDE. In our study, we focus on the 1D equation (7) with the initial and boundary conditions given in Eq. (10).

Figure 1 shows schematically the fully connected neural network architecture.

4. SEQUENTIAL TRAINING WITH DYNAMIC TIME STEPPING

Unlike standard PINNs training, we train for the entire domain at once and discretize the time domain into several segments. Note that this strategy is similar to the numerical methods used in scientific computing, where space-time problems are typically harder to solve than time-marching methods. Figure 2 illustrates the sequential training scheme versus the standard training scheme.

In a dynamic time-stepping scheme, the first time step $[0, T_1]$ is solved with a similar loss function as std-PINNs in Eq. (16).

For all the subsequent time segments, we utilize the following loss function:

$$L_{\text{tot}}^{\Delta T_n} = L_i + L_{ii} + L_{iii} + L_{iv}, \tag{18}$$

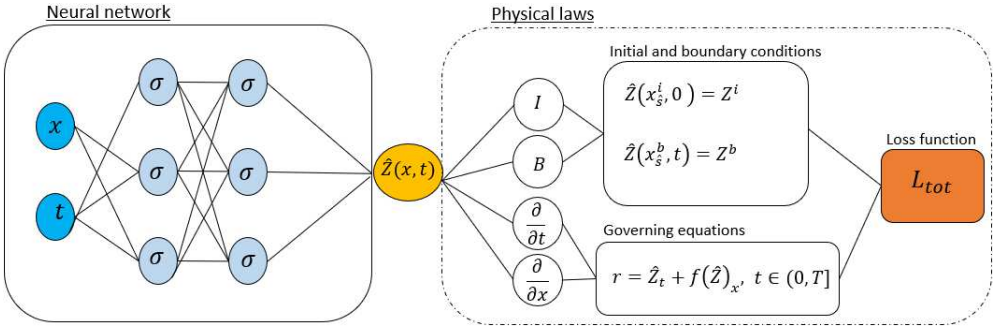


FIG. 1: Standard PINNs architecture

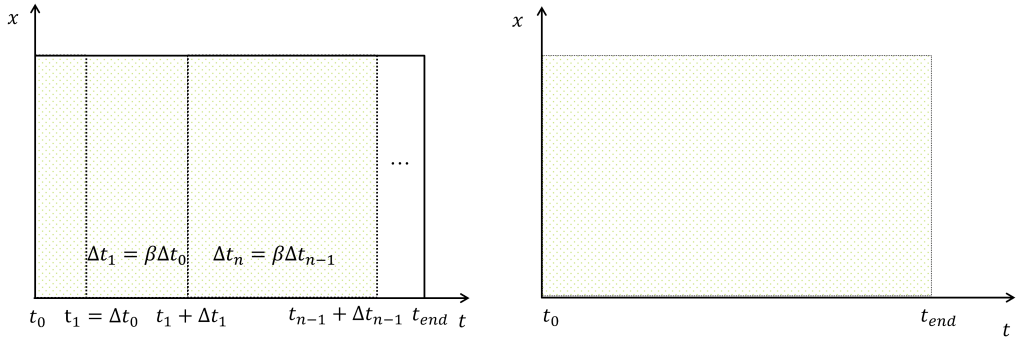


FIG. 2: Training scheme: (a) dynamic sequential time-stepping scheme; (b) standard training scheme

$$
\begin{cases}
L_i = \dfrac{\sum_K^{N_i} \tilde{Z}(x_k^i, T_{n-1}) - Z_k^i}{N_i}, \\
\quad x_k^i \in \Omega, & \text{mean squared error of the initial condition,} \\
L_{ii} = \dfrac{\sum_K^{N_b} \tilde{Z}(x_k^b, t) - Z(x_k^b, t_k)}{N_b}, \\
\quad (\mathbf{x_k^b}, t_k^b) \in \Gamma \times (T_{n-1}, T_n], & \text{mean squared error of the boundary condition,} \\
L_{iii} = \dfrac{\sum_K^{N_i} \left(R(x_k^t, t_k^k)\right)}{N}, \\
\quad (\mathbf{x_k^r}, t_k^r) \in \Gamma \times (T_{n-1}, T_n], & \text{mean squared error of the residual of the PDE,} \\
L_{iv} = \tilde{Z}(x,t) - \underline{\tilde{Z}}(x,t), \\
\quad (\mathbf{x_k^s}, t_k^s) \in \Omega \times (0, T_{n-1}], & \text{mean squared error of solution of previous time.}
\end{cases}
\tag{19}
$$

Here (x_k^i, T_{n-1}) is used to denote the collection of points where the calculation of the error on the initial condition is evaluated. $(\mathbf{x_k^b}, t_k^b)$ is the set of points where the error on the boundary conditions is calculated during the specified time interval $(T_{n-1}, T_n]$. The points on grid $(\mathbf{x_k^s}, t_k^s)$ store the solution obtained during the (n)th segment on the interval $(0, T_{n-1}]$, for its usage in the (n)th segment. By incorporating L_{iv} terms into the loss functions, the neural network can ensure backward compatibility (Mattey and Ghosh, 2022), meaning that the single neural network can

replicate the solution from all the previous time segments while solving the PDE for a specific time segment.

Next, we introduce a dynamic time-stepping scheme that instead of marching in time uniformly for training, we will make dynamic sequential training. The algorithm is as follows:

- We start with the small time step.
- If the loss function decreases as a predefined tolerance, we multiply the next time step to the fixed ratio β.
- If it fails to reach the tolerance, we divide the next time step by the same constant β.
- If the maximum time step $\Delta t_{\max}$ is reached, we keep it for further simulation.

Figure 3 shows the neural network of the sequential training with backward compatibility and dynamic time-stepping scheme over the interval $[t_{n-1}, t_n]$.

5. RESULTS

5.1 Single Timestep Training

To motivate our sequential approach, we consider the following transport equation with a given left and right boundary F_l and F_r:

$$R_{\mathrm{CO_2}} = S_{\mathrm{CO_2}} + \frac{\Delta t}{\Delta x}\big(F_r(S_{\mathrm{CO_2}}) - F_l\big). \tag{20}$$

We investigated the nonlinearity of the residual over four progressive time steps and analyzed the relationship between time-step size and nonlinearity. Figure 4 shows that as the time-step size increases, the nonlinearity of the residual also increases. This is due to the nonlinearity of the flux function, which plays a crucial role in determining the nonlinearity of the residual. The

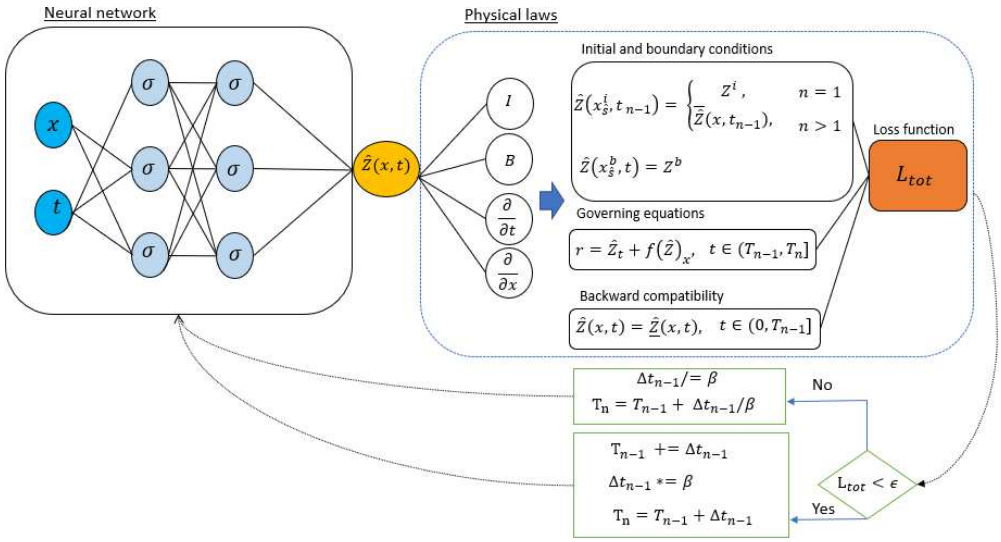
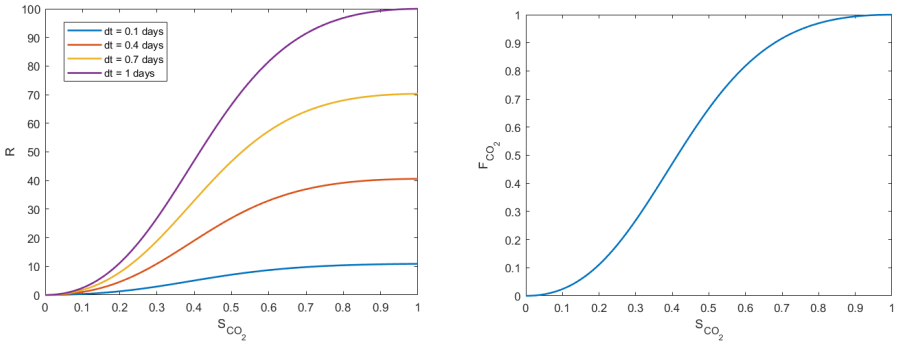


FIG. 3: Sequential PINNs architecture with backward compatibility and dynamic time-stepping scheme



(a) (b)

FIG. 4: (a) Residual for multiple time steps; (b) flux function

figure further demonstrates that the residual becomes more significant and more closely aligned with the flux function as the time-step size increases.

Here, we make a test case in which we train for a single progressive time step and observe the behavior of the loss function for the B-L everett problem. From Fig. 5, we can observe that the final value of the loss function evolves with the training for the larger time steps. We used fully connected neural networks with eight layers and each layer has 20 neurons. We use the L-BFGS-B optimizer. The training data points of the neural network are recorded in Table 1.

5.2 Full 1D Simulation Buckley–Leverett Test Case

Here we compare the solution of seq-PINN with dynamic time stepping and std-PINN for the two-phase immiscible B-L problem. Initially, the 1D domain is fully saturated by the nonwetting phase and we inject a wetting phase at the left boundary. We used fully connected neural

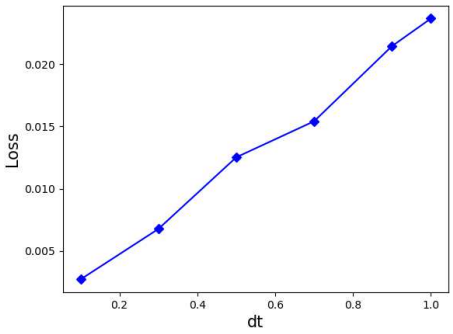


FIG. 5: Evolution of loss function for different time steps

TABLE 1: Description of training data

Variable	Description	Number
N_i	Initial condition points	300
N_b	Boundary condition points	300
N_r	Collocation points	10,000 $\times$ Δt

networks with eight layers and each layer has 20 neurons. We use the L-BFGS-B optimizer. The tolerance for the dynamic sequential time stepping is set as 3.5e–3 in this test case. Figure 6 compares the precise analytical solution to the solution predicted by the standard PINNs at time intervals $t = 0.1$, 0.3, and 0.7. As we can see, the standard PINNs cannot find the solution of the front accurately. Figure 7 compares the analytical solution to the solution of the PINNs with dynamic sequential training. As we can see, sequential training is capable of predicting the solution more accurately with respect to the standard training scheme. Figure 8 shows the full solution in time and space.

5.3 Full 1D Simulation Compositional Test Case

In the compositional test case, we see two nondifferentiable points related to phase changes in fractional flow formulations of compositional transport. In the two-phase region, the fractional

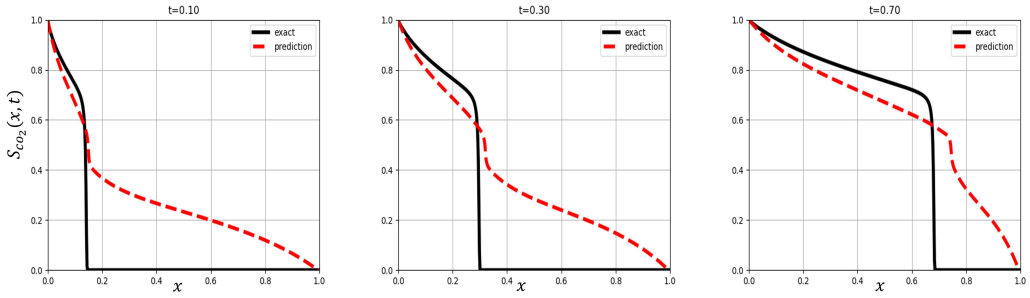


FIG. 6: Solution of the PINNs with standard training scheme

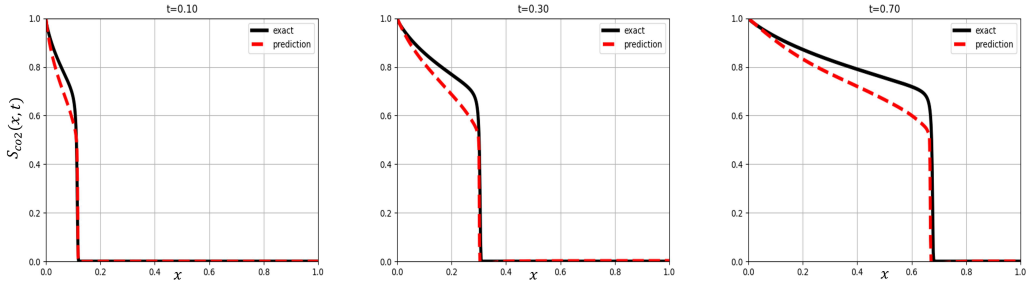


FIG. 7: Solution of the PINNs with sequential training scheme

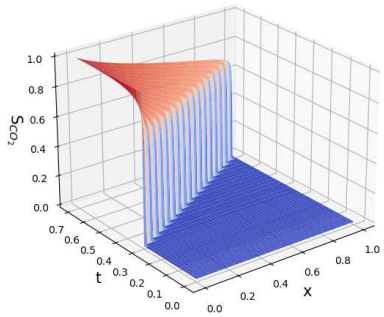


FIG. 8: Solution in spatiotemporal space

flow has an S-shaped curve, whereas it is linear in single-phase conditions. The discontinuous derivatives in the flow function can largely degrade the convergence of the loss function for the neural networks. To overcome this issue, we propose the smooth formulation of the fractional flow using the sigmoid step function:

$$F_{\text{smooth}} = H \times F + (1 - H) \times Z, \tag{21}$$

$$H = \text{Sigmoid}(S) - \text{Sigmoid}(S - 1), \tag{22}$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-\alpha S}}, \tag{23}$$

where F is the nonsmooth fractional flow given by Eq. (8). The accuracy of the smooth approximation highly depends on the parameter α which indicates how steep the sigmoid and step function could be. From Fig. 9 we can observe that the higher the α, the better the approximation.

5.3.1 Miscible Test Case

Here we test the sequential PINN training on miscible fluids with phase behavior controlled by constant K-values with $K = \{2.5, 0.3\}$. Initially, the 1D domain is fully saturated by the nonwetting phase and we inject a wetting phase at the left boundary. We use $\alpha = 20$ for the sigmoid approximation of the fractional flow as a trade-off between loss and accuracy. In this particular test case, the tolerance value for the dynamic sequential time stepping is set to 7e–3. Figures 10 and 11 show the solution of the standard training versus sequential training, respectively. We can observe that the sequential training scheme can predict the solution better than the standard training scheme.

5.3.2 Near-Immiscible

We test sequential PINNs training on quasi-immiscible compositional transport with phase behavior regulated by constant K-values of 2 and 0.002. The nonwetting phase first completely saturated the 1D domain, and we inject a wetting phase at the left border. For the sigmoid approximation, we use $\alpha = 20$. In this particular test case, the tolerance value for the dynamic sequential time stepping is set to 7e–3. Figures 12 and 13 illustrate and compare the PINNs solution trained standard and sequentially versus the analytical solution for three different times

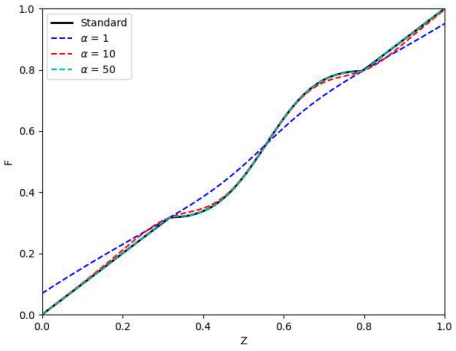


FIG. 9: Standard fractional flow versus smooth approximation of fractional flow using a sigmoid function with different alpha coefficient

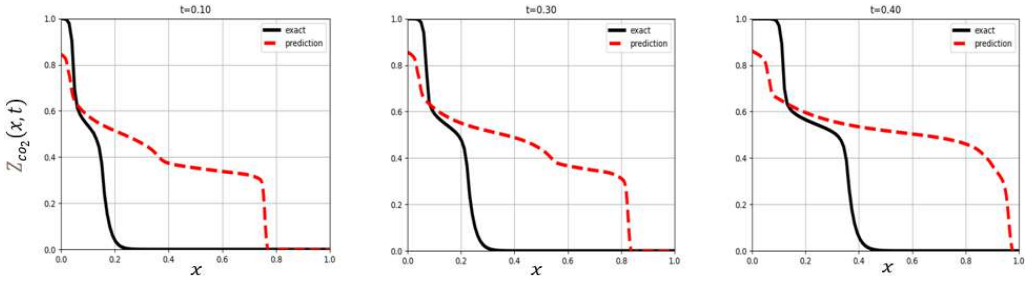


FIG. 10: Solution of the PINNs with standard training scheme and dynamic time stepping

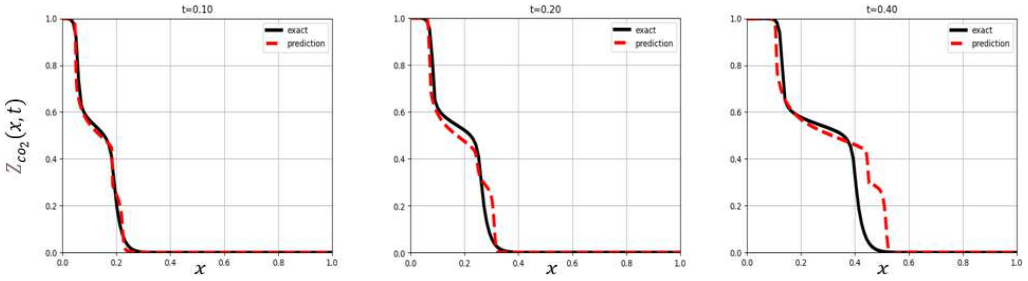


FIG. 11: Solution of the PINNs with sequential training scheme and dynamic time stepping

$t = 0.1$, 0.3, and 0.7. We can observe that the sequential training scheme can capture the shock more accurately with respect to the standard training scheme. However, there is still a small difference between the analytic solution and the PINN solution with the sequential training scheme.

6. CONCLUSIONS

CCUS are required in addition to innovative low-carbon energy solutions to mitigate global warming. For the simulation of CO_2 use and storage (CCUS) in subsurface reservoirs with complicated heterogeneous structures, a model that includes multiphase compositional flow and transport is needed. We investigated the application of a PINN for a two-phase fluid in porous media. While standard PINNs have difficulties solving hyperbolic PDEs with nonconvex flux functions, we suggested a sequential training scheme as an alternative. We can overcome this obstacle by training for shorter time intervals and marching in time dynamically. The sequential

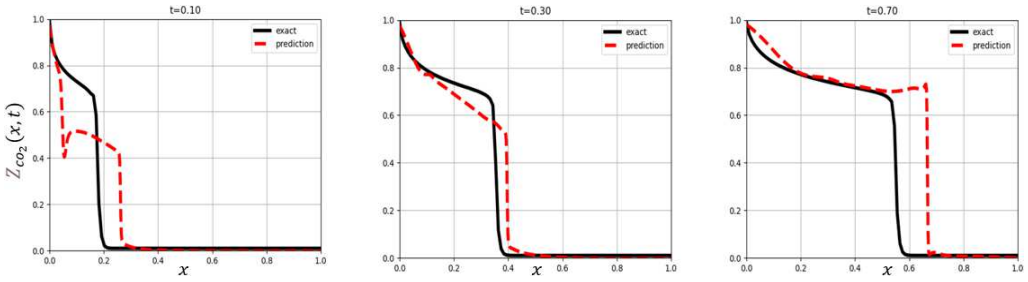


FIG. 12: Solution of the PINNs for a compositional immiscible test case with standard training scheme and dynamic time stepping

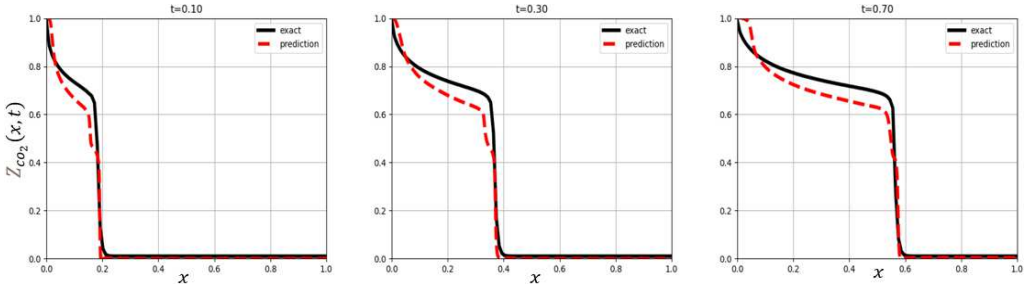


FIG. 13: Solution of the PINNs for a compositional immiscible test case with sequential training scheme and dynamic time stepping

training scheme begins with a small time step, and if the loss function decreases to a predefined tolerance, we multiply the next time step by a constant parameter, denoted as β. However, if the loss function fails to decrease to the tolerance within the given number of epochs, we divide the time interval by β. This adaptive approach allows for efficient convergence to an accurate solution while avoiding unnecessary computations and minimizing computational costs. Furthermore, we extended the sequential training strategy to the miscible binary compositional test case, where there are kink points that made it even more difficult for the optimizer to find the solution. To address this, we proposed a sigmoid function to continuously approximate the fractional flow function.

Within our study we conducted several numerical test cases to evaluate the effectiveness of our proposed sequential training scheme for simulating CO_2 use and storage in subsurface reservoirs with complex heterogeneous structures. Firstly, we designed a single-time-step numerical test case to demonstrate that increasing the time step leads to an increase in the final loss function. This observation is attributed to the heightened nonlinearity of the residual as the time step is increased, resulting in a more challenging optimization problem that negatively impacts the performance of the optimizer. Subsequently, we conducted a 1D full simulation test case to compare the standard and sequential training schemes for both immiscible and miscible test cases. Our results show that the proposed seq-PINNs approach outperforms the standard approach in terms of solution capture accuracy.

Our study identifies several promising directions for future research. Firstly, an important avenue for further exploration would be to extend the neural network to an arbitrary number of components, thereby enhancing the model's capacity to simulate complex multiphase compositional flows. Secondly, another possible future project would be to extend the model to higher dimensions and predict the solution in 2D and 3D space with heterogeneous reservoir structures.

ACKNOWLEDGMENT

We would like to acknowledge David Bruhn for his helpful feedback and suggestions on this project during our discussion.

REFERENCES

Alpak, F.O., A Mimetic Finite Volume Discretization Method for Reservoir Simulation, *SPE J.*, vol. **15**, no. 2, pp. 436–453, 2010.

Collins, D., Nghiem, L., Li, Y.K., and Grabenstetter, J., Efficient Approach to Adaptive-Implicit Compositional Simulation with an Equation of State, *SPE Reservoir Eng. (Soc. Pet. Eng.)*, vol. **7**, no. 2, pp. 259–264, 1992.

De Florio, M., Schiassi, E., Ganapol, B.D., and Furfaro, R., Physics-Informed Neural Networks for Rarefied-Gas Dynamics: Thermal Creep Flow in the Bhatnagar–Gross–Krook Approximation, *Phys. Fluids*, vol. **33**, no. 4, p. 047110, 2021.

Fraces, C.G. and Tchelepi, H., Physics Informed Deep Learning for Flow and Transport in Porous Media, arXiv:2104.02629, 2021.

Fuks, O. and Tchelepi, H.A., Limitations of Physics Informed Machine Learning for Nonlinear Two-Phase Transport in Porous Media, *J. Mach. Learn. Model. Comput.*, vol. **1**, no. 1, pp. 19–37, 2020.

Jagtap, A.D., Kharazmi, E., and Karniadakis, G.E., Conservative Physics-Informed Neural Networks on Discrete Domains for Conservation Laws: Applications to Forward and Inverse Problems, *Comput. Methods Appl. Mech. Eng.*, vol. **365**, p. 113028, 2020.

Jagtap, A.D., Mao, Z., Adams, N., and Karniadakis, G.E., Physics-Informed Neural Networks for Inverse Problems in Supersonic Flows, *J. Comput. Phys.*, vol. **466**, p. 111402, 2022a.

Jagtap, A.D., Mitsotakis, D., and Karniadakis, G.E., Deep Learning of Inverse Water Waves Problems Using Multi-Fidelity Data: Application to Serre–Green–Naghdi Equations, *Ocean Eng.*, vol. **248**, p. 110775, 2022b.

Jenny, P., Tchelepi, H.A., and Lee, S.H., Unconditionally Convergent Nonlinear Solver for Hyperbolic Conservation Laws with S-Shaped Flux Functions, *J. Comput. Phys.*, vol. **228**, no. 20, pp. 7497–7512, 2009.

Ji, W., Qiu, W., Shi, Z., Pan, S., and Deng, S., Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics, *J. Phys. Chem. A*, vol. **125**, no. 36, pp. 8098–8106, 2021.

Jiang, J. and Pan, H., Dissipation-Based Nonlinear Solver for Fully Implicit Compositional Simulation, *SPE J.*, vol. **27**, no. 4, p. SPE-209233-PA, 2022.

Kim, S., Ji, W., Deng, S., Ma, Y., and Rackauckas, C., Stiff Neural Ordinary Differential Equations, *Chaos: Interdiscip. J. Nonlinear Sci.*, vol. **31**, no. 9, p. 093122, 2021.

Krishnapriyan, A.S., Gholami, A., Zhe, S., Kirby, R.M., and Mahoney, M.W., Characterizing Possible Failure Modes in Physics-Informed Neural Networks, arXiv:20109.01050, 2021.

Lecun, Y., Bengio, Y., and Hinton, G., Deep Learning, *Nature*, vol. **521**, no. 7553, pp. 436–444, 2015.

Lou, Q., Meng, X., and Karniadakis, G.E., Physics-Informed Neural Networks for Solving Forward and Inverse Flow Problems via the Boltzmann-BGK Formulation, *J. Comput. Phys.*, vol. **447**, p. 110676, 2021.

Mao, Z., Jagtap, A.D., and Karniadakis, G.E., Physics-Informed Neural Networks for High-Speed Flows, *Comput. Methods Appl. Mech. Eng.*, vol. **360**, p. 112789, 2020.

Mattey, R. and Ghosh, S., A Novel Sequential Method to Train Physics-Informed Neural Networks for Allen Cahn and Cahn Hilliard Equations, *Comput. Methods Appl. Mech. Eng.*, vol. **390**, p. 114474, 2022.

Michelsen, M.L., The Isothermal Flash Problem. Part II. Phase-Split Calculation, *Fluid Phase Equilibria*, vol. **9**, no. 1, pp. 21–40, 1982.

Pour, K.M., Voskov, D., and Bruhn, D., Nonlinear Solver Based on Trust Region Approximation for CO_2 Utilization and Storage in Subsurface Reservoir, *Geoenergy Sci. Eng.*, vol. **225**, p. 211698, 2023.

Raissi, M., Perdikaris, P., and Karniadakis, G., Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations, *J. Comput. Phys.*, vol. **378**, pp. 686–707, 2019.

Rodriguez-Torrado, R., Ruiz, P., Cueto-Felgueroso, L., Green, M.C., Friesen, T., Matringe, S., and Togelius, J., Physics-Informed Attention-Based Neural Network for Hyperbolic Partial Differential Equations: Application to the Buckley–Leverett Problem, *Sci. Rep.*, vol. **12**, no. 1, p. 7557, 2022.

Tipireddy, R., Barajas-Solano, D.A., and Tartakovsky, A.M., Conditional Karhunen-Loève Expansion for Uncertainty Quantification and Active Learning in Partial Differential Equation Models, *J. Comput. Phys.*, vol. **418**, p. 109604, 2020.

Voskov, D. and Tchelepi, H., Comparison of Nonlinear Formulations for Two-Phase Multi-Component EoS Based Simulation, *J. Pet. Sci. Eng.*, vols. **82–83**, pp. 101–111, 2012.

Yang, L., Meng, X., and Karniadakis, G.E., B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data, *J. Comput. Phys.*, vol. **425**, p. 109913, 2021.