Received: 04 January 2024 | Revised: 08 March 2024 | Accepted: 12 March 2024

https://doi.org/10.56946/jce.v3i1.345

Journal of Chemistry and Environment

Research Article

Sequencing Initial Conditions in Physics-Informed Neural Networks

Saman Hooshyar[1], Arash Elahi[2,*]

[1]Department of Computer Science, University of Illinois Chicago, Chicago, IL, 60607, USA

[2]Department of Chemical Engineering, University of Illinois Chicago, Chicago, IL, 60607, USA

*Correspondence: aelahi4@uic.com

Abstract

The scientific machine learning (SciML) field has introduced a new class of models called physics-informed neural networks (PINNs). These models incorporate domain-specific knowledge as soft constraints on a loss function and use machine learning techniques to train the model. Although PINN models have shown promising results for simple problems, they are prone to failure when moderate level of complexities are added to the problems. We demonstrate that the existing baseline models, in particular PINN and evolutionary sampling (Evo), are unable to capture the solution to differential equations with convection, reaction, and diffusion operators when the imposed initial condition is non-trivial. We then propose a promising solution to address these types of failure modes. This approach involves coupling Curriculum learning with the baseline models, where the network first trains on PDEs with simple initial conditions and is progressively exposed to more complex initial conditions. Our results show that we can reduce the error by 1 – 2 orders of magnitude with our proposed method compared to regular PINN and Evo.

Keywords: Scientific machine learning PINN, soft-regularization, multiphysics modeling, chemical engineering PDEs,

1. Introduction

Partial differential equations (PDEs) are frequently adopted to explain various occurrences in realms of science and engineering, generally founded on fundamental laws such as the conservation of mass or energy. Typically, finding analytical solutions to these PDEs for many real-world settings is not trivial and, in some cases, not feasible. Many conventional approaches have been proposed and studied throughout years ,e.g., finite element methods (FEM) [1], Gradient Discretization method [2], Spectral method [3], etc. to approximate the solution to PDEs numerically. However, these solutions can be computationally expensive since they involve discretizing the problem domain into a grid and updating the solution at each grid point. This can require many calculations and iterations, especially for complex problems such as turbulence simulations [4]. For this reason, as well as the availability enormous data in scientific and engineering domains, there has been an increasing interest in developing machine learning (ML)/Deep Learning (DP) methods to solve complex partial differential equations or complement numerical solutions. Thus, the area of Scientific Machine Learning (SciML) has emerged, integrating traditional scientific models based on differential equations with data-driven ML techniques, such as neural network training.

One of these methods is the so-called physics-informed neural networks (PINNs) [5-9]. The basic idea of PINNs for solving a forward PDE is to train a neural network to minimize errors with respect to the solution provided at initial/boundary points of a spatiotemporal domain, as well as the PDE residuals observed over a sample of interior points, referred to as collocation points. Due to the capability of PINNs to incorporate physical laws and their ability to provide a flexible structure for the solution PDEs, they have been extensively utilized for the multiphysics modeling of systems in the field of chemical engineering. For example, PINNs have been adopted to model the systems related to heat transfer [10-12], compressible and incompressible flows [13-18], convection, reaction, and advection-diffusion systems [19-25]. The applications of the PINN method have also been extended to study of environmental and materials engineering systems, such as, mitigation of carbon emissions [26-29], and prediction of materials properties [30-32].

Despite the advantages PINNs offer, several recent studies show that training PINNs can be quite challenging for complicated systems [33-36]. In general, PINNs try to leverage the power of deep neural networks to learn the behavior of complex systems while respecting the underlying physical laws. This is achieved by incorporating the governing equations or physical laws as a soft constraint on the loss function, that is then minimized using ML techniques. However, solving the optimization problem may not be straightforward as the imposed physical term in the loss function often involves nonlinearities that cause the loss function to be ill-conditioned [37-39]. Several works propose novel methods to tackle the challenges of training PINNs [40-42]. One early work identifies a mode of failure of PINNs due to the existence of unbalanced gradients during training and proposes an adaptive model that utilizes gradient statistics to assign appropriate weights to different terms in the PINNs composite loss function [43]. Karishnapriyan et al. [33] describes curriculum regularization and sequence-to-sequence (seq2seq) learning as two promising solutions to address failure modes associated with large PDE coefficients. The importance of sampling strategies on the performance of PINNs has been the focus of many researchers. Subramanian et al. [35] argues that the location of collocation points greatly influences the trainability of PINNs, motivating the development of an adaptive collocation scheme that progressively accumulates more collocation points around areas where the model yields higher errors. In another recent line of work [43], it was shown that the PINN's performance depends on the successful propagation of solution for boundary/initial points to the interior points. To mitigate the "propagation failure" they proposed the so-called evolutionary sampling (Evo) strategy, where collocation points evolve over training iterations to prioritize high-density regions. In contrast, the work of Wang et al. [44] demonstrates the rapid transition in the transition layer as the cause of failure and introduces a curriculum-based approach that encourages neural networks to prioritize the learning on easier non-layer regions.

In all the aforementioned literature, the initial conditions are assumed to be fixed during training and PINN has to be retrained for problems with different initial conditions. Conventional methods for handling the complex initial conditions face significant challenges, as they typically require a fine resolution to capture the steep gradients. In such cases, the traditional methods often lead to extensive computational costs and numerical instabilities. For example, in the coating process of semiconductors and MEMS devices, the final thickness of resist film is predicted using numerical simulations based on the governing equations of Liquid film flows [45, 46]. These problems are highly sensitive to the initial condition and performing multiple simulations for different initial conditions is not cost-effective. Problems related to fluid dynamics and heat transfer, such as high-speed aerodynamic flows [47], biomedical flows [48], and estimation of air pollution in a spatiotemporal domain [49], are mostly modeled using convection, diffusion, and reaction PDEs. Changing the initial condition when training PINN for these problems may lead to a significant deviation of the predicted solution from the ground truth. Therefore, it is critical to investigate new approaches to improve the robustness of the model against variation in initial condition. Motivated by this, we propose to combine the existing baseline models "PINN" and "Evo" [43] with "Curriculum Regularization" [33], where the neural network first trains with easier initial conditions and progressively approaches the target initial condition, which could be hard to optimize from the beginning.

2. Methods

The formulation of PINNs starts with constructing a neural network $f_\theta(x,t)$ to deduce the solution u of a non-linear partial differential equation [43]:

$$u_t + \mathcal{N}_x[u] = 0,\ x \in \mathcal{X},\ t \in [0,T];\ \ u(x,0) = h(x),\ x \in \mathcal{X};\ \ u(x,t) = g(x,t),\ \in t[0,T],\ x \in \partial\mathcal{X}$$

Here, $\mathcal{N}_x$ is non-linear spatial operator, x and t denotes space and time, respectively, $\partial\mathcal{X}$ is boundary of spatial domain, and T is the time horizon. $h(x)$ is the initial condition and $g(x,t)$ is the boundary condition. To solve the PDE, we first need to compute the residual function $\mathcal{R}_\theta(x,t)$ and the corresponding loss function $\mathcal{L}_r(\theta)$ on a set of collocation points $\{x_i, t_i\}_{i=1}^{N}$ sampled from a uniformly from the entire spatio-temporal domain ($\Omega = \mathcal{X} \times [0,T]$).

$$\mathcal{R}_\theta(x,t) = \frac{\partial}{\partial t} f_\theta(x,t) + \mathcal{N}_x[f_\theta(x,t)], \tag{1}$$

$$\mathcal{L}_r(\theta) = \frac{1}{N}\sum_{i=1}^{N} [\mathcal{R}_\theta(x_i,t_i)]^2. \tag{2}$$

where N is the number of collocation points. PINNs approximate the solution of given PDE by minimizing the overall mean-squared losses consists of $\mathcal{L}(\theta) = \lambda_r\mathcal{L}_r(\theta) + \lambda_{ic}\mathcal{L}_{ic}(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta)$. The subscripts "$r$", "$ic$", and "$bc$" corresponds to the residual, initial condition, and boundary condition, respectively. The hyperparameter λ signifies the importance of each loss term on the overall loss function. Please note that the main complication in training PINNs arise from the existence of the differential operator in $\mathcal{L}(\theta)$, which causes the loss function to be ill-conditioned. This is very different from norm based L_1 and L_2 regularizations where the regularization operator corresponds to a simple convex function.

2.1. Convection System

We consider a one-dimensional convection problem with the following governing equation

$$\frac{\partial u}{\partial t} + \beta\frac{\partial u}{\partial x} = 0, \quad x \in \mathcal{X},\ t \in [0,T]. \tag{3}$$

where β is the convection coefficient. The initial and periodic boundary conditions are as follows:

$$h(x) = \sin(\alpha x + k\pi), \qquad u(0,t) = u(2\pi,t). \tag{4}$$

α is the rate of change of the function and k is the phase. The general loss function can be obtained as follow:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} [\lambda_{ic}(\hat{u} - h(x_i))^2 + \lambda_r(\frac{\partial\hat{u}}{\partial t} + \beta\frac{\partial\hat{u}}{\partial x})^2 + \lambda_{bc}(\hat{u}(0,t) - \hat{u}(2\pi,t))^2]. \tag{5}$$

where $\hat{u} = f_\theta(x,t)$ presents the neural network's output.

2.2. Reaction-Diffusion System

The one-dimensional reaction-diffusion problem can be described using the following governing equation:

$$\frac{\partial u}{\partial t} - \nu\frac{\partial^2 u}{\partial x^2} - \rho u(1-u) = 0, \quad x \in \mathcal{X},\ t \in (0,T] \tag{6}$$

where $\nu > 0$ and ρ are the diffusion and reaction coefficients, respectively. We consider Gaussian distribution for the initial condition and a periodic boundary condition:

$$h(x) = e^{-\frac{1}{2}(\frac{x-\pi}{\pi/\eta})^2}, \qquad u(0,t) = u(2\pi,t). \tag{7}$$

Here, $\frac{\pi}{\eta}$ (= σ) is the standard deviation and η is a constant used to scale σ. The overall loss function for this problem is given as:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} [\lambda_{ic}(\hat{u} - h(x_i))^2 + \lambda_r(\frac{\partial\hat{u}}{\partial t} - \nu\frac{\partial^2\hat{u}}{\partial x^2} - \rho u(1-\hat{u}))^2 + \lambda_{bc}(\hat{u}(0,t) - \hat{u}(2\pi,t))^2]. \tag{8}$$

Table 1. Hyperparameters applied in experiments on different PDE system.

PDE	Method	$\lambda_r/\lambda_{ic}/\lambda_{bc}$	lr.scheduler
Convection	PINN PINN + Curriculum	1/1/1	No

	Evo
	Evo + Curriculum
		1/100/100	StepLR
			Rate = 0.9
			Steps = 5000

Reaction-Diffusion	PINN PINN + Curriculum	1/100/100	No

2.3. Experiment Setup

We first perform experiments on the time-dependent convection system using four different baselines: “PINN”, “PINN + Curriculum”, “Evo”, and “Evo + Curriculum”. We consider two different convection coefficients (β = 5 and β = 15) and the initial condition $\sin(\alpha x + k\pi)$ with α ranging from 1 to 5 and k ranging from 0 to 0.5. We then study the reaction-diffusion problem ($\nu = \rho = 3$) using “PINN” and “PINN + Curriculum” baselines. The η parameter in the initial condition in equation (7) varies from 2 to 8. The neural network architecture consists of four fully connected layers with 50 neurons per layer and a hyperbolic tangent activation function. For all cases, we use a periodic boundary condition and Adam optimizer with a learning rate *lr* = 1e-3. After training the models, we obtain the L_2 absolute error between the predicted result and the analytical solution (ground truth) as:

$$\text{Absolute error} = \frac{1}{N}\sum_{i=0}^{N}\|\hat{u} - u\|_2 \qquad (9)$$

where u is the exact solution and $\hat{u}$ is the output of *NN*. The number of collocation points is kept constant at N = 1000 over the whole domain Ω ($n_x \times n_t = 512 \times 256$). Note that n_x and n_t denotes the number of grid points in spatial and temporal domains, respectively. Table 1 provides a list of other hyperparameter settings for different baseline method.

3. Results and discussion

The concept behind curriculum learning [50], inspired by human education, is to start optimizing the problem with easier training criterion and gradually increase the level of difficulty over time. Curriculum learning has been shown to improve various ML models for different applications including PINNs for solving PDEs with large coefficients. Motivated by this, we propose to implement Curriculum learning in predicting the solution to problems with complex initial conditions. The main idea is to train the base model using a simple initial condition and progressively transition to a more difficult initial condition after a certain number of iterations. This way, the model has the opportunity to learn the easier constraint and construct a solid foundation for learning the target constraint.

For the one-dimensional convection problem with an initial condition given in Eq. 4 ($\sin(\alpha x + k\pi)$), the baseline PINN model is trained with and without Curriculum learning for different values of the constants in the initial condition. After training, we measured the absolute errors between the analytical and predicted solution using Eq. 9. Please note that the data points associated to PINN (denoted with hollow circles) in Figure1 are obtained by running the code for 2.5 × 10^4 iterations for each distinct value of the specified constant (α and k), whereas the data points associated to PINN + Curriculum are obtained for that many iterations over the whole range of the initial condition constant. For example, at k = 0 in Figure 1(a), PINN trains for 2.5 × 10^4 iterations, but Curriculum learning trains for 5 × 10^3. This clearly grants an unfair advantage to the “Vanilla” PINN model.

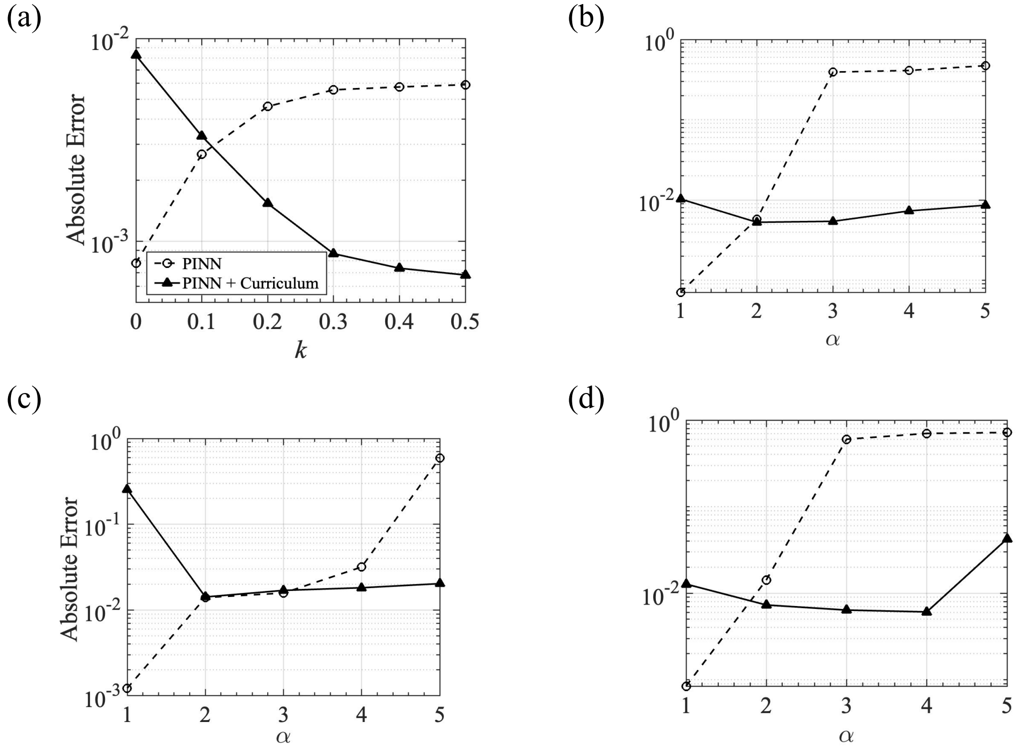

Figure 1. Variation of Absolute error with initial condition parameters for "PINN" and "PINN + Curriculum" models. The PDE is a convection problem with (a) β = 5, α = 1, (b) β = 5, k = 0, (c) β = 5, k = 0.5, and (d) β = 15, k = 0.

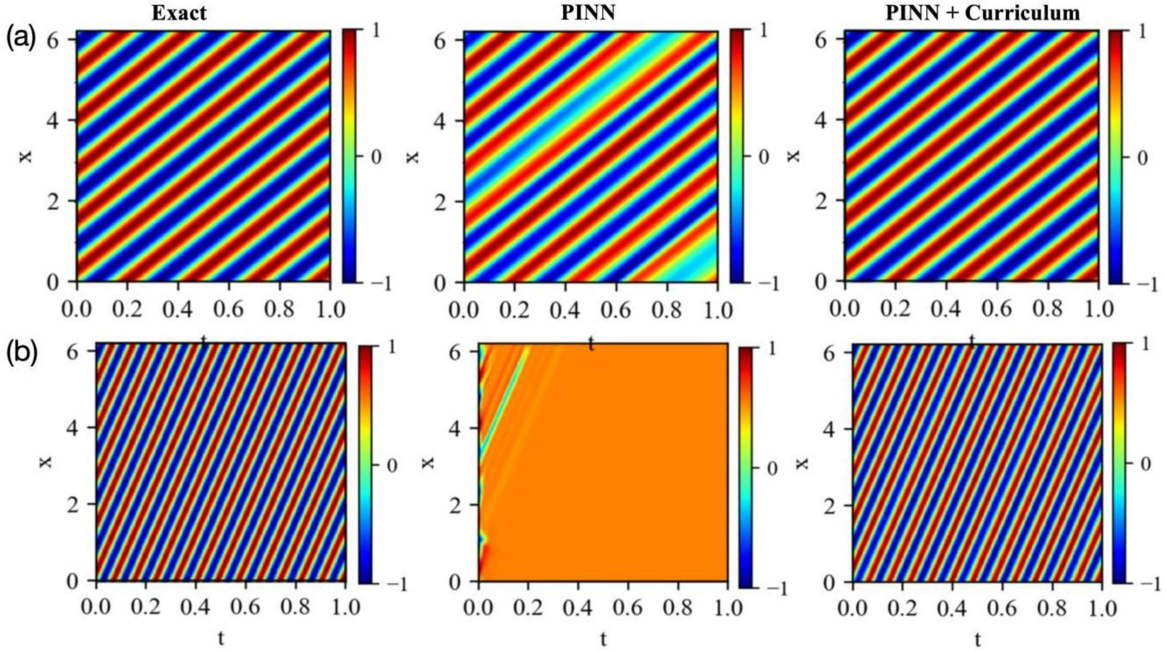

Figure 2. Predicting the solution to a 1D convection problem using "PINN" and "PINN + Curriculum" baseline models. The other parameters are (a) β = 5, α = 5 and (b) β = 15, α = 5. k = 0 for both cases

The variation of absolute error as a function of phase angle k for a convection system with β = 5 and α = 5 is depicted in Figure 1(a). As one can see, the error corresponding to the model with Curriculum learning decreases with increasing k while the error rises for the PINN model. Although the value of k influences the trainability of the network, its effect is not as notable as the effect of angular frequency α. Figures 1(b) & (c) show the trends of absolute error with α increasing from 1 to 5, when k = 0 & 0.5, respectively. As expected, the Curriculum model outperforms the PINN model for complex initial conditions (large values of α). The reason that PINN delivers better performance for small values of α is the unfair advantage it has over Curriculum. For example, at α = 1, PINN trains for 2.5 × 10^4, but Curriculum only trains for 5 × 10^3; hence, obtaining lower error for PINN. The predicted solutions of these models as well as the exact solution when β = 5, α = 5, and k = 0 are reported in Figure 2(a). It can be seen that, unlike PINN, Curriculum method successfully captures the solution on the entire spatiotemporal domain. To test the robustness of our proposed model, we even add further complexity to the problem by increasing the convection strength to β = 15. Similar to the previous case, Curriculum learning notably improves the performance as the absolute error drops by around 2 orders of magnitude for intermediate values of α and 1 order of magnitude for α = 5, see Figure 1(d).

.

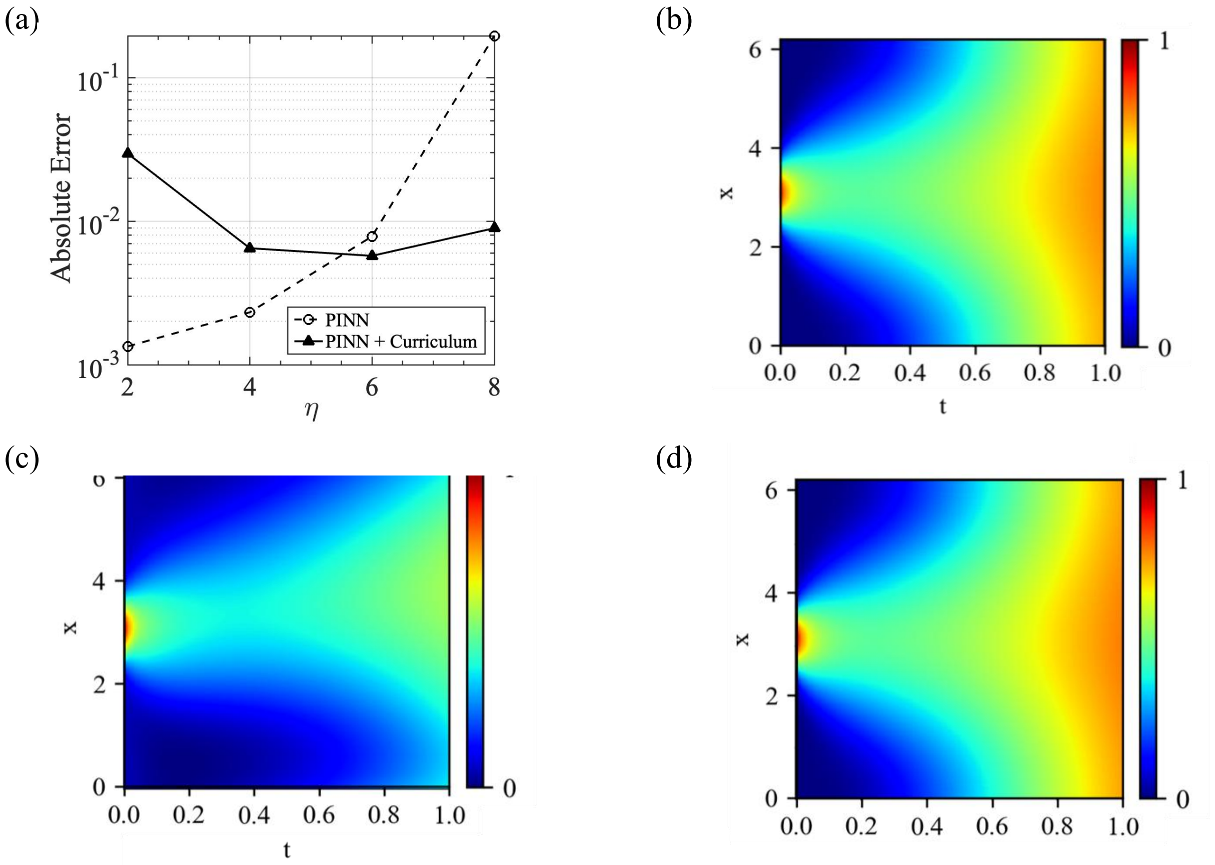

Figure 3. (a) Variation of Absolute error with initial condition parameters. (b) The exact solution and predicted solution using (c) “Evo”, and (d) “Evo + Curriculum” baseline models to a 1D reaction-diffusion problem with ν = ρ = 3 and η = 8.

The visualizations of the predicted solutions in Figure 2(b) show that the “Vanilla” PINN model clearly fails at learning the solution; however, combining PINN and Curriculum learning results in an accurate prediction.

We next look at a one-dimensional reaction-diffusion flow with a Gaussian initial condition, see Eq. 7. Here, we consider four different values for η ranging from 2 to 8. In general, as η increases the diffusivity of the flow rises, making the problem more difficult to learn. The variation of absolute error with η is illustrated in Figure 3(a) for a case with $\nu = \rho = 3$. One can clearly see that at the extreme case $\eta = 8$, the Curriculum learning lowers the absolute error of PINN from 1.96 × 10^{-1} to 8.94 × 10^{-3}, improving it by almost 2 orders of magnitude. We also show the predicted solutions and the ground truth solution obtained using analytical methods in Figures 3(b) – (d). It is obvious that the PINN model is uncapable of predicting the reaction or the diffusion components. However, by first exposing the network to the easier problem ($\eta = 2$) and gradually increasing η, we were able to capture the solution in the whole domain.

Inspired by algorithms used for biological evolution, this iterative sampling strategy was developed to address the propagation failure of collocation points when solving a PDE having high residuals in very narrow regions. In this method, we first generate collocation points through a uniform distribution. Then, throughout each subsequent iteration, we retain collocation points whose absolute value of its PDE residual is greater than a predefined threshold and resample the remainder points from a uniform distribution. We finally merge the resampled population with the retained population to create the population for the next iteration. The main idea behind Evo sampling is to include more collocation points from high PDE residual regions to embolden the representation of these regions in the overall residual loss.

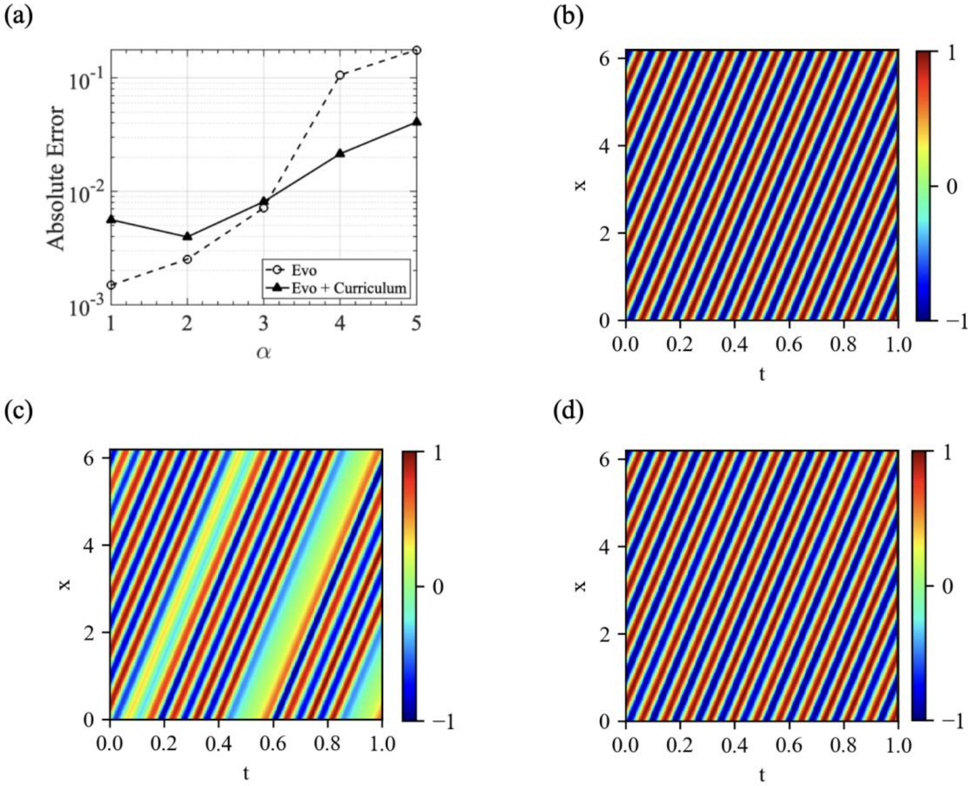

Figure 4. (a) Variation of Absolute error with initial condition parameters. (b) The exact solution and predicted solution using (c) “Evo”, and (d) “Evo + Curriculum” baseline models to a 1D convection problem with $\beta = 15$, $\alpha = 5$, $k = 0$.

Here, we again consider a one-dimensional convection problem with $\beta = 15$ and $k = 0$. We examined this scheme by utilizing the identical NN architecture as before. Compared to PINNs, Evo sampling methods require a higher number of iterations to converge, therefore, we trained the model for 10^5 epochs (four times the number of epochs for the PINN method). Similar to the previous section, we give an advantage to the Evo method over Evo + Curriculum method in terms of number of iterations. Figure 4(a) shows that employing Curriculum learning on top of Evo sampling causes the absolute error to drop for larger values of angular frequency α. Moreover, the exact and predicted solutions are depicted in Figures 4(b) – (d). We observe that Evo + Curriculum learning method, unlike Evo sampling, successfully captures the solution in the entire domain. Comparing PINN + Curriculum and Evo + Curriculum methods, we see that the latter produces slightly more accurate results, however, it should be noted that this comparison is biased in favor of Evo + Curriculum method since the total number of iterations for this method is considerably higher. In general, Evo is proven to outperform PINN when trained for a relatively high number of epochs, especially in the case of PDEs with very large coefficients (e.g., convection equation with $\beta > 30$)

4. Conclusion

SciML models, more specifically Physics-informed neural networks, present an exciting opportunity to extend the use of ML techniques to tackle a variety of scientific and engineering problems. However, incorporating ML approaches with PDE-based constraints served as a soft regularization term can result in failure modes that prevent the learning of fundamental physics governing a problem. We studied one-dimensional convection and reaction-diffusion problems and showed that the "Vanilla" PINN and Evo sampling models are unable to predict the solutions to these problems when we impose a non-trivial initial condition. We proposed implementing Curriculum learning where the baseline model trains on simple initial conditions before being exposed to the complex target initial condition. We showed this approach lowers the absolute error by 1 – 2 orders of magnitude and can successfully capture the solution to the PDEs. Addressing the limitations associated with SciML models will be crucial if we hope to build a closer integration between scientific theories and Machine Learning formulations.

Authors Contribution

Saman Hooshyar had the highest contribution in conducting the research and writing the paper. Arash Elahi enhanced data representation, provided suggestions, and revised the narrative.

Conflicts of Interest

There are no conflicts of interest reported by the writers.

Acknowledgment

I, Saman Hooshyar, the first author of this manuscript, would like to express my gratitude to Dr. Xinhua Zhang for the patient guidance, encouragement, and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I would also like to thank Dr. Abolfazl Asudeh for serving as my secondary committee member and providing his expertise in the review of this report.

Data Availability statement

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

Funding: Not applicable (N/A).

REFERENCES

1. Zienkiewicz O., Taylor R., Zhu J., 2005. The finite element method: its basis and fundamentals. Elsevier.
2. Droniou J., Le N., 2020. The gradient discretization method for slow and fast diffusion porous media equations. SIAM Journal on Numerical Analysis, 58(3), pp: 1965-1992.
3. Hussaini M., Zang T.A., 1987. Spectral methods in fluid dynamics. Annual Review of Fluid Mechanics, 19(1), pp: 339-367.
4. Stegmeir A., Ross A., Body T., Francisquez M., Zholobenko W., Coster D., Kang S., 2019. Global

turbulence simulations of the tokamak edge region with GRILLIX. Physics of Plasmas, 26(5), 052517.

5. Ainsworth M., Dong J., 2021. Galerkin neural networks: a framework for approximating variational equations with error control. SIAM J. Sci. Comput., 43, A2474-A2501.
6. Beck C., Hutzenthaler M., Jentzen A., Kuckuck B.J., 2020. An overview on deep learning-based approximation methods for partial differential equations.
7. Karniadakis G.E., Kevrekidis I.G., Lu L., Perdikaris P., Wang S., Yang L., 2021. Physics-informed machine learning. Nature Reviews Physics, pp: 422-440.
8. Raissi M., Perdikaris P., Karniadakis G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, pp: 686-707.
9. Gusmão G. S., Medford A. J., 2023. Maximum-likelihood Estimators in Physics-Informed Neural Networks for High-dimensional Inverse Problems. arXiv preprint arXiv:2304.05991.
10. Cai S., Wang Z., Wang S., Perdikaris P., Karniadakis G. E., 2021. "Physics-Informed Neural Networks for Heat Transfer Problems." ASME. J. Heat Transfer, June 2021; 143(6): 060801. https://doi.org/10.1115/1.4050542
11. Mishra S., Molinaro R. Physics informed neural networks for simulating radiative transfer. Journal of Quantitative Spectroscopy and Radiative Transfer, 2021, 270, 107705.
12. Chervonyi Y. et al. Semi-analytical industrial cooling system model for reinforcement learning. arXiv preprint arXiv:2207.13131, 2022.
13. Jagtap A.D. et al. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering, 2020, 365, 113028.
14. Cai S. et al. Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mechanica Sinica/Lixue Xuebao, 2021, 37(12), p. 1727-1738.
15. Hooshyar S. et al. The impact of imposed Couette flow on the stability of pressure-driven flows over porous surfaces. Journal of Engineering Mathematics, 2022, 132(1).
16. Hooshyar S. et al. Energy budget analysis of plane poiseuille-couette flow over a permeable surface. Bulletin of the American Physical Society, 2020.
17. Acharya N. et al. Instability analysis of poiseuille flow between two parallel walls partially obstructed by porous surfaces. In APS Division of Fluid Dynamics Meeting Abstracts, 2019, pp. NP05–140.
18. Elahi A. et al. Temperature-Transferable Coarse-Grained Model for Poly(propylene oxide) to Study Thermo-Responsive Behavior of Triblock Copolymers. J. Phys. Chem. B, 2022, 126, p. 292–307.
19. Pang G. et al. fPINNs: Fractional Physics-Informed Neural Networks. SIAM Journal on Scientific Computing, 2019, 41(4), A2603–A2626.
20. Aygün A. Physics informed neural networks for computational fluid dynamics. Middle East Technical University, Master's thesis, 2023.
21. Dekhovich A. et al. iPINNs: Incremental learning for Physics-informed neural networks. arXiv preprint arXiv:2304.04854, 2023.
22. Du H. et al. Modeling density-driven flow in porous media by physics-informed neural networks for CO2 sequestration. Computers and Geotechnics, 2023, 159, 105433.
23. Guo J. et al. Pre-training strategy for solving evolution equations based on physics-informed neural networks. arXiv preprint, 2022, arXiv:2212.00798.
24. Shi S., Liu D., Huo Z. Simulation of flow field in silicon single-crystal growth using physics-informed neural network with spatial information. Physics of Fluids, 2022, 34(11), 113610.

25. Elahi A., Chaudhuri S. Computational Fluid Dynamics Modeling of the Filtration of 2D Materials Using Hollow Fiber Membranes. ChemEngineering, 2023, 7, 108.
26. Ngo S.I., Lim Y. Solution and Parameter Identification of a Fixed-Bed Reactor Model for Catalytic CO2 Methanation using Physics-Informed Neural Networks. Catalysts, 2021, 11(11), 1304.
27. Ngo S.I., Lim Y. Forward Physics-Informed Neural Networks Suitable for Multiple Operating Conditions of Catalytic CO2 Methanation Isothermal Fixed-Bed. IFAC-PapersOnLine, 2022, 55(7), 429-434.
28. Ngo S.I., Lim Y. Modeling density-driven flow in porous media by physics-informed neural networks for CO2 sequestration. Computer Aided Chemical Engineering, 2022, pp. 1675-1680.
29. Du H., Zhao Z., Cheng H. Modeling Density-Driven Flow in Porous Media by Physics-Informed Neural Networks for CO2 Sequestration. Computers and Geotechnics, 2023, 159, 105433.
30. Batra R. et al. Emerging Materials Intelligence Ecosystems Propelled by Machine Learning. Nature Reviews. Materials, 2021, 6(8), 655-678.
31. Zhu Q., Liu Z., Yan J. Machine Learning for Metal Additive Manufacturing: Predicting Temperature and Melt Pool Fluid Dynamics using Physics-Informed Neural Networks. Computational Mechanics, 2021, 67(2), 619-635.
32. Bharadwaja B.V.S.S. et al. Physics-Informed Machine Learning and Uncertainty Quantification for Mechanics of Heterogeneous Materials. Integrating Materials and Manufacturing Innovation, 2022, 11(4), 607-627.
33. Krishnapriyan A. et al. Characterizing possible failure modes in physics-informed neural networks. Advances in Neural Information Processing Systems, 2021, 34, 26548-26560.
34. Münzer M., Bard C. A Curriculum-Training-Based Strategy for Distributing Collocation Points during Physics-Informed Neural Network Training. arXiv preprint, 2022, arXiv:2211.11396.
35. Subramanian S. et al. Adaptive Self-supervision Algorithms for Physics-informed Neural Networks. arXiv preprint, 2022, arXiv:2207.04084.
36. Cooley M. et al. Fourier PINNs: From Strong Boundary Conditions to Adaptive Fourier Bases. OpenReview, 2023, https://openreview.net/forum?id=40Mw2GJnlZ.
37. Edwards C. Neural networks learn to speed up simulations. Communications of the ACM, 2022, 65(5), 27–29.
38. Wang S. et al. Respecting causality is all you need for training physics-informed neural networks. arXiv preprint, 2022, arXiv:2203.07404.
39. Subramanian S. et al. Adaptive Self-supervision Algorithms for Physics-informed Neural Networks. arXiv preprint, 2022, arXiv:2207.04084.
40. Chen Y. et al. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. Optics Express, 2020, 28(8), 11618-11633.
41. Geneva N., Zabaras N. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. Journal of Computational Physics, 2020, 403, 109056.
42. Jin X. et al. NSFNNets (Navier-Stokes Flow Nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. Journal of Computational Physics, 2021, 426, 109951.
43. Daw A. et al. Mitigating propagation failures in PINNs using evolutionary sampling. arXiv preprint, 2022, arXiv:.
44. Wang S., Teng Y., Perdikaris P. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv preprint, 2020, arXiv:2001.04536.
45. Shiratori S. et al. Spatio-temporal thickness variation and transient Marangoni number in striations during

spin coating. International Journal of Heat and Mass Transfer, 2020, 154, 119678.

46. Shiratori S., Kubokawa T. Double-peaked edge-bead in drying film of solvent-resin mixtures. Physics of Fluids, 2015, 27, 102105.
47. Mao Z. et al. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering, 2020, 360, 112789.
48. Yin M. et al. Non-invasive inference of thrombus material properties with physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering, 2021, 375, 113603.
49. Sak H. et al. Modeling dependence dynamics of air pollution: Pollution risk simulation and prediction of PM2.5 levels. arXiv preprint, 2016, arXiv:1602.05349.
50. Bengio Y. et al. Curriculum learning. Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 41-48.

How to cite this article:
Hooshyar S, Elahi A.. (2024). Sequencing Initial Conditions in Physics-Informed Neural Networks. *Journal of Chemistry and Environment.* 3(1). p. 98-108

Authors Bibliography

Saman Hooshyar is a computer scientist with a Master's degree from the Computer Science department of the University of Illinois Chicago, USA.

Arash Elahi is a Ph.D. student at the Chemical Engineering Department of the University of Illinois, Chicago, USA.