IRSTI 28.23.37 https://doi.org/10.26577/jpcsit2023v1i4a4

Y. Kenzhebek[1*], A. Amangeldiyev[2]

[1]National Engineering Academy of the Republic of Kazakhstan, Almaty, Kazakhstan
[2]Astana IT University, Astana, Kazakhstan
*e-mail: kenzhebekyerzhan@gmail.com

PHYSICS-INFORMED NEURAL NETWORKS FOR PREDICTING PRESSURE DISTRIBUTION IN POROUS MEDIA

Abstract. In recent years, the integration of modern information technologies has become pervasive across various industries, and the oil sector is no exception. The utilization of high-performance computing technologies, artificial intelligence algorithms, and advanced methods for data collection, processing, and storage has been instrumental in addressing challenges related to enhancing oil recovery. While deep learning has demonstrated significant advancements in diverse applications, its application to solving partial differential equations has recently gained prominence. A noteworthy strategy entails substituting conventional numerical techniques with neural networks that approximate solutions to partial differential equations. Physics-informed neural networks (PINNs) represent a significant development in this domain by incorporating partial differential equations directly within the loss function of neural network through automatic differentiation. This study presents a numerical algorithm and a PINNs to solve the one-dimensional equation describing the distribution of water and oil pressure within the context of the Buckley-Leverett mathematical model. The obtained results include the numerical solution and predictions derived from the PINN neural network to solve the pressure distribution. The insights gained from the comparative analysis underscore the promising role of PINNs as a robust and competitive tool for addressing intricate problems within the realm of complex fluid dynamics.

Key words: Enhanced Oil Recovery, Physics-Informed Neural Networks (PINNs), Deep Learning, Numerical Method.

1 Introduction

In the last decade and a half, deep learning, represented by deep neural networks, has demonstrated significant effectiveness in diverse applications like computer vision and natural language processing [1]. Although its success in these areas, the widespread adoption of deep learning in scientific computing has been limited. Nevertheless, a contemporary trend is evolving, centering around the utilization of deep learning to address partial differential equations [2]. In this strategy, conventional methods of numerical discretization are substituted with neural networks that provide approximations for solving differential equations.

Achieving an approximate solution to differential equations through deep learning techniques involves a crucial step: restricting the network to minimize the residual of partial differential equations. Various methods have been suggested for this objective. In contrast to traditional grid-oriented techniques like finite difference methods and finite element methods, deep learning presents a mesh-free alternative, capitalizing on automatic differentiation [3]. Some of these approaches may be applicable only to specific categories of issues, such as input domains like images [4] or parabolic partial differential equations [5].

The first glimpses of the prospect of using structured prior information to create data-efficient and physics-aware machine learning have already been demonstrated in recent research [6]. In that work, the authors employed Gaussian process regression to develop functional representations adapted to a given linear operator, accurately deriving solutions. Additionally, they provided uncertainty assessments for diverse model scenarios in the realm of mathematical physics. Subsequent research [7, 8] has suggested expansions to nonlinear issues within the domains of logical inference and identifying systems. In the light of the versatility and mathematical sophistication of Gaussian processes in capturing prior information, dealing with nonlinear challenges imposes two significant limitations.

Physics-Informed Neural Networks are described in works [9-13], where the use of these networks adheres to the laws of physics (for problems described by differential equations). In [9], the application of PINNs to classical fluid mechanics and quantum mechanics problems is discussed.

In [10, 11], the authors introduce a deep learning approach called physics-informed neural networks for quantitative uncertainty assessment in ordinary differential equation systems. In 2020, this method started to be applied for mapping heart activations [12] and evaluating fluid conductivity governed by Darcy's law [13]. The results of these works demonstrate that applying PINNs can yield results comparable to those of physical models.

The early stages of developing Physically Informed Neural Networks primarily concentrated on comprehending and enhancing their training dynamics. Initially, there was a significant challenge related to the disparate convergence rates among various elements of the loss function, a pivotal aspect in neural network training. This discrepancy often resulted in scenarios where the network prioritized learning the physical dynamics at the expense of fitting the data or vice versa. In [14], the authors contribute to this comprehension by conducting an wide-ranging survey of literature on PINNs, elucidating their characterization, advantages, and disadvantages. Various PINN variants, including physics-constrained neural networks (PCNN), variational hp-VPINN, and conservative PINN (CPINN), are discussed, highlighting the diversity within the field. The seminal work in [15] delves deep into this challenge, providing crucial insights into the concepts of training PINNs. It underscores the necessity for a balanced approach that ensures equitable representation of both physical laws and data throughout the learning process. This understanding played a pivotal role in guiding subsequent advancements in the field.

Establishing a robust theoretical foundation, particularly concerning error analysis, is a crucial aspect of the development and utilization of physics-informed neural networks. Linear parabolic differential equations are commonly employed for modeling time-dependent phenomena such as heat transfer and diffusion processes. The error analysis outlined in [16] holds significance as it provides valuable insights into the precision and dependability of PINNs when employed in the context of these equations. This study systematically investigates various error sources, including discretization, approximation, and algorithmic errors, offering benchmarks to assess the efficacy of PINNs. A noteworthy contribution of this analysis lies in addressing the curse of dimensionality, a challenge prevalent in high-dimensional spaces where the space volume escalates exponentially with the number of dimensions. Traditionally, this curse poses computational and analytical challenges in numerical methods. The findings presented in [16] illustrate that PINNs, equipped to handle high-dimensional data while incorporating physical laws, adeptly surmount this challenge. This error analysis not only enriches the comprehension of PINNs' capabilities but also instills confidence in their application to intricate, high-dimensional issues.

2 Physics-Informed Neural Networks

Automatic differentiation. The method of automatic differentiation for calculating derivatives of network outputs relative to network inputs is considered. Considering that neural networks are compositional functions, automatic differentiation repeatedly applies the chain rule to calculate derivatives. Automatic differentiation consists of two steps: a forward propagation to compute the values of all variables and a subsequent backward propagation to calculate the derivatives.

The DeepXDE library was explored to implement a physics-informed neural network. DeepXDE is a deep learning library on top of TensorFlow that supports many features: construction of primitive and complex geometries, support for multiple boundary conditions for partial differential equations, 6 sample filling methods, ease of saving and loading the model during training.

The algorithm for solving differential equations using PINNs consists of four stages:

1. Construction of a neural network $u(x; \theta)$ with parameters θ;

2. Specify two training sets: for the partial differential equation and the boundary/initial conditions that are built into the loss function;

3. Determination of the loss function by aggregating the weighted L^2 norms as residuals from partial differential equations and boundary conditions;

4. Training a network to determine the optimal parameter θ* through the reduction of the loss function.

3 PINNs for modeling the pressure distribution in the Buckley-Leverett model

A numerical algorithm has been developed to solve the equation for distribution of pressure from the Buckley–Leverett mathematical model. To numerically solve the pressure equation, the Jacobi iterative method was used.

$$div(\vec{v}_w) + div(\vec{v}_o) = 0, \qquad (1)$$

where $\vec{v}_w, \vec{v}_o$ – fluid flow speed, which is expressed by the following Darcy's law:

$$\vec{v}_i = -K_0 \frac{f_i(s)}{\mu_i} \nabla P, i = w, o, \qquad (2)$$

$f_i(s), \mu_i$ – relative phase permeabilities and viscosities of the water and oil phases, respectively, K_0 – absolute permeability. Substituting equation (2) to equation (1), we obtain a one-dimensional equation for pressure:

$$\frac{\partial}{\partial x}\left(Mx \frac{\partial P}{\partial x}\right) = 0, \qquad (3)$$

where Mx is denoted as follows:

$$Mx = (-K_0 \frac{f_1(s)}{\mu_1}) + (-K_0 \frac{f_2(s)}{\mu_2}).$$

A neural network PINN was built to solve the one-dimensional pressure equation from the Buckley-Leverett mathematical model. Figure 1 shows the PINN architecture for solving this problem:

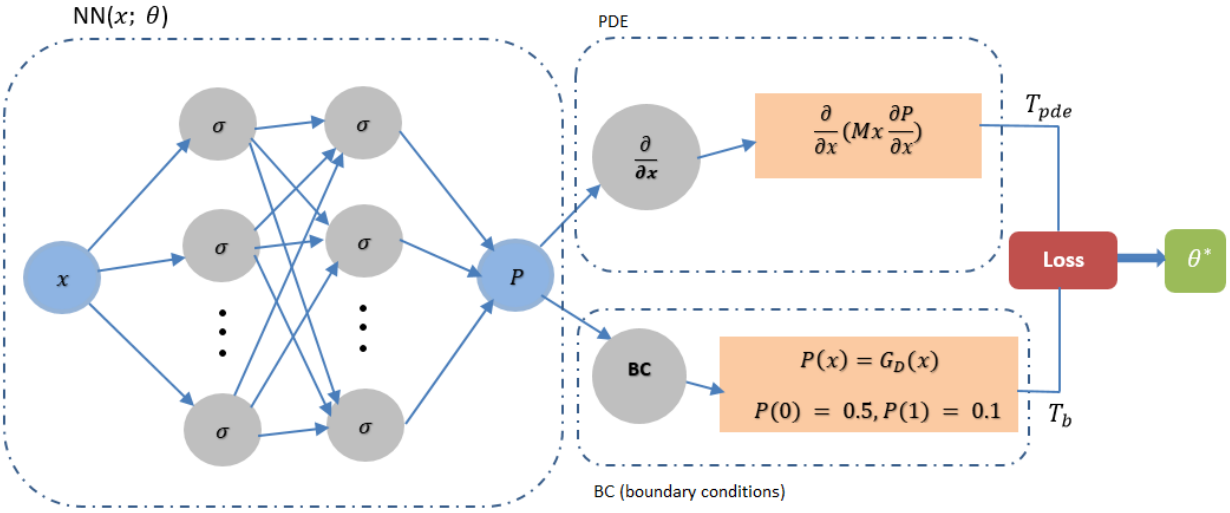

Figure 1 – PINN neural network architecture for pressure equation

In PINN, an initial step involves constructing a neural network, denoted as P(x; θ), to serve as a substitute for the solution p(x). This neural network takes an input x and produces a vector with dimensions identical to *p*. Here, θ = {W, b} represents the collection of all weight matrices and bias vectors within the neural network P. An advantageous feature of PINN, particularly in opting for neural networks to approximate *p*, lies in the ability to compute derivatives of P concerning input x. This is achieved through the chain rule for differentiating compositions of functions using automatic differentiation (AD), conveniently integrated into machine learning packages.

The loss function is considered, defined as a weighted summation of the L^2 norm of residuals from the equation and boundary conditions:

$$L(\theta;T) = w_{pde}L_{pde}(\theta;T_{pde}) + w_b L_b(\theta;T_b).$$

This loss function will be used for model training so that it fits the pressure equation for the oil displacement problem and satisfies the boundary conditions.

Here L_b is the loss function for the boundary conditions. This is used to ensure that the model satisfies the boundary conditions.

$$L_b = \frac{1}{N_b}\sum_{i=1}^{N_b}((\mathrm{p}(x_i) - p)^2|_{x_i},$$

where N_b is the number of points on the border, and x_i are the coordinates of these points.

Whereas L_{pde} is the loss function for the partial differential equation. This is used to allow the model to approximate the equation. For the pressure equation from the Buckley-Leverett model is defined as L^2 norm between the left and right sides of the equation, that is, the discrepancy of the equations. As can be seen from the loss function, labeled data is not used here, that is, the physical limitations of the equation under consideration are considered.

$$L_{pde} = \frac{1}{N_{pde}}\sum_{j=1}^{N_{pde}}\left|\frac{\partial}{\partial x}\left(Mx\frac{\partial P}{\partial x}\right)\right|^2 |_{x_j},$$

where N_{pde} is the number of points at which the partial differential equation is applied.

A fully connected neural network was used, consisting of 4 layers (3 hidden layers) and a width of 32 neurons: [1] + 32*[3] + [1]. The x-space component is taken as the input parameters of the network. Optimizers of the "Adam" type was selected as the network hyperparameter, and the learning rate was 0.001. Testing of 10,000 epochs was carried out for training a neural network, where the number of trained (collocation) points is 100 and two points are used for the Dirichlet boundary condition. PINN construction is implemented using the deepxde deep learning library on top of TensorFlow, which supports many functions for constructing geometries. In Figures 2 and 3 you can see the network training results and prediction:

For testing, 120 points of the numerical solution of the pressure equation and the predicted 120 points of the PINN neural network were used. Figure 3 shows a visualization of the numerical solution and the PINN solution. From Figure 3 it is noticeable that the values are very close, the next Figure 4 shows the absolute error between the two solutions.

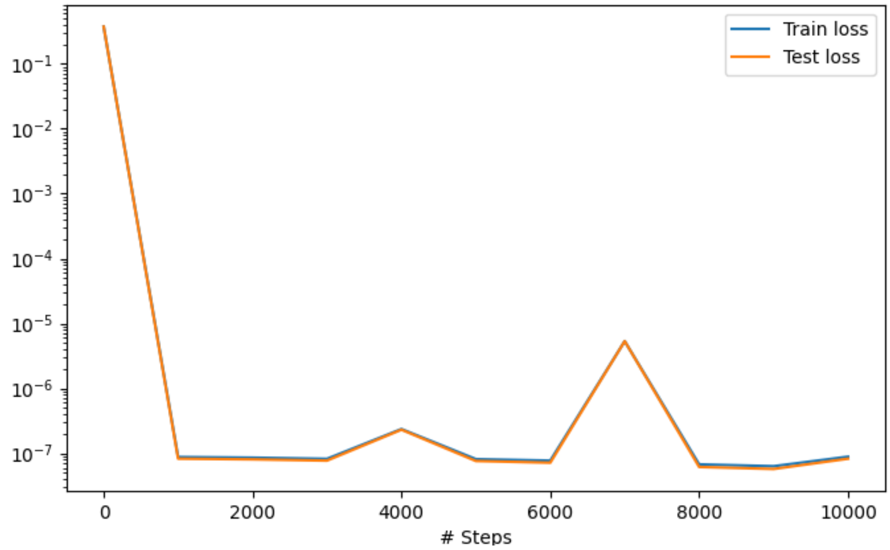

Figure 2 – PINN training loss history

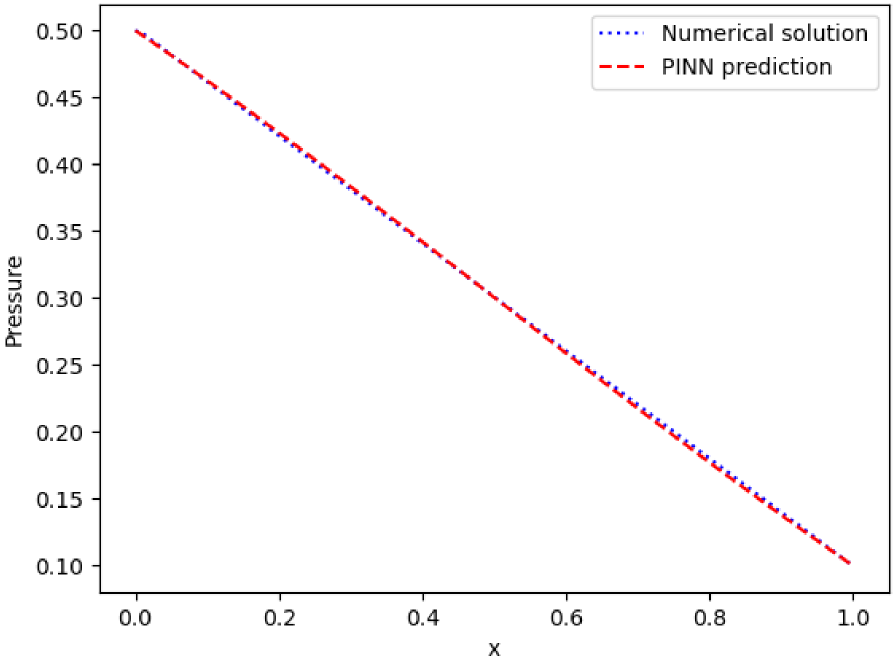

Figure 3 – Comparison of the numerical solution of the pressure equation with the PINN prediction

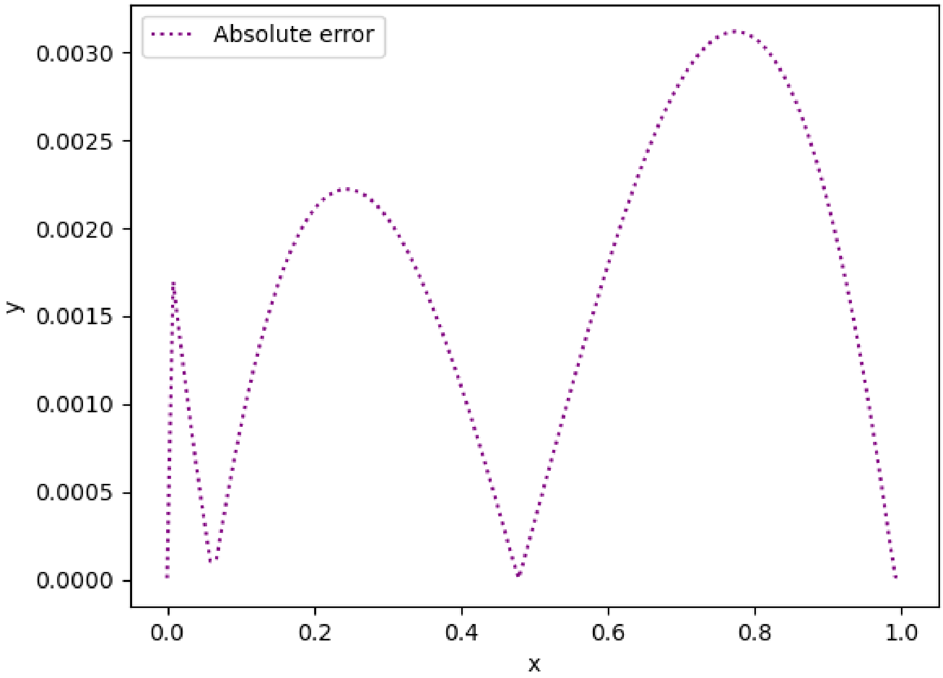

Figure 4 – Absolute error between the numerical solution and the PINN prediction

4 Conclusion

A numerical algorithm and a fully connected neural network PINN have been developed and tested to solve the pressure equation. The results of the numerical solution of the pressure equation are compared with the prediction of a physics-informed neural network. In the comparative analysis, it is revealed that the absolute error between the PINN predictions and the numerical solution is within the range of 0.0005-0.003. This quantifiable measure underscores the precision and effectiveness of PINNs in capturing the complex dynamics of the pressure equation, positioning them as a valuable alternative to traditional linear approximation methods. At a more fundamental level, physics-informed neural networks provide a nonlinear approximation of a function and its derivatives, whereas traditional methods provide a linear approximation.

Acknowledgements

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP14871644).

References

1 LeCun Y., Bengio Y., Hinton G. (2015) Deep learning. Nature, 521, p. 436.

2 Baker N., Alexander F., Bremer T., Hagberg A., et al. (2019) Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Tech. report, US DOE Office of Science, Washington, DC (United States).

3 Poggio T., Mhaskar H., Rosasco L., Miranda B., Liao Q. (2017) Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. International Journal of Automation and Computing, 14, 503–519.

4 Khoo, Yuehaw & Lu, Jianfeng & Ying, Lexing. (2017). Solving parametric PDE problems with artificial neural networks. European Journal of Applied Mathematics, 32, 421-435. 10.1017/S0956792520000182.

5 Beck C., Ee W., Jentzen A. (2019) Machine learning approximation algorithms for highdimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. Journal of Nonlinear Science, 29, 1563–1619. https://doi.org/10.1007/s00332-018-9525-3

6 Raissi M., Perdikaris P., Karniadakis G.E. (2017) Inferring solutions of differential equations using noisy multi-fidelity data. J. Comput. Phys., 335, 736–746.

7 Raissi M., Perdikaris P., Karniadakis G.E. (2017) Numerical Gaussian processes for time-dependent and non-linear partial differential equations. SIAM J. Sci. Comput., 40(1), 172–198.

8 Raissi M., Karniadakis G.E. (2018) Hidden physics models: machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357, 125-141.

9. Raissi M., Perdikaris P., Karniadakis G.E. (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686-707.

10 Zhu Y., Zabaras N., Koutsourelakis P.S., Perdikaris P. (2019) Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics, 394, 56-81.

11 Yang Y., Perdikaris P. (2019) Adversarial uncertainty quantification in physics-informed neural networks. Journal of Computational Physics, 394, 136-152.

12 Sahli F., Yang Y., Perdikaris P., Hurtado D.E., Kuhl E. (2020) Physics-Informed Neural Networks for Cardiac Activation Mapping. Frontiers in Physics, 8. 10.3389/fphy.2020.00042.

13 Tartakovsky A.M., Marrero C.O., Perdikaris P., Tartakovsky G.D., Barajas-Solano D. (2020) Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems. Water Resources Research, 56 (5). https://doi.org/10.1029/2019WR026731

14 S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, “Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next,” Journal of Scientific Computing, vol. 92, no. 3, 2022, doi: 10.1007/s10915-022-01939-z.

15 S. Wang, X. Yu, and P. Perdikaris, “When and why PINNs fail to train: A neural tangent kernel perspective,” Journal of Computational Physics, vol. 449, p. 110768, Jan. 2022, doi: 10.1016/j.jcp.2021.110768.

16 T. De Ryck and S. Mishra, “Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs,” Adv Comput Math, vol. 48, no. 6, p. 79, Nov. 2022, doi: 10.1007/s10444-022-09985-9.