Solving Burger's Equation Using Physics-Informed Neural Networks (PINNs)

Rhydal Esi Eghan N'ganomah Abigail Addita Yaokumah Evans

Kwame Nkrumah University of Science and Technology
Department of Mathematics

ABSTRACT

Polymer flooding is an enhanced oil recovery method that boosts oil extraction by increasing fluid viscosity, but tracking polymer concentration in reservoirs can be challenging due to instabilities in traditional numerical methods. This research applies Physics-Informed Neural Networks (PINNs) to solve the nonlinear Burgers' equation, which models polymer concentration. By integrating physical laws into the training, PINNs improve accuracy and stability compared to conventional methods like Euler, Runge-Kutta, and Finite Difference. The study shows PINNs outperform traditional methods, offering a more reliable and efficient approach for predicting polymer behavior in various reservoir conditions.

BACKGROUND OF STUDY

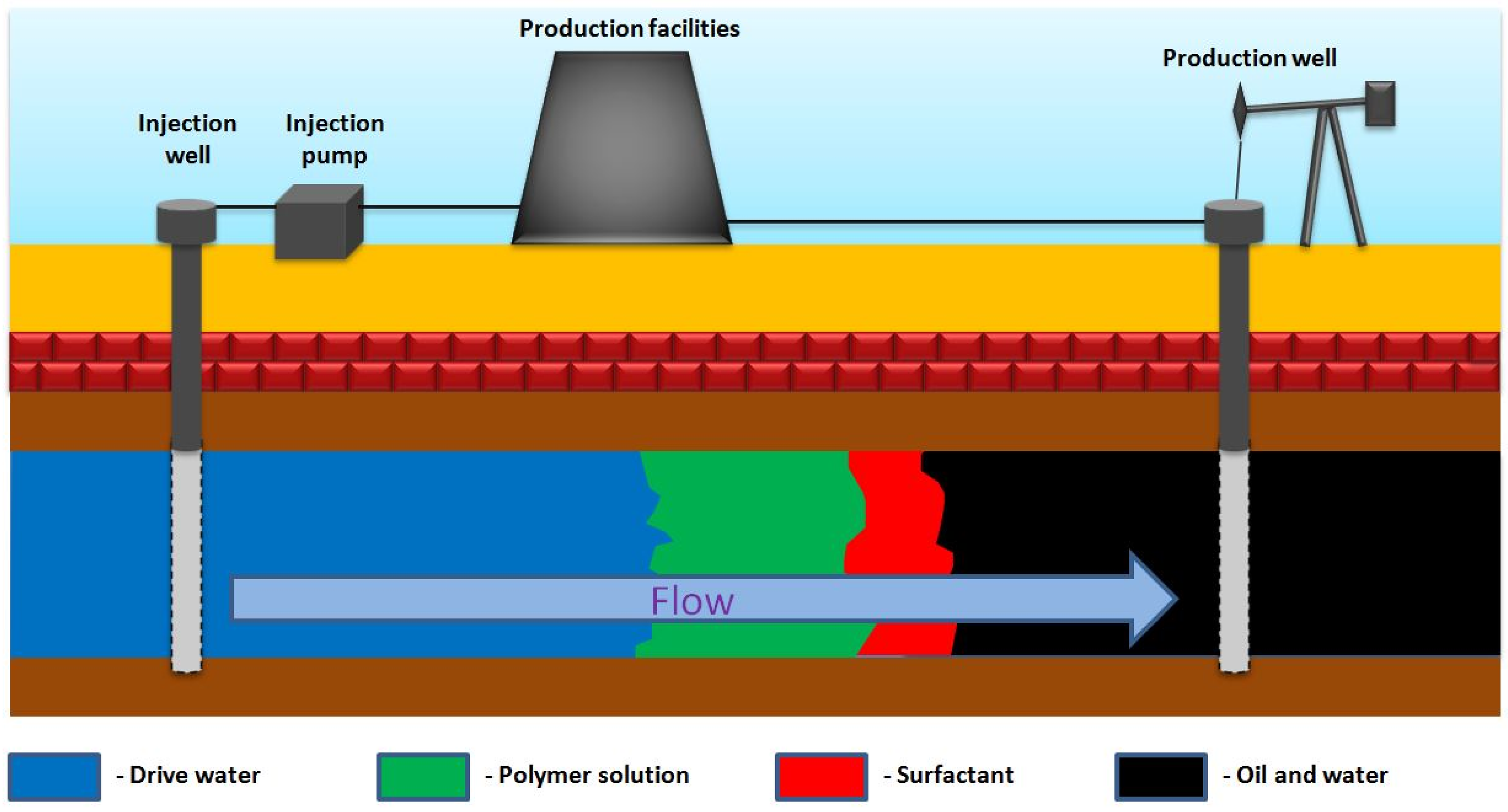


Figure 1. Polymer Flooding in Reservoir Simulation

PROBLEM STATEMENT

Traditional methods(Euler, Runge-Kutta, Finite Difference, etc) for solving nonlinear PDEs often suffer from numerical instabilities due to the violation of the physical laws or constraints governing the equation.

OBJECTIVES

To employ Physics-Informed neural network (PINN)–based methods to address numerical instabilities in solving nonlinear partial differential equation.

METHODOLOGY

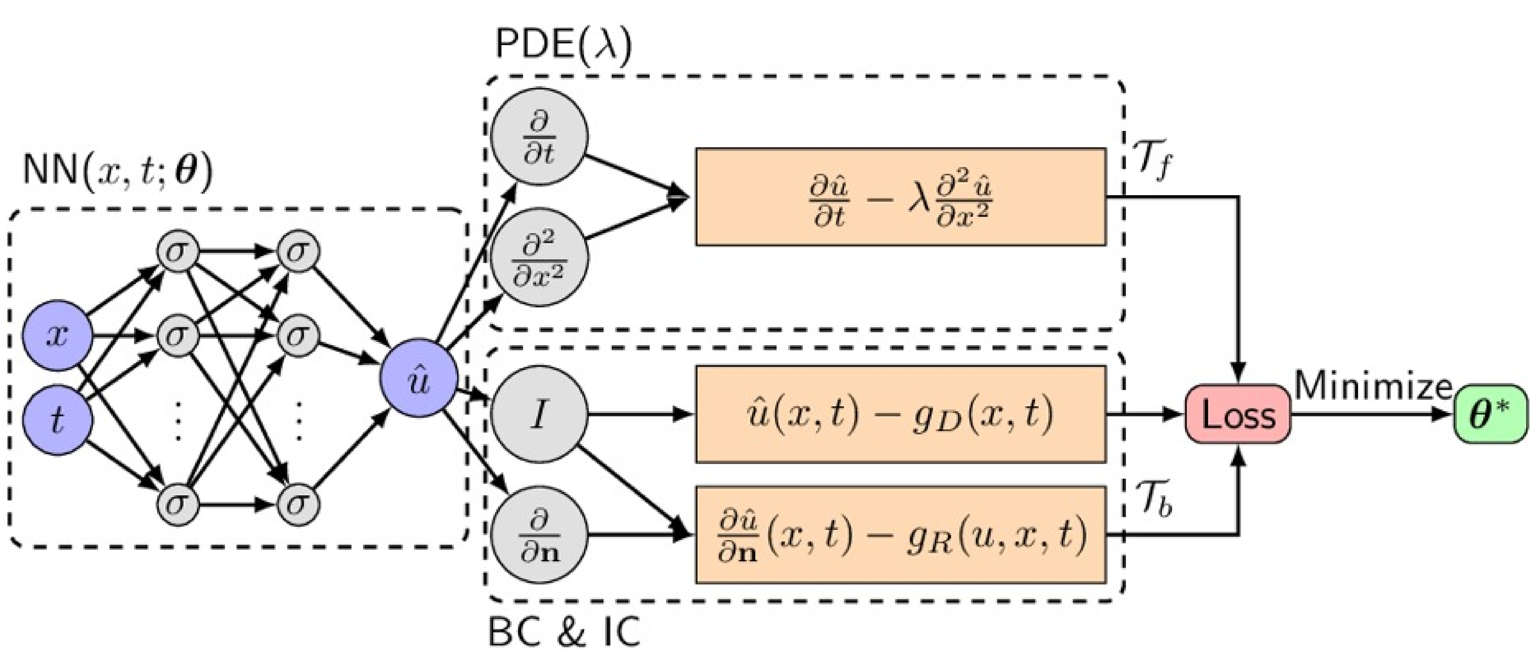


Figure 2. Model Architecture

For the input layer

$$\mathbf{X} = (x, t) \tag{1}$$

For the hidden layer:

$$\mathbf{h}^{(j)} = \sigma(\mathbf{W}^{(j)}\mathbf{h}^{(j-1)} + \mathbf{b}^{(j)}) \tag{2}$$

For the output layer

$$u(x,t) = \mathbf{W}^{(j+1)}\mathbf{h}^{(j)} + \mathbf{b}^{(j+1)} \tag{3}$$

- **Polymer Concentration Dynamics in Oil Reservoirs Using Burgers' Equation:**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \tag{4}$$

- $u(x,t)$: Polymer Concentration (kgm^-3)
- $u = 0.5m^2/hours$ and $v = 0.01m^2/hours$
- Initial Condition: $u(x,0) = e^{\frac{-(x-0.5)^2}{v}} \quad kgm^{-3}$
- Boundary Conditions: $u(0,t) = 0$ and $u(100,t) = 0$

RESULTS ANS ANALYSIS

Spacial Domain(metres)	time(hours)	u(x,t)(kgm^{-3})
0	0	77.880
1.124	1	86.048
2.247	2	92.702
3.370	3	77.000
4.494	4	67.782
⋮	⋮	⋮
95.505	95	0.000
96.629	96	0.000
97.752	97	0.000
98.876	98	0.000
100	100	0.000

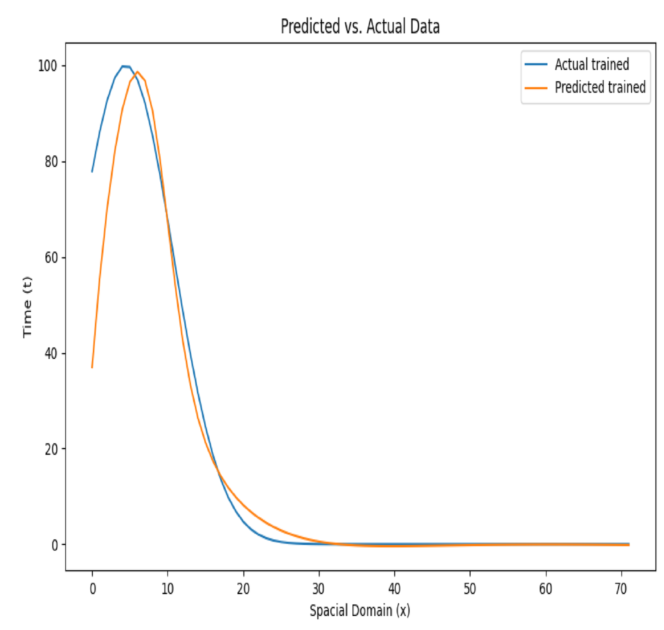
Table 1. 100 Dataset of inputs (x,t) and output $(u(x,t))$ values.

x	t	u	u_p	Abs. Err
0.00	0	77.88	37.00	40.88
1.12	1	86.05	55.48	30.57
2.25	2	92.70	70.24	22.46
3.37	3	77.00	80.15	3.15
⋮	⋮	⋮	⋮	⋮
77.50	77	0.62	0.57	0.05
78.70	78	0.58	0.42	0.16
79.80	79	0.56	0.32	0.24

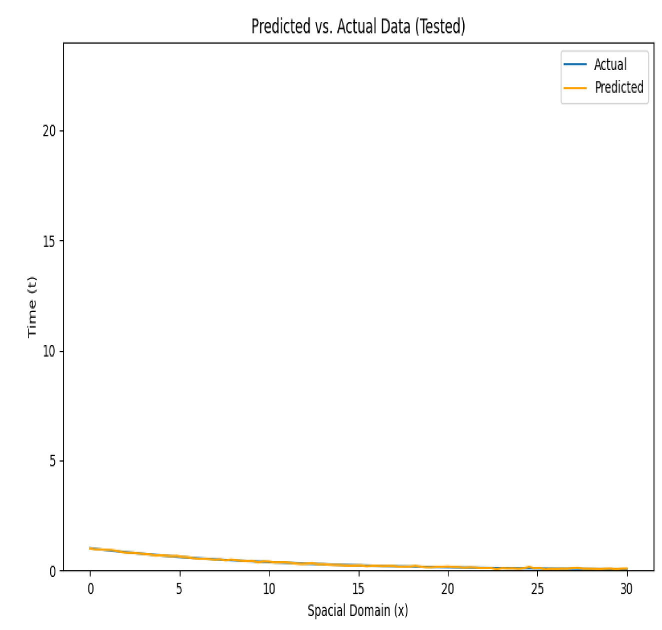
Table 2. Trained Data

x	t	u	u_p	Abs. Err
80.90	80	0.52	0.51	0.01
82.00	82	0.36	0.36	0.00
83.10	83	0.22	0.21	0.01
⋮	⋮	⋮	⋮	⋮
97.80	97	0.00	0.00	0.00
98.90	98	0.00	0.00	0.00
100.00	100	0.00	0.00	0.00

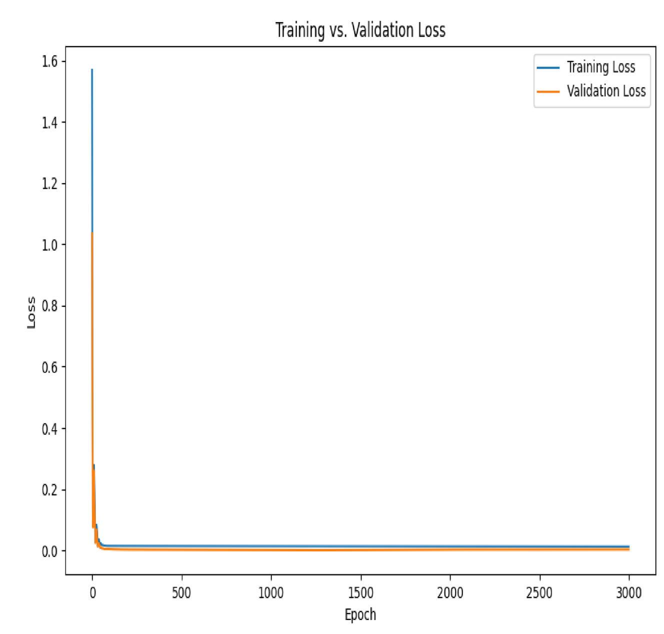
Table 3. Tested Data



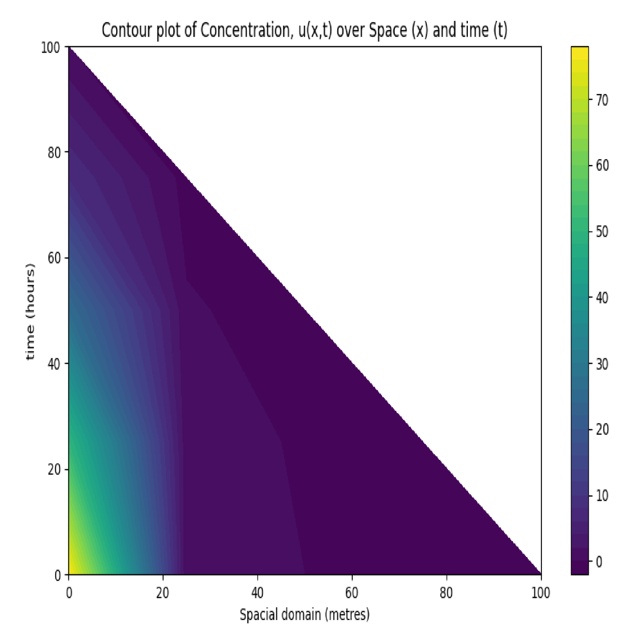
. Predicted vs. Actual Data (Training)



. Predicted vs. Actual Data (Tested)



. Training vs Validation Loss



. Contour plot of Concentration

CONCLUSION

The research demonstrated the successful application of PINNs for solving Burgers' equation, achieving accurate predictions and reducing numerical errors.

REFERENCES

[1] Justin Grimmer, Margaret E Roberts, and Brandon M Stewart. Machine learning for social science: An agnostic approach. *Annual Review of Political Science*, 24:395–419, 2021.

[2] Yanan Guo, Xiaoqun Cao, Bainian Liu, and Mei Gao. Solving partial differential equations using deep learning and physical constraints. *Applied Sciences*, 10(17):5917, 2020.

[3] Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021.

[4] Apostolos F Psaros, Kenji Kawaguchi, and George Em Karniadakis. Meta-learning pinn loss functions.

addiwade100@gmail.com, additaabigail@gmail.com rhydaleghan@gmail.Com