IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3744~3756
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3744-3756

Backpropagation neural networks for solving gas flow equations in porous media

Adrianto[1], Zuher Syihab[2], Sutopo[2], Taufan Marhaendrajana[2]
[1]Doctoral Program of Petroleum Engineering, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Bandung, Indonesia
[2]Department of Petroleum Engineering, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Bandung, Indonesia

Article Info

Article history:

Received Nov 05, 2024
Revised Jul 15, 2025
Accepted Aug 06, 2025

Keywords:

Backpropagation
Gas flow
Neural networks
Nonlinear solver
Reservoir simulation

ABSTRACT

This study proposes a backpropagation neural network (BPNN) as an alternative solver for nonlinear equations in gas flow simulation through porous media. Conventional solvers like the Newton-Raphson (N-R) method are accurate but may become inefficient for large-scale or heterogeneous systems. We develop a feedforward BPNN architecture with adaptive learning rates to solve discretized residual equations from the one-dimensional gas flow model. The methodology includes finite difference discretization and mapping the nonlinear algebraic system into a four-layer neural network. The BPNN solver is validated against the Newton method across various grid sizes and heterogeneous permeability-porosity distributions. Results show that BPNN achieves high accuracy, with maximum absolute errors (MAE) of only 0.241 psi in the homogeneous model and 0.0418 psi in the heterogeneous model. While the BPNN requires more iterations and longer computation time, especially for finer grids, it exhibits the ability to learn pressure patterns and improve efficiency over time. This approach demonstrates that BPNN can serve as a viable nonlinear solver in reservoir simulation, offering flexibility in handling nonlinearities while maintaining accuracy.

Corresponding Author:

Adrianto
Doctoral Program of Petroleum Engineering, Faculty of Mining and Petroleum Engineering
Institut Teknologi Bandung
Ganesha Street No.10, Bandung-40132, Indonesia
Email: adrianto.itb@gmail.com

1. INTRODUCTION

Fossil fuels, particularly those originating from hydrocarbon (oil and gas), remain as the primary global source of energy despite the rising use of renewable resources [1]–[3]. To effectively develop the oil and gas field, we need to understand how fluids move through the porous medium in the subsurface. In addition, the knowledge of fluid flow in porous media also plays an important role in the study of groundwater [4]–[6], geothermal energy [7], [8], and CO_2 sequestration [9]–[11]. Reservoir simulation is one of the fields in petroleum reservoir engineering where numerical computer programs are used to predict fluid flow behavior within oil and gas reservoirs. Since this model contains uncertainties, the physical properties of rocks and fluids, and initial reservoir conditions can be adjusted as long as they are within reasonable engineering limits until a production history match is achieved [12]–[14]. A good history match implies that the reservoir model can predict the future behaviour of hydrocarbon production. Therefore, reservoir simulation can assist the decision-making process in reservoir management and development as it can estimate oil and/or gas recovery under various production scenarios [15].

Due to the nonlinearity of the system of equations in a reservoir simulation, solving it accurately and efficiently is a challenging task. A reservoir model is a simplification of the properties of rocks, fluids, and their interactions as well as the fluid flow mechanism in the reservoir during production. Reservoir geometry can be modeled with various grid shapes, such as regular cartesian, rectilinear, curvilinear, or unstructured. Reservoir fluids could be represented by black-oil or compositional models, depending on the number of phases and the number of components the modeler is concerned with. The production period is divided into several timesteps based on the limit of accuracy to be achieved. We found that the more complex the reservoir numerical model and the higher the expected accuracy, the greater the cost and computational time to solve it. Thus, our study aims to develop a new approach to accurately solve the nonlinear equations found in oil/gas reservoir simulations.

Artificial neural networks (ANN) have been widely used successfully in modeling nonlinearities in many fields [16]–[18]. Among ANNs, backpropagation neural networks (BPNN) are the most popularly used neural network models. BPNN is basically a feedforward network trained using the error gradient calculation, which is called backward pass. The BPNN network can learn and remember extensive input-output mapping relations without requiring prior knowledge of the mathematical equation that defines these relationships. The learning rule employs the steepest descent method, utilizing backpropagation to adjust/modify the weight to reduce the sum of squared errors. This feature makes BPNN popular for predicting complex nonlinear systems.

Besides BPNN, other ANN methods have also been used in fluid flow studies. Iskandar and Kurihara [19] utilize long short-term memory (LSTM) to forecast the produced oil, CO_2 and water during the carbon capture, utility, and storage (CCUS) operations. Zhang *et al*. [20] successfully combined Bayesian Markov chain Monte Carlo (MCMC) and LSTM to assist the history matching process and capture subsurface uncertainty in the 10th SPE comparative model. In the work by Zhang *et al*. [21], recurrent neural network (RNN), LSTM, and gated recurrent unit (GRU) can accurately and effectively predict reservoir outflow in water resources. Li *et al*. [22] proposed a deep neural network (DNN)-based reservoir simulator for hydraulic fracturing and validated by simulating 3D synthetic model and unconventional field. Santos *et al*. [23] proposed a 3D convolutional neural network (CNN) that is able to predict fluid flow in 3D digital rock images. These methods excel in terms of computational time because they are stand-alone models and are not applied to solve nonlinear discrete equations in reservoir simulation. However, the aforementioned neural network approaches necessitate a substantial quantity of historical data to forecast a limited temporal span, and are comparatively less effective in providing an explanation of the fluid flow phenomena occurring within the reservoir.

Raissi *et al*. [24] introduced physics-informed neural networks (PINN) which utilises neural networks that attempt to obtain continuous solutions of partial differential equations (PDE) by incorporating the physics of initial and boundary conditions as the loss function. Ihunde and Olorode [25] demonstrated that PINN can incorporate physical constraints without significantly reducing the accuracy of compositional model, but require up to millions of unique data in their studies. Han *et al*. [26] proposed the physics-informed neural network based on domain decomposition (PINN-DD) which successfully solved the problem of large-scale reservoir simulation with limited data, but at a high computational time and cost. Zhang [27] came up with physics-informed deep convolutional neural network (PIDCNN), which is more efficient than fully connected neural networks, as 2D variables may be treated as images. The primary constraint of PIDCNN is its restriction to structured grids, while the method for representing features set up on unstructured grids in an image-like format remains unclear. A PINN using a capacitance resistance model (CRM) was developed by Maniglio *et al*. [28] to forecast oil production in reservoirs with waterflooding, eliminating the need for a 3D model and ensuring consistency in production data. While ANN and the combination of some methods with other types of neural networks have been developed, conventional reservoir simulation is still required for various types of reservoir management problems such as optimization of infill drilling campaign and application of appropriate enhanced oil recovery (EOR) methods.

This study aims to harness the nonlinearity modeling capabilities of BPNN to be used as a new alternative nonlinear solver in reservoir simulation. This paper is organized as follows: in section 2, the governing equations for the 1D gas flow in porous medium, discretization, and workflows for Newton method and neural network approach is presented. Section 3 contains the results of the neural network approach to solving one-dimensional gas flow in a porous medium, validated by comparing the solutions obtained for the Newton method. The effect of learning rate parameters on the computation time and the number of iterations is investigated on both homogeneous and heterogeneous model. Section 4 summarizes the performance of the proposed neural network-based nonlinear solver in reservoir simulation and provides some suggestions for future research.

2. METHOD

2.1. Model description

Figure 1 illustrates the one-dimensional model the single-phase gas flow in a porous medium. At initial conditions, all cells are assumed to have the identical pressure of 5,000 psi. The Dirichlet condition is assigned at the left by keeping the pressure constant, whereas the Neumann condition of no-flow boundary is assigned at the right. Other model parameters which related to the simulation are summarized in Table 1.

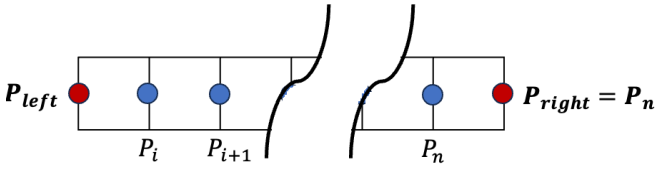


Figure 1. A one-dimensional model of gas flow in a porous medium

Table 1. Model parameters

Parameters	Value
Temperature	200
Initial pressure (psi)	5,000
C1 (%)	100
Permeability (md)	1, 10-20 (heterogeneous)
Porosity (fraction)	0.1, 0.1-0.2 (heterogeneous)
Distance (ft)	1,000
Duration (day)	6.9
Δt	0.0069

In one dimension (linear flow), the equations ruling the single-phase gas flow in a porous medium is as (1):

$$\frac{\partial}{\partial x}\left(\frac{p}{\mu Z}\frac{\partial p}{\partial x}\right) = \frac{1}{\eta}\frac{\partial}{\partial t}\left(\frac{p}{Z}\right) \tag{1}$$

where μ is gas viscosity; Z is gas deviation factor ; p is pressure; x is distance in the x-coordinate; t is time; η is diffusivity coefficient, equal to $0.00633\ k/\phi$, with k and ϕ is the permeability and porosity, respectively. We used the direct correlation of Papay [29] to estimate gas deviation factor, Z, and the correlation of Gonzalez *et al*. [30] to estimate gas viscosity, μ.

To obtain a numerical solution, (1) is discretized using the finite difference method. The left-hand side of (1) relating to space is discretized using centered finite difference, whereas the right-hand side relating to time uses forward finite difference, thus resulting in the residual in (2):

$$r_i = \alpha_i\, p_{i-1}^{t+1} - \beta_i p_i^{t+1} + \gamma_i p_{i+1}^{t+1} \tag{2}$$

with coefficients α, β and γ defined as (3-5):

$$\alpha_i = \frac{\left(\left(\frac{p}{\mu Z}\right)_{i-1}^{t+1} + 4\left(\frac{p}{\mu Z}\right)_i^{t+1} - \left(\frac{p}{\mu Z}\right)_{i+1}^{t+1}\right)}{4\Delta x^2} \tag{3}$$

$$\beta_i = \frac{2\left(\frac{p}{\mu Z}\right)_i^{t+1}}{\Delta x^2} + \frac{1}{\eta \Delta t}\left(\frac{1}{Z_i^{t+1}} - \frac{p_i^t}{p_i^{t+1} Z_i^t}\right) \tag{4}$$

$$\gamma_i = \frac{\left(-\left(\frac{p}{\mu Z}\right)_{i-1}^{t+1} + 4\left(\frac{p}{\mu Z}\right)_i^{t+1} + \left(\frac{p}{\mu Z}\right)_{i+1}^{t+1}\right)}{4\Delta x^2} \tag{5}$$

Figure 2 shows the distribution of rock properties to see the effect of reservoir heterogeneity on solver performance. The permeability distribution is presented in Figure 2(a), while the porosity distribution is shown in Figure 2(b). We used uniform distribution to generate permeability with a range of 10 to 20 milidarcies, and porosity with a range of 0.1 to 0.2.

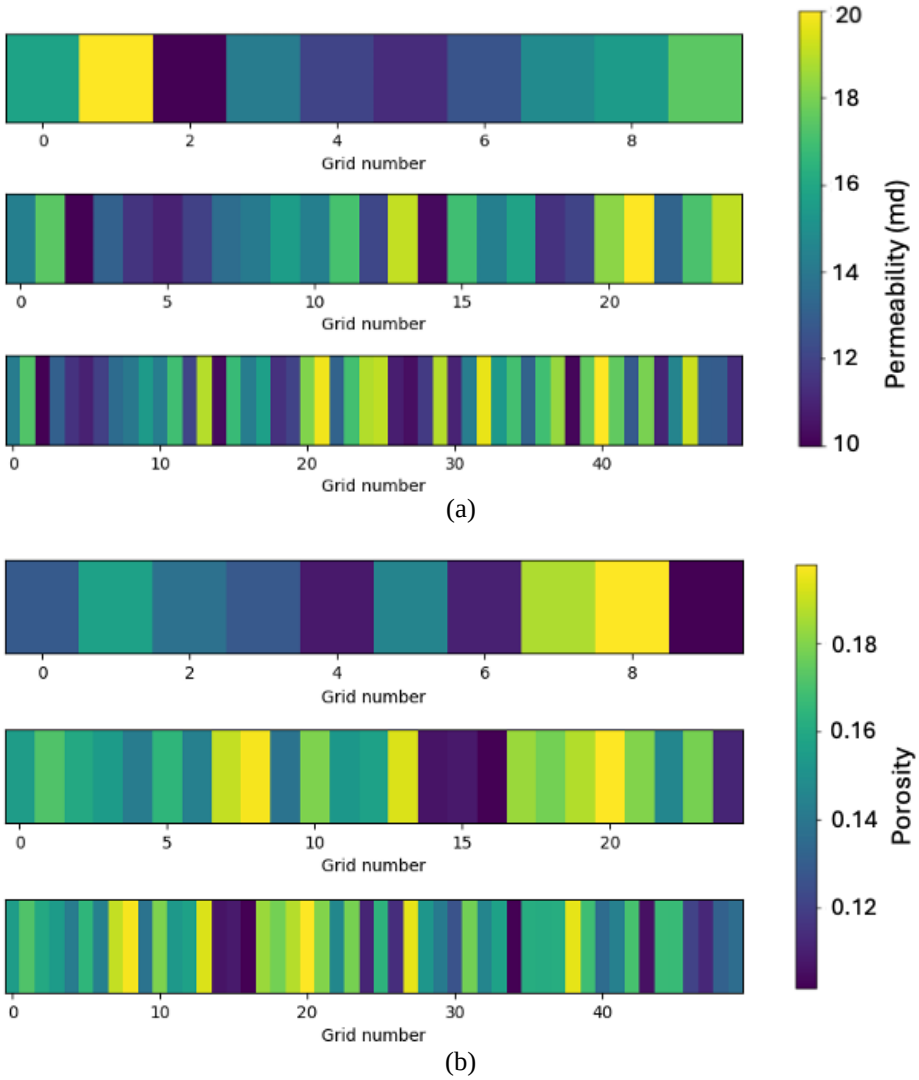


Figure 2. Distribution of (a) permeability in 10 (top), 25 (middle), and 50 (bottom) grids, and (b) porosity in 10 (top), 25 (middle), and 50 (bottom) grids in heterogeneous models

2.2. Newton method

The Newton-Raphson (N-R) method is one of the most frequently used and efficient techniques in solving systems of equations found in mathematical and engineering problems. The N-R method searches for the roots of an equation using the tangent line of a curve iteratively until it approaches the solution. The residual in (2) can be written in matrix form as $Ax = b$, as stated in (6).

$$\begin{bmatrix} -\beta_i & \gamma_i & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_{i+1} & -\beta_{i+1} & \gamma_{i+1} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \alpha_{n-1} & -\beta_{n-1} & \gamma_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_n & -\beta_n + \gamma_n \end{bmatrix} \times \begin{bmatrix} P_i \\ P_{i+1} \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \begin{bmatrix} \alpha_i P_{left} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

To update the solution at each iteration, the solution at the previous iteration and the inverse of the Jacobian matrix are required in (7).

$$P^{k+1} = P^k - J^{-1} r(P^k) \quad (7)$$

The Jacobian matrix is a collection of all the first partial derivatives of the residual in (8).

$$J = \begin{bmatrix} \frac{\partial r_1}{\partial P_1} & \cdots & \frac{\partial r_1}{\partial P_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial P_1} & \cdots & \frac{\partial r_n}{\partial P_n} \end{bmatrix} \quad (8)$$

2.3. Backpropagation neural networks

Werbos [31] first proposed the use of chain rules to systematically calculate gradients in neural networks, which is the basic concept in backpropagation. Later, Rumelhart *et al*. [32] popularized the backpropagation algorithm by using it to train neural networks with multiple layers. Goulianas *et al*. [33] introduced a method to solve a system of nonlinear algebraic equations by using the backpropagation method. For example, for system of equations with n equations and n unknown variables as (9):

$$\begin{aligned} \boldsymbol{F}_1(x) &= \boldsymbol{F}_1(x_1, x_2, \dots, x_n) = \alpha_{11} f_{11}(\boldsymbol{x}) + \alpha_{12} f_{12}(\boldsymbol{x}) + \cdots + \alpha_{1,k_1} f_{1k_1}(\boldsymbol{x}) - b_1 = 0 \\ \boldsymbol{F}_2(x) &= \boldsymbol{F}_2(x_1, x_2, \dots, x_n) = \alpha_{21} f_{21}(\boldsymbol{x}) + \alpha_{22} f_{22}(\boldsymbol{x}) + \cdots + \alpha_{2,k_2} f_{2k_2}(\boldsymbol{x}) - b_2 = 0 \\ &\vdots \\ \boldsymbol{F}_n(x) &= \boldsymbol{F}_n(x_1, x_2, \dots, x_n) = \alpha_{n1} f_{n1}(\boldsymbol{x}) + \alpha_{n2} f_{n2}(\boldsymbol{x}) + \cdots + \alpha_{n,k_n} f_{nk_n}(\boldsymbol{x}) - b_n = 0 \end{aligned} \quad (9)$$

In (9) above is analogous to residual (2). So, for example, the residual for the second grid can be written in (10):

$$r_2(P_1, P_2, P_3) = F_2(x_1, x_2, x_n) \quad (10)$$

Hence, we have the following (11):

$$\alpha_2(P)P_1 + \beta_2(P)P_2 + \gamma_2(P)P_3 = \alpha_{21} f_{21}(x) + \alpha_{22} f_{22}(x) + \alpha_{23} f_{23}(x) \quad (11)$$

where $f_{21}(x) = \alpha_2(P)P_1;\ f_{22}(x) = \beta_2(P)P_2;\ f_{23}(x) = \gamma_2(P)P_3$ and $\alpha_{21} = \alpha_{22} = \alpha_{23} = 1$.

A neural network architecture with four layers can be formed as shown in Figure 3. The weight value connecting the first and second layers is updated iteratively to get the $\boldsymbol{x}$ value that is closest to the solution. Then, the weight that connects layer 2 to layer 3 represents the function $f(\boldsymbol{x})$. While the weight on layer 3 to layer 4 puts the α coefficient in the equation. The residual equation in (2) does not add a fixed constant so that this neural network model does not include bias in the last layer or other layers. The iteration process to update the solution is carried out with the following (12):

$$x_k^{m+1} = x_k^m - \beta(k) \sum_{l=1}^{n} F_l(x) \frac{\partial F_l(x)}{\partial x_k} \quad (12)$$

with $\beta(k)$ is the adaptive learning rate parameter at each iteration in (13):

$$\beta(k) < \frac{2}{\sum_{l=1}^{n} \left(\frac{\partial F_l^m(x)}{\partial x_k} \right)^2} \quad (13)$$

The above system of nonlinear equations is similar to the residual equations found in reservoir simulations. One neuron in layer 1 has a fixed value of 1. In layer 2, this value is the pressure solution that will be solved for each grid so the number of neurons will depend on the number of grids in the model. The neurons in layer 3 represent the three terms in the left-hand segment of the residual equation. While the neuron in layer 4 is the output layer whose value is expected to be close to zero. Table 2 shows the number of neurons in each network layer.

Table 2. Number of neurons in each network layer

Number of grids	Number of neurons			
	Layer 1	Layer 2	Layer 3	Layer 4
10	1	10	30	10
25	1	25	75	25
50	1	50	150	50

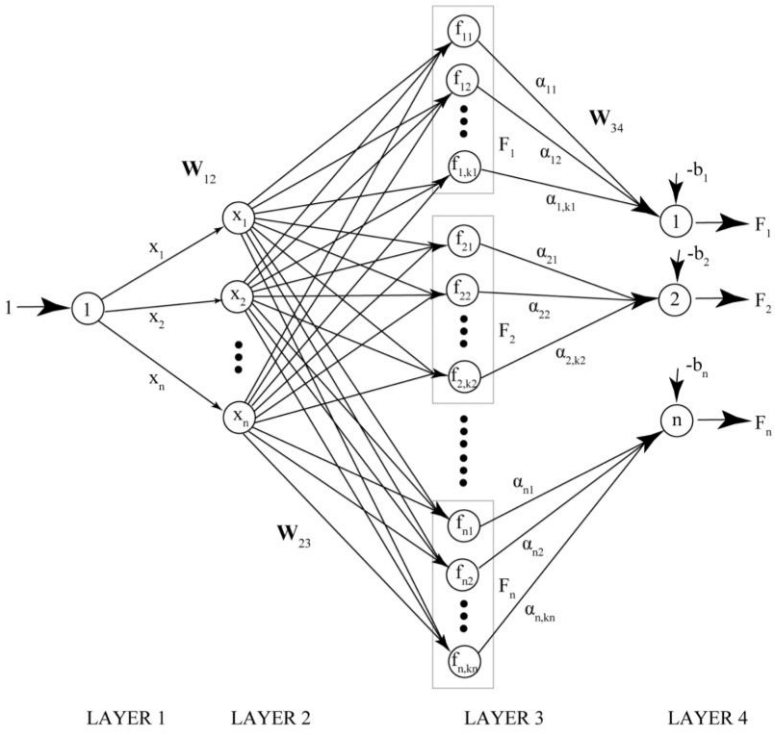


Figure 3. BPNN for solving nonlinear equations [33]

3. RESULTS AND DISCUSSION

3.1. Model validation

We validated the simulation results by comparing the pressure distribution solution from the calculation using the BPNN approach with the solution from the Newton method. Constant pressure at the left boundary condition and no flow at the right boundary condition lead to the pressure distribution profile as seen in Figure 4. Because the permeability of the heterogeneous model (10-20 md) is relatively higher than the homogeneous model (1 md), the pressure drop in the heterogeneous model is much faster. Higher permeability facilitates the more efficient movement of fluids through the rock matrix [34], [35]. This indicates that fluid can be quickly extracted from the reservoir upon the beginning of production, resulting in a more rapid decrease in pressure. At the end of the simulation time of 6.9 days, the pressure at the right end drops to 4987 psi and 4962 psi for the homogeneous and heterogeneous models, respectively. Figure 4 shows that the pressure solutions on each grid are in good agreement between the pressure points from the BPPN approach and from Newton's method in both the homogeneous model in Figure 4(a) and the heterogeneous model in Figure 4(b). The lines denote the solution of the Newton method, while the symbols denote the solution of the BPNN method.

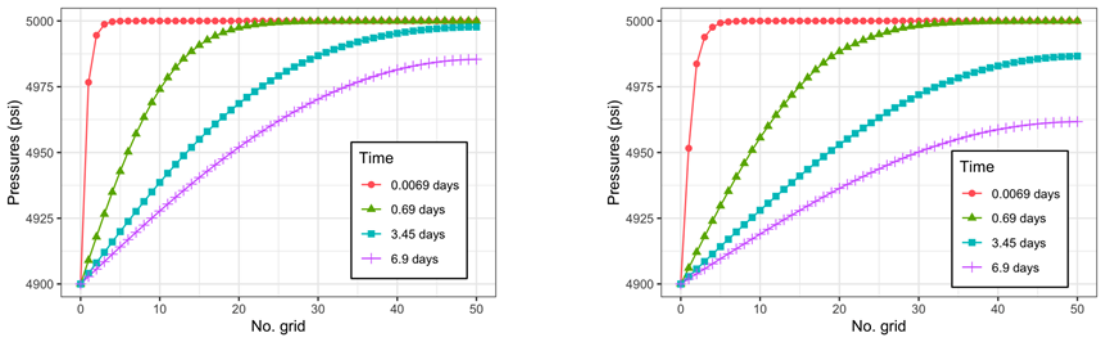
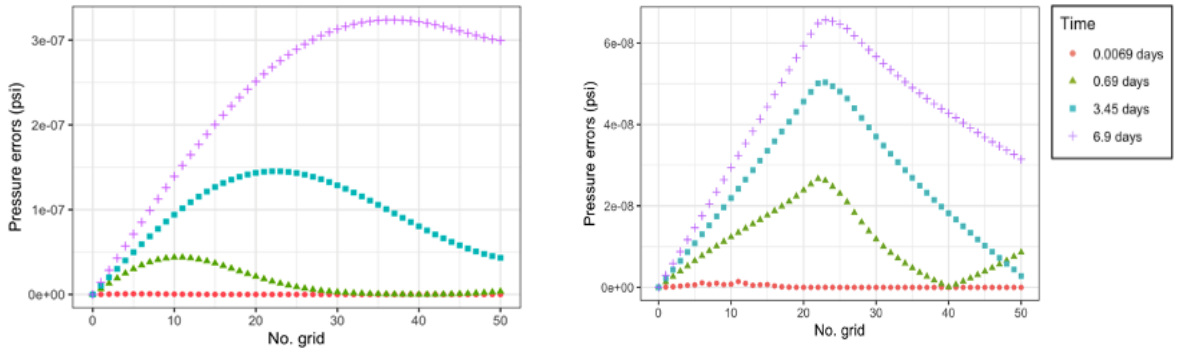


Figure 4. Pressure distribution in (a) homogeneous model and (b) heterogeneous model

Figure 5 shows the pressure error increasing over time in both the homogeneous model in Figure 5(a) and the heterogeneous model in Figure 5(b). This may be due to the truncation error of the finite difference method growing as time progresses [36], [37]. Moreover, finite precision in numerical computations causes round-off errors. These small inaccuracies can accumulate over a long simulation and affect solution accuracy [38], [39]. To overcome this, some researchers usually use high-order finite differences [40], [41] or use smaller time steps during critical parts of the solution [42]. However, it shows that the magnitude of the pressure error is very small with maximum absolute error (MAE) of 2.41×10^{-7} psi and 4.18×10^{-8} psi in the homogeneous and heterogeneous models. This indicates that BPNN produces an accurate solution in predicting the pressure distribution in this model.



(a) (b)

Figure 5. Absolute error of pressures between NN solver and Newton method in (a) homogeneous model and (b) heterogeneous model

3.2. Number of iterations

To achieve an efficient BPNN computation, Goulianas [33] recommends setting the learning rate between 0 and 2. Therefore, we compared the learning rate parameter which is a hyperparameter in neural networks with values from 0.1 to 0.9 with a range of 0.2. Figure 6 shows the performance of the BPNN solver as indicated by the average iterations against the adaptive learning rate parameter on the homogeneous model in Figure 6(a) and the heterogeneous model in Figure 6(b). It shows that the iterations required to obtain the solution on each timestep increase by using a finer grid.

Increasing the resolution of the grid results in a greater number of unknowns while solving issues, thus increases the size and complexity of the system of equations as well as the dimension Jacobian matrix. Nonlinearities in the governing equations may be higher on finer grids. Therefore, addressing these nonlinearities may necessitate more iterations, as the approach must continuously linearize and resolve the system.

Moreover, heterogeneous models also are more likely to require a larger number of iterations. The heterogeneous model with 50 grids require the highest number of iterations among other models. This may because physical variables like permeability and porosity exhibit localized variability, resulting in a more complex set of equations to solve. Significant variations in these characteristics might lead to sharp gradients that require more iterations [43]. In addition, errors arising at locations with major property differences (e.g., boundaries between high- and low-permeability zones) could propagate across the solution domain, leading to additional iterations for corrections [44], [45].

In each simulation run, we set a tolerance of absolute maximum error of 10^{-6} psi, so that the iterations of the Newton and BPNN methods will continue to run until they reach that limit. Figure 7 shows the decrease in residual error as iterations progress for the homogeneous model in Figure 7(a) and the heterogeneous model in Figure 7(b). It shows that BPNN requires more iterations than Newton method in all cases.

An interesting finding from Figure 7 is that the Newton method requires fewer iterations at the beginning of the simulation and then increases. The vice versa is observed for BPNN, where more iterations are required at the beginning of the simulation but decrease as the simulation progresses. This may be due to the ability of BPNN to learn the pattern of pressure drop so that fewer iterations are required at the end of simulation.

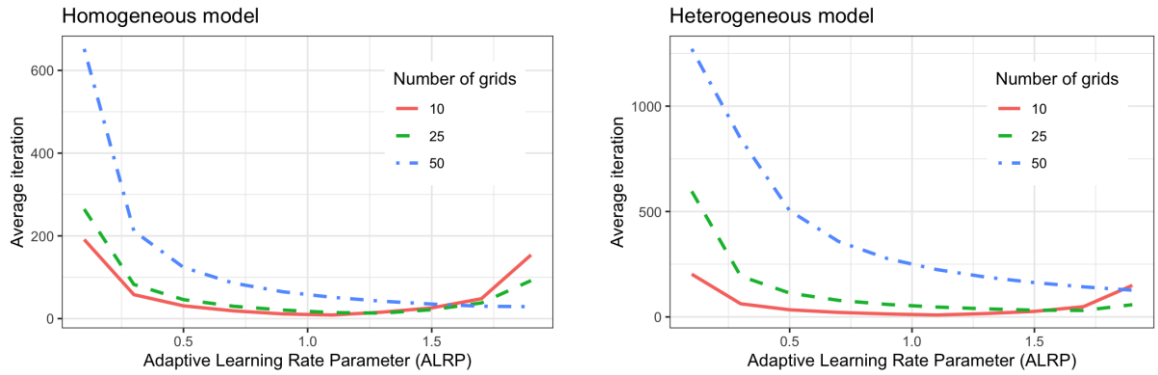


Figure 6. Numbers of iteration at several adaptive learning rate parameters (ALRP) in (a) homogeneous model and (b) heterogeneous model

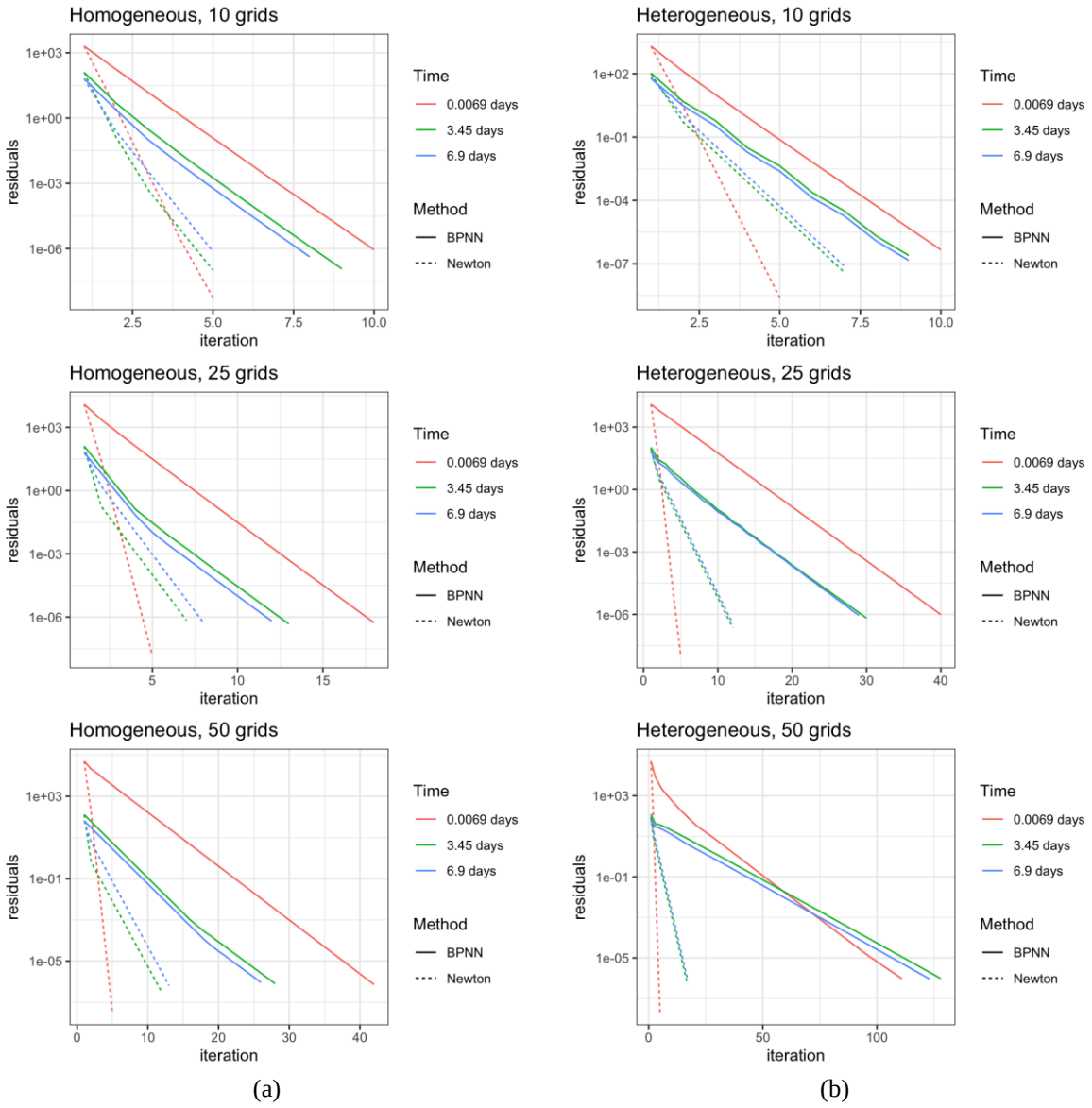


Figure 7. The decrease in residuals at each iteration for all case studies of (a) homogeneous model and (b) heterogeneous model

3.3. Computation time

Computation time indicates the efficiency of a nonlinear solver in reaching a solution. Figure 8 shows the effect of the learning rate parameter on the computation time to reach a converged solution for the homogeneous model in Figure 8(a) and the heterogeneous model in Figure 8(b). It shows that the efficiency of the BPNN method is affected by the learning rate which is set on the network. In general, a learning rate of 0.1 requires the most computation time compared to the others in all cases. The computation time will decrease as the learning rate increases until a certain point where increasing the learning rate will increase the computation time. The learning rate determines the size of the steps taken in the training process of the neural network, especially in minimizing the loss function [46]. A small learning rate can avoid the risk of overshooting but consequently takes more time. While, a large learning rate can speed up the convergence rate but has the risk of overshooting. In the homogeneous model, it was found that the optimum learning rate is 1.1, 1.3, and 1.9 for the number of grids 10, 25 and 50. While in the heterogeneous model, the optimum learning rate is at 1.1, 1.7, and 1.9 for the number of grids 10, 25, and 50.

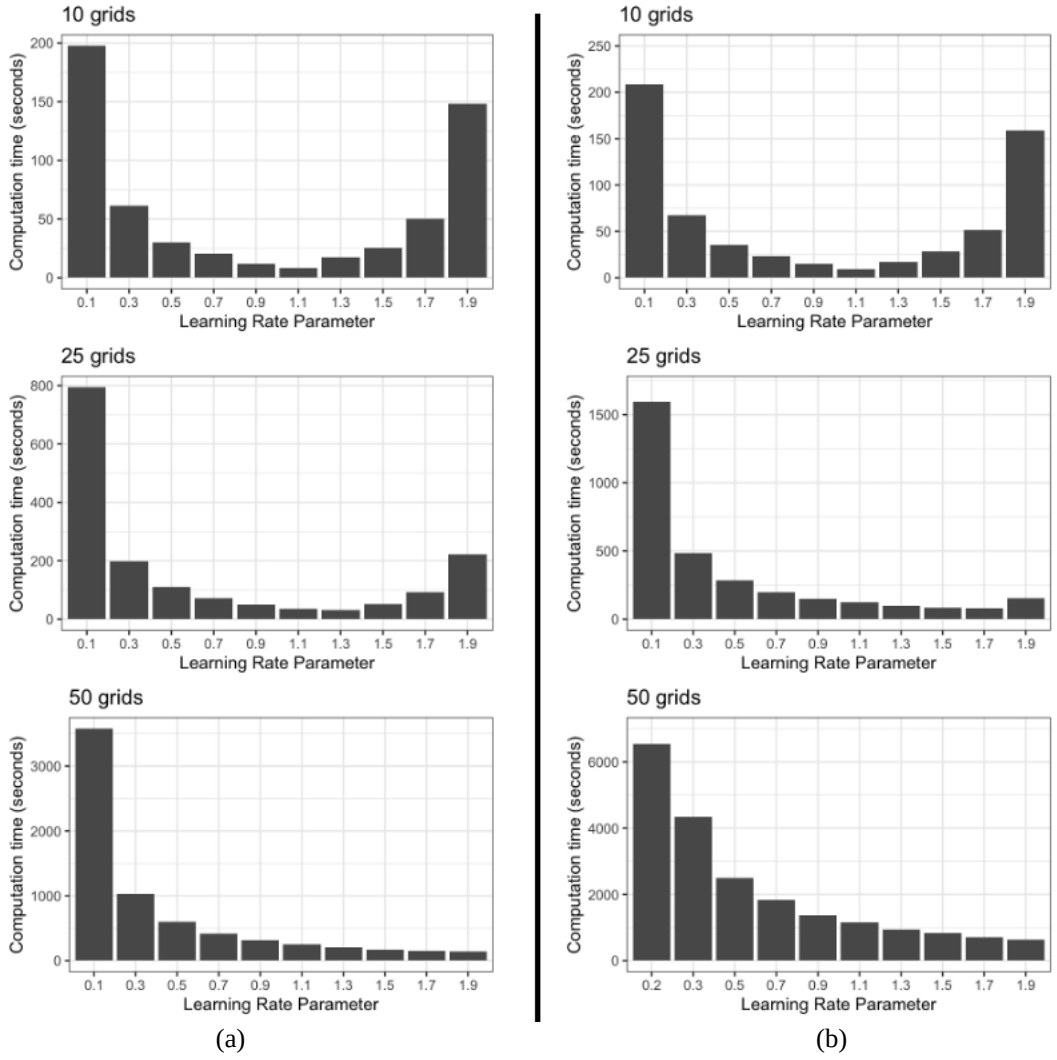


Figure 8. Computation time for (a) homogeneous model and (b) heterogeneous model

Table 3 shows that BPNN is not as efficient as Newton method in solving the single-phase gas flow in this study. On homogeneous models, the computation time of BPNN is about double that of Newton method. Meanwhile, the difference in computation time increases non-linearly in the heterogeneous model. The difference is about 1.5 times in the heterogeneous model with 10 grids, while it is 10 times in the model with 50 grids. The inefficiency of BPNN may be due to the nature of BPNN which involves forward pass and

backward pass so that the calculation is expensive [47]. Moreover, the flow model in this study is relatively simple since it is only one phase and involves only about a dozen grids where Newton method is known for its efficiency on small problem dimensions [48].

We examined computational performance at various Δt values: 0.0138s, 0.0069s, and 0.000345s. Reducing time-step sizes (Δt) increase computational costs for both methods across all models and grid resolutions. For example, in the homogeneous model with 50 grids, the Newton method takes 30.5 seconds for $\Delta t = 0.0138$s, 45.5 seconds for $\Delta t = 0.0069$s, and 69.9 seconds for $\Delta t = 0.000345$s. Increasing the number of grids significantly increases computational time for both methods. This effect is more pronounced in the BPNN approach. For instance, in the homogeneous model with $\Delta t = 0.0138$s, BPNN takes 4.3 seconds for 10 grids but 178.9 seconds for 50 grids. The performance gap between Newton and BPNN is more pronounced in models with higher grid resolutions and smaller time steps. For example, in the heterogeneous model with 50 grids and $\Delta t = 0.0069$s, the Newton method takes 90.4 seconds compared to 610.9 seconds for BPNN. The computational times for the heterogeneous model are typically longer than those for the homogeneous one, indicating the greater complexity involved in solving heterogeneous grids. For example, at $\Delta t = 0.0138s$ and 50 grids, the Newton method requires 30.5 seconds for the homogeneous model and 63.3 seconds for the heterogeneous model. In addition, the difference in computation time also looks nonlinear as shown in Table 4.

Table 3. Computation time for BPNN approach and Newton method

Model	Computational time (seconds)					
	$\Delta t = 0.0138$ s		$\Delta t = 0.0069$ s (base case)		$\Delta t = 0.000345$ s	
	Newton	BPNN	Newton	BPNN	Newton	BPNN
Homogeneous						
10 grids	2.3	4.3	3.8	7.9	7.3	16.7
25 grids	8.9	23.5	14.8	30.7	24.5	66.9
50 grids	30.5	178.9	45.5	172.3	69.9	518
Heterogeneous						
10 grids	3.3	6.1	6.2	8.1	12.6	19.5
25 grids	15.1	76.1	22.5	73.2	36.6	198.4
50 grids	63.3	547.99	90.4	610.9	104.1	496.6

Table 4. Relative computational time of BPNN with Newton method

Model	$\Delta t = 0.0138$ s	$\Delta t = 0.0069$ s (base case)	$\Delta t = 0.000345$ s
Homogeneous			
10 grids	1.8	2	2.2
25 grids	2.6	2	2.7
50 grids	5.8	3.7	7.4
Heterogeneous			
10 grids	1.8	1.3	1.5
25 grids	5	3.2	5.4
50 grids	8.6	6.7	4.7

4. CONCLUSION

This research presents a novel method for solving nonlinear equations in reservoir simulation based on BPNN. The study successfully demonstrates the solver's ability to produce highly accurate solutions, which were verified against the classic Newton method. The pressure solutions achieved a MAE of only 2.41×10^{-7} psi and 4.18×10^{-8} psi for homogeneous models and heterogeneous models, respectively. However, a detailed performance analysis reveals that the BPNN solver, in its current form, is less computationally efficient than the Newton method for the problems studied. Regarding computation time, the BPNN approach was approximately twice as slow for homogeneous models. This performance gap widened non-linearly for more complex, heterogeneous models, where the BPNN solver was between 1.5 to 10 times slower, depending on the grid resolution. The number of grids, rock heterogeneity, and the adaptive learning rate parameter heavily influence the solver's efficiency. The simulation results showed that while the BPNN method required more iterations overall, it exhibited a learning behavior; the iteration needed in each time step decreased over the simulation time, in contrast to the Newton method. This suggests that the network adapts to the solution's pattern over time. The selection of an optimal learning rate was also critical, as it significantly impacted convergence speed and computational cost. Further research should test the BPNN method on more complex fluid models and investigate different network architectures and optimizations to improve its efficiency for larger simulations.

FUNDING INFORMATION

This work is supported and funded by PPMI FTTM Program 2025 from the Faculty of Mining and Petroleum Engineering, ITB.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Adrianto	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
Zuher Syihab	✓	✓		✓	✓				✓	✓		✓		✓
Sutopo		✓		✓		✓				✓		✓	✓	
Taufan Marhaendrajana		✓		✓		✓				✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] R. P. Botão, H. K. de Medeiros Costa, and E. M. dos Santos, "Global Gas and LNG markets: demand, supply dynamics, and implications for the future," *Energies*, vol. 16, no. 13, 2023, doi: 10.3390/en16135223.
[2] Z. Wang, S. Li, Z. Jin, Z. Li, Q. Liu, and K. Zhang, "Oil and gas pathway to net-zero: review and outlook," *Energy Strategy Reviews*, vol. 45, 2023, doi: 10.1016/j.esr.2022.101048.
[3] P. Kosowski, K. Kosowska, and D. Janiga, "Primary energy consumption patterns in selected European countries from 1990 to 2021: a cluster analysis approach," *Energies*, vol. 16, no. 19, 2023, doi: 10.3390/en16196941.
[4] M. Malekzadeh, S. Kardar, and S. Shabanlou, "Simulation of groundwater level using MODFLOW, extreme learning machine and wavelet-extreme learning machine models," *Groundwater for Sustainable Development*, vol. 9, 2019, doi: 10.1016/j.gsd.2019.100279.
[5] M. Bianchi *et al.*, "Simulation of national-scale groundwater dynamics in geologically complex aquifer systems: an example from Great Britain," *Hydrological Sciences Journal*, vol. 69, no. 5, pp. 572–591, 2024, doi: 10.1080/02626667.2024.2320847.
[6] C. Di Salvo, "Groundwater hydrological model simulation," *Water*, vol. 15, no. 4, Feb. 2023, doi: 10.3390/w15040822.
[7] S. Jalilinasrabady, T. Tanaka, R. Itoi, and H. Goto, "Numerical simulation and production prediction assessment of Takigami geothermal reservoir," *Energy*, vol. 236, 2021, doi: 10.1016/j.energy.2021.121503.
[8] I. G. B. Darmawan, Rustadi, and Karyanto, "Hydrothermal fluid flow modeling using numerical simulation to identify potential reservoir zones in the Mount Rajabasa geothermal field," *Kuwait Journal of Science*, vol. 50, no. 3, pp. 376–386, 2023, doi: 10.1016/j.kjs.2023.02.025.
[9] T. Urych, J. Chećko, M. Magdziarczyk, and A. Smoliński, "Numerical simulations of carbon dioxide storage in selected geological structures in North-Western Poland," *Frontiers in Energy Research*, vol. 10, 2022, doi: 10.3389/fenrg.2022.827794.
[10] R. Ramadhan, M. Abdurrahman, R. Bissen, and K. Maneeintr, "Numerical simulation of potential site for CO_2 sequestration in a depleted oil reservoir in northern Thailand," *Energy Reports*, vol. 9, pp. 524–528, 2023, doi: 10.1016/j.egyr.2023.09.096.
[11] C. Tang, W. Zhou, Z. Chen, and J. Wei, "Numerical simulation of CO_2 sequestration in shale gas reservoirs at reservoir scale coupled with enhanced gas recovery," *Energy*, vol. 277, 2023, doi: 10.1016/j.energy.2023.127657.
[12] G. E. Slater and E. J. Durrer, "Adjustment of reservoir simulation models to match field performance," *Society of Petroleum Engineers Journal*, vol. 11, no. 3, pp. 295–305, 1971, doi: 10.2118/2983-pa.
[13] F. Portella, D. Buchaca, J. R. Rodrigues, and J. L. Berral, "TunaOil: a tuning algorithm strategy for reservoir simulation workloads," *Journal of Computational Science*, vol. 63, 2022, doi: 10.1016/j.jocs.2022.101811.
[14] M. Lopes, J. C. von H. Filho, and D. J. Schiozer, "The effect of dynamic data adjustments in production system simulation models on oil production forecasting applied to reservoir simulation models," *Journal of Petroleum Science and Technology*, vol. 11, no. 1, 2021, doi: 10.22078/jpst.2021.4339.1701.
[15] K. H. Coats, "Use and misuse of reservoir simulation models," *Journal of Petroleum Technology*, vol. 21, no. 11, pp. 1391–1398, 1969, doi: 10.2118/2367-pa.
[16] L. Wang, Y. Zeng, and T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series

forecasting," *Expert Systems with Applications*, vol. 42, no. 2, pp. 855–863, 2015, doi: 10.1016/j.eswa.2014.08.018.
[17] H. Li, Q. Wang, and D. Wei, "A novel hybrid model combining BPNN neural network and ensemble empirical mode decomposition," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00446-3.
[18] R. K. Tipu, V. Batra, Suman, K. S. Pandya, and V. R. Panchal, "Predicting compressive strength of concrete with iron waste: a BPNN approach," *Asian Journal of Civil Engineering*, vol. 25, no. 7, pp. 5571–5579, 2024, doi: 10.1007/s42107-024-01130-9.
[19] U. P. Iskandar and M. Kurihara, "Long short-term memory (LSTM) networks for forecasting reservoir performances in carbon capture, utilisation, and storage (CCUS) operations," *Scientific Contributions Oil and Gas*, vol. 45, no. 1, pp. 35–50, 2022, doi: 10.29017/SCOG.45.1.943.
[20] Z. Zhang, X. He, M. AlSinan, H. Kwak, and H. Hoteit, "Robust method for reservoir simulation history matching using Bayesian inversion and long-short-term memory network-based proxy," *SPE Journal*, vol. 28, no. 3, pp. 983–1007, 2023, doi: 10.2118/203976-PA.
[21] D. Zhang, Q. Peng, J. Lin, D. Wang, X. Liu, and J. Zhuang, "Simulating reservoir operation using a recurrent neural network algorithm," *Water*, vol. 11, no. 4, 2019, doi: 10.3390/w11040865.
[22] R. Li, J. Fu, T. Gao, and P. Zhang, "Application of deep neural network (DNN) in reservoir simulation for hydraulic fracturing and production of unconventional wells," *International Petroleum Technology Conference, IPTC 2024*, 2024, doi: 10.2523/IPTC-23759-MS.
[23] J. E. Santos, D. Xu, H. Jo, C. J. Landry, M. Prodanović, and M. J. Pyrcz, "PoreFlow-net: a 3D convolutional neural network to predict fluid flow through porous media," *Advances in Water Resources*, vol. 138, 2020, doi: 10.1016/j.advwatres.2020.103539.
[24] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.
[25] T. A. Ihunde and O. Olorode, "Application of physics informed neural networks to compositional modeling," *Journal of Petroleum Science and Engineering*, vol. 211, 2022, doi: 10.1016/j.petrol.2022.110175.
[26] J. X. Han *et al.*, "Physics-informed neural network-based petroleum reservoir simulation with sparse data using domain decomposition," *Petroleum Science*, vol. 20, no. 6, pp. 3450–3460, 2023, doi: 10.1016/j.petsci.2023.10.019.
[27] Z. Zhang, "A physics-informed deep convolutional neural network for simulating and predicting transient Darcy flows in heterogeneous reservoirs without labeled data," *Journal of Petroleum Science and Engineering*, vol. 211, 2022, doi: 10.1016/j.petrol.2022.110179.
[28] M. Maniglio, G. Fighera, L. Dovera, and C. C. Stabile, "Physics informed neural networks based on a capacitance resistance model for reservoirs under water flooding conditions," *Society of Petroleum Engineers - Abu Dhabi International Petroleum Exhibition and Conference, ADIP 2021*, 2021, doi: 10.2118/207800-MS.
[29] J. Papay, "ATermelestechnologiai parameterek valtozasa a gazlelepk muvelese soran," *OGIL MUSZ, Tud, Kuzl., Budapest*, pp. 267–273, 1968.
[30] M. H. Gonzalez, R. F. Bukacek, and A. L. Lee, "The Viscosity of Methane," *Society of Petroleum Engineers Journal*, vol. 7, no. 01, pp. 75–79, 1967, doi: 10.2118/1483-pa.
[31] P. J. Werbos, "Beyond regression: new tools for prediction and analysis in the behavioral sciences," 1974. [Online]. Available: https://gwern.net/doc/ai/nn/1974-werbos.pdf
[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pp. 399–421, 2013, doi: 10.1016/B978-1-4832-1446-7.50035-2.
[33] K. Goulianas, A. Margaris, I. Refanidis, K. Diamantaras, and T. Papadimitriou, "A back propagation-type neural network architecture for solving the complete n × n nonlinear algebraic system of equations," *Advances in Pure Mathematics*, vol. 06, no. 06, pp. 455–480, 2016, doi: 10.4236/apm.2016.66033.
[34] A. Sheykhinasab *et al.*, "Prediction of permeability of highly heterogeneous hydrocarbon reservoir from conventional petrophysical logs using optimized data-driven algorithms," *Journal of Petroleum Exploration and Production Technology*, vol. 13, no. 2, pp. 661–689, 2023, doi: 10.1007/s13202-022-01593-z.
[35] M. Y. Khan and A. Mandal, "The impact of permeability heterogeneity on water-alternating-gas displacement in highly stratified heterogeneous reservoirs," *Journal of Petroleum Exploration and Production Technology*, vol. 12, no. 3, pp. 871–897, 2022, doi: 10.1007/s13202-021-01347-3.
[36] J. D. Hoffman, "Relationship between the truncation errors of centered finite-difference approximations on uniform and nonuniform meshes," *Journal of Computational Physics*, vol. 46, no. 3, pp. 469–474, 1982, doi: 10.1016/0021-9991(82)90028-6.
[37] M. B. Kalinowska and P. M. Rowiński, "Truncation errors of selected finite difference methods for two-dimensional advection-diffusion equation with mixed derivatives," *Acta Geophysica*, vol. 55, no. 1, pp. 104–118, 2007, doi: 10.2478/s11600-006-0046-3.
[38] J. Liu, M. Möller, and H. M. Schuttelaars, "Balancing truncation and round-off errors in FEM: One-dimensional analysis," *Journal of Computational and Applied Mathematics*, vol. 386, 2021, doi: 10.1016/j.cam.2020.113219.
[39] G. Vilmart, "Reducing round-off errors in rigid body dynamics," *Journal of Computational Physics*, vol. 227, no. 15, pp. 7083–7088, 2008, doi: 10.1016/j.jcp.2008.04.013.
[40] H. S. Price, J. C. Cavendish, and R. S. Varga, "Numerical methods of higher-order accuracy for diffusion- convection equations," *Society of Petroleum Engineers Journal*, vol. 8, no. 03, pp. 293–303, 1968, doi: 10.2118/1877-pa.
[41] M. S. Young, P. C. Casinader, and D. C. Wilson, "The use of higher order finite difference methods in reservoir simulation," *Society of Petroleum Engineers - European Offshore Technology Conference and Exhibition*, pp. 337–348, 1981, doi: 10.2118/180-1980-ms.
[42] E. Ahmed, Klemetsdal, X. Raynaud, O. Møyner, and H. M. Nilsen, "Adaptive timestepping, linearization, and a posteriori error control for multiphase flow of immiscible fluids in porous media with wells," *SPE Journal*, vol. 28, no. 2, pp. 554–574, 2023, doi: 10.2118/203974-PA.
[43] M. Zhai *et al.*, "Three-dimensional numerical simulation and analysis of geomechanical controls of hydraulic fracturing in heterogeneous formations," *Arabian Journal for Science and Engineering*, vol. 47, no. 9, pp. 11533–11552, 2022, doi: 10.1007/s13369-021-06225-y.
[44] S. W. Chapman, B. L. Parker, T. C. Sale, and L. A. Doner, "Testing high resolution numerical models for analysis of contaminant storage and release from low permeability zones," *Journal of Contaminant Hydrology*, vol. 136–137, pp. 106–116, 2012, doi: 10.1016/j.jconhyd.2012.04.006.
[45] M. Yan, C. Lu, J. Yang, Y. Xie, J. Luo, and X. Yu, "Impact of low- or high-permeability inclusion on free convection in a porous medium," *Geofluids*, vol. 2019, 2019, doi: 10.1155/2019/8609682.
[46] O. V. Johnson, C. XinYing, O. E. Johnson, K. W. Khaw, and M. H. Lee, "Learning rate schedules and optimizers, a game changer for deep neural networks," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 210, pp. 327–340, 2024, doi: 10.1007/978-3-031-59711-4_28.

[47] R. Hecht-Nielsen, "Theory of the backpropagation neural network," *Neural Networks for Perception, Computation, Learning, and Architectures*, pp. 65–93, 1992, doi: 10.1109/ijcnn.1989.118638.
[48] B. T. Polyak, "Newton's method and its use in optimization," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1086–1096, 2007, doi: 10.1016/j.ejor.2005.06.076.

BIOGRAPHIES OF AUTHORS

Adrianto is a student of Doctoral Program of Petroleum Engineering in Institut Teknologi Bandung, Indonesia. He received his Master's degree in Geothermal Engineering in 2010 and Bachelor's degree in Petroleum Engineering in 2009 from Institut Teknologi Bandung, Indonesia. His research interests are primarily in the fields of reservoir simulation and artificial intelligence for petroleum and geothermal reservoir engineering. He can be contacted at email: adrianto.itb@gmail.com.

Zuher Syihab is an Assistant Professor at Petroleum Engineering Department in Institut Teknologi Bandung, Indonesia. He received his Ph.D. in Petroleum Engineering from Texas A&M University, USA, in 2009. He obtained Bachelor's degree in Petroleum Engineering from Institut Teknologi Bandung, Indonesia in 1997. His research interests are reservoir modeling and simulation, optimization, artificial intelligence, EOR, and deep learning. He can be contacted at email: zuher.syihab@itb.ac.id.

Sutopo is an Assistant Professor at Geothermal Master Program in Institut Teknologi Bandung, Indonesia. He received his M.Eng. and Dr.Eng. in Petreoleum Engineering from Waseda University, Japan in 1999 and 2003, respectively. He obtained Bachelor's degree in Petroleum Engineering from Institut Teknologi Bandung, Indonesia, in 1991. His research interests are mathematical modelling, upscaling of subsurface flow phenomena, algorithm development, numerical method of partial differential equations. He can be contacted at email: sutopo@itb.ac.id.

Taufan Marhaendrajana is a Professor at Petroleum Engineering Department in Institut Teknologi Bandung, Indonesia. He received his M.S. and Ph.D. degrees in Petroleum Engineering from Texas A&M University, USA, in 1995 and 2000, respectively. His research interests are reservoir simulation, enhanced oil recovery, and reservoir characterization. He can be contacted at email: tmarhaendrajana@itb.ac.id.