



A Hybrid Probabilistic-Backpropagation Neural Network Solver for Nonlinear Systems in Reservoir Simulation

Adrianto¹, Zuher Syihab², Sutopo², and Taufan Marhaendrajana²

¹Doctoral Program of Petroleum Engineering, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung
Ganesha Street No. 10 Bandung, Indonesia.

²Petroleum Engineering Department, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung
Ganesha Street No. 10 Bandung, Indonesia.

Corresponding author: adrianto.itb@gmail.com.

Manuscript received: June 12th, 2025; Revised: July 29th, 2025

Approved: August 15th, 2025; Available online: October 21th, 2025; Published: October 21th, 2025.

ABSTRACT - Reservoir simulation requires solving large, sparse systems of nonlinear equations, where iterative Krylov subspace solvers such as the conjugate gradient (CG), stabilized biconjugate gradient (BiCG-STAB), and generalized minimal residual (GMRES) are widely applied. However, these methods often have limitations in terms of their stability and accuracy in nonlinear systems. This paper introduces a hybrid probabilistic backpropagation neural network (Prob-BPNN) solver that integrates neural-network-based initialization with probabilistic inference to improve robustness. The solver was benchmarked against CG, BiCG-STAB, and GMRES using two synthetic reservoir models with the GMRES solution at a tolerance of 10^{-10} , serving as the reference solution. The results show that Prob-BPNN consistently achieved production profiles closely matching the reference solution, with errors of $MAE \leq 0.066$, $RMSE \leq 0.071$, $MAPE \leq 2.04\%$, and $R^2 \geq 0.945$. In contrast, CG and BiCG-STAB produced unstable and nonphysical results, with errors exceeding 292% and negative R^2 values. In terms of computational performance, Prob-BPNN required 9.96 s in Case 1 and 45.90 s in Case 2, compared to 2.85 s and 1.53 s for GMRES, respectively. Although more computationally expensive, Prob-BPNN delivered convergence on the same residual order of magnitude (below 10^{-3}) as GMRES while avoiding the severe instabilities observed in CG and BiCG-STAB. These findings indicate that the Prob-BPNN is preferable in applications where solver robustness and accuracy are critical, even at the expense of a higher execution time. Future research should focus on reducing computational overhead through parallelization and hybridization strategies to enhance the scalability of large-scale reservoir models.

Keywords: nonlinear solver, probabilistic, neural networks, backpropagation, reservoir simulation.

© SCOG - 2025

How to cite this article:

Adrianto, Zuher Syihab, Sutopo, and Taufan Marhaendrajana, 2025, A Hybrid Probabilistic-Backpropagation Neural Network Solver for Nonlinear Systems in Reservoir Simulation, Scientific Contributions Oil and Gas, 48 (3) pp. 185-201. DOI [org/10.29017/scog.v48i3.1751](https://doi.org/10.29017/scog.v48i3.1751).

INTRODUCTION

Reservoir simulation is a method used in reservoir engineering to quantitatively predict the dynamic behavior and transport of multiphase fluids within porous media over time. It can provide reliable forecasts of reservoir performance, as the primary function of a simulator is to predict reservoir behavior under a variety of operating scenarios (Habib & Joslin 2020; Kristanto et al., 2025; Mithani et al., 2022; Sugihardjo, 2022; Swadesi et al., 2025; Yan et al., 2025). The workflow for obtaining solutions in the reservoir simulator is shown in Figure 1. The main challenge in reservoir simulations is solving a large, sparse system of nonlinear algebraic equations that result from discretizing the governing partial differential equations (PDEs) for fluid flow (Alpak et al., 2023; Chen et al., 2022; Jammoul et al., 2023). Therefore, research on nonlinear solvers for reservoir simulations is being conducted to develop robust and efficient methods.

Review of existing solvers for reservoir simulation

Nonlinear solvers are generally categorized into two main types: direct and iterative. Direct solvers aim to obtain an exact solution in a finite number of operations and are typically robust and independent of initial guesses (Duran & Tuncel 2014; Yang et al., 2025). However, they are computationally expensive and have high memory demands owing to matrix factorization and fill-in effects, particularly when applied to large sparse systems. By contrast, iterative solvers generate a sequence of approximations that gradually converge to the desired solution and generally require less memory than direct solvers. Consequently, the solution of equations in reservoir simulators is typically carried out using iterative solvers such as the conjugate gradient (CG), stabilized biconjugate gradient (BiCG-STAB), and generalized minimal residual (GMRES).

The CG method (Hestenes & Stiefel 1952) is typically applied to the pressure equation in two-phase (oil-water or gas-water) models solved sequentially (e.g., Implicit Pressure, Explicit Saturation). The CG only requires the storage of a few vectors, which is a significant advantage for large problems in which memory is a constraint. In addition, CG often converges quickly in well-conditioned problems. The main drawback of the CG is its strict requirements for symmetric and positive-definite matrices. This is often not the case in fully implicit models, which result in highly nonsymmetric

Jacobian matrices owing to the strong coupling between variables such as pressure, saturation, and composition (Gharieb et al., 2024; Jiang & Pan 2022; Tokuda & Hashimoto 2023).

Unlike CG, BiCG-STAB (van der Vorst 1992) is designed to handle nonsymmetric matrices that arise from fully implicit formulations. BiCG-STAB also requires the storage of a limited number of vectors, making it memory-efficient for large problems. The “stabilized” part of its name refers to its ability to smooth out the erratic convergence behavior often seen in other biconjugate gradient-based methods. However, the performance of BiCG-STAB is highly dependent on the quality of the preconditioner (Benzi 2002; Benzi & Tùma 1999).

GMRES (Saad & Schultz 1986) determines an approximate solution by minimizing the residual norm over a subspace. This minimizes the residual at each step, guaranteeing a monotonic decrease in the residual norm. The GMRES is particularly effective in solving linear systems from fully implicit formulations because it results in highly nonsymmetric Jacobian matrices.

However, GMRES has issues related to memory consumption and computational load. The high memory consumption is because this algorithm requires storage of the basis of the Krylov subspace, which grows at each iteration (Aliaga et al., 2023; Zhao et al., 2022). The computational cost is high because the orthogonalization process at each step of the Arnoldi iteration is computationally expensive, which adds to the overall cost of the solver (He et al., 2016).

Review of advances research in nonlinear solvers

Bakhvalov (1966) introduced multi-grid methods that use a hierarchy of grids to solve a linear system. This method works by smoothing the high-frequency error components on a fine grid and then solving the remaining low-frequency error components on a coarser grid, where the problem is smaller and cheaper to solve. This process is repeated across multiple coarsening levels. The main advantage of this method is its numerical stability because its computational work scales linearly with the number of unknowns. However, the implementation of the multigrid method is challenging because it requires significant effort to tune an effective coarsening strategy and intergrid transfer operators for complex geology (Stüben et al., 2007). While highly effective for the elliptic pressure equation, it is less suited

for the hyperbolic-dominated transport equations for saturation and concentration (Hørsholt et al., 2019). Because Newton-based methods require the assembly of both the Jacobian and residual for a fully coupled system of equations, an operator-based linearization (OBL) approach was introduced. In this algorithm, the discretized mass and energy conservation equations are reformulated into an operator form that clearly distinguishes between the spatially dependent components and the state-dependent properties of the governing equations (Asif et al., 2025; Khait & Voskov 2017). However, the OBL exhibits a decline in performance in cases with very high nonlinearity, such as in combinations

of cAmerican petroleum institute llarity, gravity, and multiphase flow (Li & Abushaikh 2022).

In addition, several studies have focused on solvers that can be used in parallel frameworks. Bhogeswara and Killough (1994) introduced a hybrid numerical approach combining domain decomposition and multigrid techniques to efficiently solve large, sparse linear systems in reservoir simulations on both sequential and parallel computers. Manea et al. (2016) presented a parallel algebraic multiscale (AMS) solver for the pressure equation in heterogeneous reservoir models, formulated as a two-level domain decomposition algorithm with a localization assumption, and

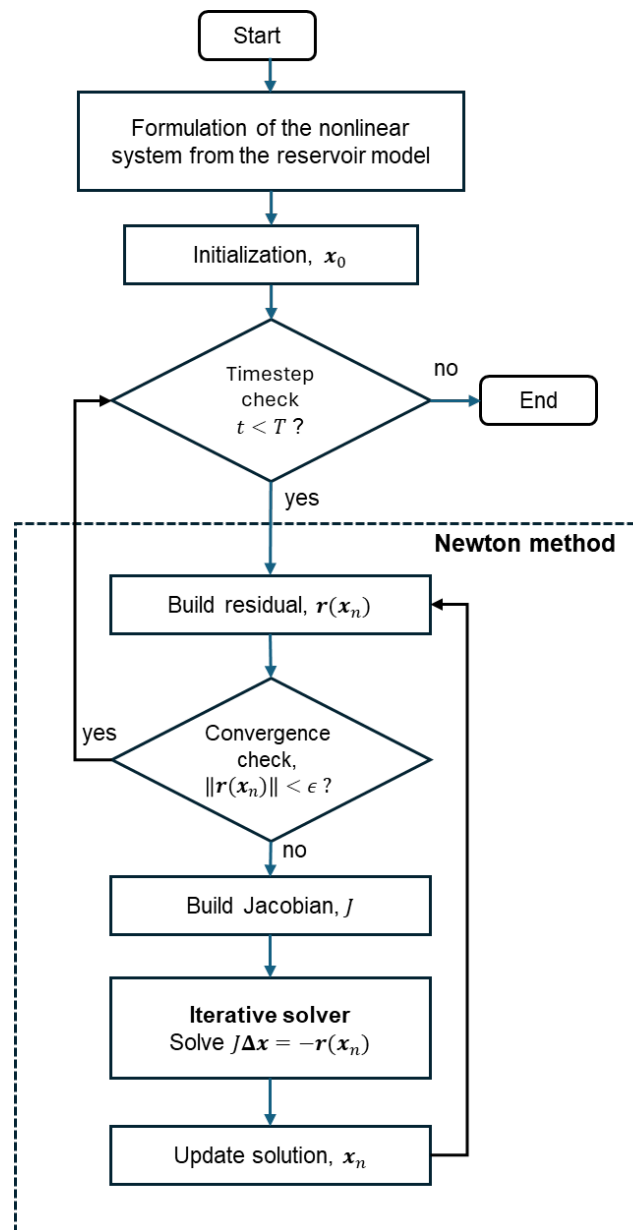


Figure 1. Typical workflow of a reservoir simulator for solving nonlinear equations.

implemented a scalable AMS on multicore and many-core architectures. Kang et al. (2018) proposed a parallel method using OpenACC to accelerate serial code and ease GPU porting, combined with GPU-aided domain decomposition to enhance reservoir simulation efficiency. In their studies, the GPU-aided approach outperformed the CPU-based version by up to two times while reducing code modification to approximately 22% with OpenACC. Furthermore, domain decomposition boosts the execution efficiency by up to 1.7 times. Gasparini et al. (2021) utilized a hybrid OpenMP/MPI programming model with a two-level hierarchical data structure that was efficiently mapped onto OpenMP threads and MPI processes.

Review of neural networks for reservoir simulation

Artificial neural networks (ANN) are widely used in proxy modeling for reservoir simulations, but their roles differ from those of nonlinear solvers. A proxy model serves as a surrogate or approximation for a full-physics reservoir simulator. Instead of directly solving a large system of nonlinear partial differential equations that describe multiphase flow in porous media, the ANN was trained on input–output data generated from a set of precomputed simulations. Once trained, an ANN can quickly predict reservoir responses, such as production rates, pressure, or saturation profiles, under new operating conditions (Zubarev 2009). Kim et al. (2017) integrated a deep neural network (DNN) with a stacked autoencoder (SAE) for multi-objective history matching modeling applications. Isaiah et al. (2013) combined an ANN with the Kriging geostatistical algorithm to predict porosity and permeability, and then applied it to address the challenges of reservoir simulation in mature fields with sparse and outdated data.

Mamo & Dennis (2020) used a nonlinear autoregressive network with an external input (NARX) structure as a prediction model and found that the Bayesian regularization algorithm was the most suitable for training the model. This is because Bayesian regularization can generalize datasets with noise and prevent overfitting; however, it requires a longer computation time. (Alakeely & Horne 2022) examined the use of generative deep learning, specifically a variational autoencoder (VAE), as a proxy for simulating oil and water production profiles. The VAE was trained to map temporal and spatial inputs to a series of production rate data over time, and the results showed that the

VAE significantly outperformed machine learning architectures in predicting production flow rates at unseen locations.

Despite rAmerican petroleum institute d progress, the proxy model approach faces several challenges. Proxy models are considered black boxes because they do not incorporate the physical meaning of a dataset, resulting in predictions that may be physically inconsistent (Karniadakis et al., 2021). Second, the proxy models may not be sufficiently robust to make long-term predictions. This is because the proxy models are only applied to the training dataset interval (Almajid & Abu-Al-Saud 2022).

Another neural network method that has recently been used in reservoir modeling is the physics-informed neural network (PINN), which was first introduced by (Raissi et al., 2019). The PINN algorithm integrates the data, boundary conditions, and initial conditions expressed in the form of partial differential equations to construct a function loss equation. (Gasmi & Tchelepi 2021) showed that the PINN could be used to solve fluid flow equations in synthetic reservoir models with an accuracy comparable to that of classical numerical methods. One of its advantages is that PINN can infer unknown parameters, such as permeability and porosity, with limited pressure data. Furthermore, (Gasmi & Tchelepi 2022) introduced the concept of parameterized PINN (P-PINN), which is a PINN architecture that uses random parameters, such as stochastic permeability distributions, as inputs. (Zhang et al., 2024) applied a PINN to the Buckley-Leverett problem and demonstrated that it can capture the evolution of the saturation front, even under heterogeneous conditions with sharp gradients.

The challenge in the application of PINN in reservoir models is their scalability. Field-scale reservoir models render PINN training computationally cumbersome. The significant differences between the dynamics around the well and areas far from the well could disrupt the training stability and thus slow convergence (Han et al., 2023). Moreover, the PINN generates computational complexity because training the PINN on models with millions of grid cells requires the evaluation of a massive number of residuals, which is still very difficult to accommodate even with GPU acceleration (Gasmi & Tchelepi 2022).

This study introduces a novel nonlinear solver that combines backpropagation neural networks (BPNN) with a probabilistic approach rather than

physics-informed neural networks (PINNs). The proposed method is more readily integrated into existing reservoir simulators, whereas PINNs, despite their accuracy, typically involve substantially higher computational demands, even when applied to relatively simple synthetic models.

METHODOLOGY

Simulation framework

The computational framework for this study integrates two main platforms: the MATLAB Reservoir Simulation Toolbox (MRST) and Python. MRST, developed by SINTEF Digital (Lie 2019), served as the reservoir simulation environment. Its open-source and modular design makes it particularly suitable for research purposes, as it allows for the seamless incorporation of prototype methods and numerical experiments. During the simulation run, MRST formulates a system of nonlinear equations arising from the discretization of the governing flow equations. Instead of solving these equations using their default solvers, MRST is configured to call an external Python function containing the proposed solver algorithm. This setup ensures that the Python-based solver is directly embedded within the MRST simulation workflow. MRST provides the system matrices and residuals, whereas Python executes the solver computations and returns the solution to the MRST for the subsequent simulation steps.

Hybrid probabilistic – backpropagation neural network solver (Prob-BPNN)

A complete explanation of the technique for solving nonlinear systems using a backpropagation neural network (BPNN) can be found in Goulianas et al. (2018). This technique works by constructing a neural network that is a direct structural mirror of a system of equations that must be solved. The output neurons are specifically designed to calculate the value of each equation in the system. In summary, four layers are involved in the feedforward neural network architecture, with each layer as follows: 1). Layer 0 (input): A single input neuron that always has a value of one; 2). Layer 1 (the solution vector) contains n neurons, where n is the number of variables in the system ($x_1, x_2, \dots, x_n$). The synaptic weights connecting layers 0 to 1 were the only variable weights in the entire network. After successful training, the values of these weights become components of the system root; 3). Layer 2 (term calculation): This layer comprises specialized

neurons that calculate each multiplicative term. The weights connecting layers 1 and 2 are fixed and correspond to the exponents in each term of the equations; 4). Layer 3 (output) contained n output neurons, one for each system equation. Each neuron sums the terms calculated in layer 2 to reconstruct its corresponding polynomial equation. The weights connecting layers 2 and 3 are also fixed, and represent the coefficient terms.

For a given phase (e.g., oil, gas, and water) in a specific grid cell i , the residual equation, R_i , at a new time level, $n+1$, is written as

$$R_i = [Accumulation] - [Flux] - (Source / Sink) = 0 \quad (1)$$

The accumulation and flux terms are represented by the main structure of the network, whereas the source/sink terms correspond to a fixed bias in the BPNN. We use the identity function as an activation function and an adaptive learning rate, β , defined as:

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

Probabilistic linear solvers (Wenger & Hennig 2020) reformulate the solution for linear systems as a Bayesian inference problem. Unlike conventional deterministic solvers that provide a single-point estimate, probabilistic solvers yield a probability distribution over the solution space.

Suppose we have a task of solving nonlinear equations:

$$f(x) = 0, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (3)$$

At iteration k , the nonlinear function is approximated using a first-order Taylor expansion, as follows:

$$f(x_k + \delta) \approx f(x_k) + J(x_k)\delta \quad (4)$$

where $J(x_k) = \frac{\partial f}{\partial x} \Big|_{x_k}$ is the Jacobian. This transforms the nonlinear problem into a local linear system, as follow:

$$J(x_k)\delta \approx -f(x_k) \quad (5)$$

Instead of solving the linear system exactly, a probabilistic model is imposed:

$$\delta \sim \mathcal{N}(\mu_0, \Sigma_0) \quad (6)$$

where μ_0 and Σ_0 encode prior belief about the solution update. After observing the matrix vector products, for example $J(x_k)v$, Bayesian updating yields a posterior:

$$p(\delta \mid \text{observations}) = \mathcal{N}(\mu_k, \Sigma_k) \quad (7)$$

The mean μ_k acts as the solution update, while Σ_k quantifies uncertainty. Subsequently, the iteration is obtained as

$$x_{k+1} = x_k + \mu_k \quad (8)$$

In this study, we used a BPNN to make an initial guess for the probabilistic solver. The workflow of the hybrid probabilistic-backpropagation neural network (Prob-BPNN) solver is shown in Figure 2, and can be explained as follows: 1). Take as input: the matrix-vector multiplication operator $A(\cdot)$, the vector b , and optionally a prior distribution for A and its inverse H ; 2). Make an initial guess x_0 : The value is calculated using the BPNN approach; 3). Calculate the residual norm, r_0 : The residual measures the error level of the current solution, i.e., how far Ax_0 is from the target b .

Steps 4 through 10 are iterative procedures until satisfy the solution criteria:

$$\sqrt{\text{tr}(\text{Cov}[\mathbf{x}])} > \max(\delta_{rtol} \|\mathbf{b}\|_2, \delta_{atol}) \quad (9)$$

4). Compute action or search direction via policy, s_i : This direction is calculated by applying the expectation of the inverse matrix $E[H]$ to the negative of the previous residual r_{i-1} ; 5). Make observations, y_i : This process applies matrix A in the direction of search s_i ; 6). Calculate the optimal step size, α_i : This process determines how far to move along the search direction s_i ; 7). Update the solution estimate, x_i : The estimate of x_{i-1} is moved by α_i toward the search for

s_i to obtain a new, better estimate, x_i . 8). Update the residual, r_i : This is a more computationally efficient way to obtain the new residual r_i without recalculating $Ax_i - b$; 9). Infer posterior distributions: This process uses the collected information to infer new posterior distributions for the matrix A and its inverse H ; 10). Calibrate the uncertainty: This process uses all search directions (S) and observation results (Y) that have been collected to adjust uncertainties (Φ, Ψ) .

After the iteration stops, this solver defines the final confidence about the solution x .

RESULTS AND DISCUSSION

In this study, we evaluated the performance of the solver using two reservoir geometry models, namely, Cases 1 and 2, as shown in Figure 3. Case 1 had dimensions of 200, 200, and 50 m in the x-, y-, and z-directions, respectively, with a grid configuration of $5 \times 4 \times 5$, resulting in 100 cells (Figure 3a). It consisted of five vertical layers, each 10 m in thickness. In the x-direction, local grid refinement was applied at $x = 100$ m with a thickness of 0.001 m to represent the fracture. The fracture grids are characterized by a porosity of 0.99 and a permeability of 5000 mD, whereas the surrounding matrix grids have a porosity of 0.05 and a permeability of 0.1 mD. A single vertical production well (P1) was located at the edge of the model, extending from a depth of 0 to 30 m.

In Case 2, we used an unstructured grid with a configuration of $6 \times 6 \times 5$ (Figure 3b). A homogeneous reservoir model was employed with uniform rock properties throughout the domain, including a permeability of 30 mD and porosity of 0.3. In both cases, we used a single-phase oil system with a density of 750 kg/m³ and a viscosity of 5 cp. The properties of the Jacobian matrix resulting from both cases are listed in Table 1.

Table 1. Properties of the Jacobian matrix

	Case 1	Case 3
Matrix size	102 × 102	182 × 182
Number of elements	10,404	33,404
Number of nonzero elements	579	1327
Condition number	2.3×10^8	2.8×10^8

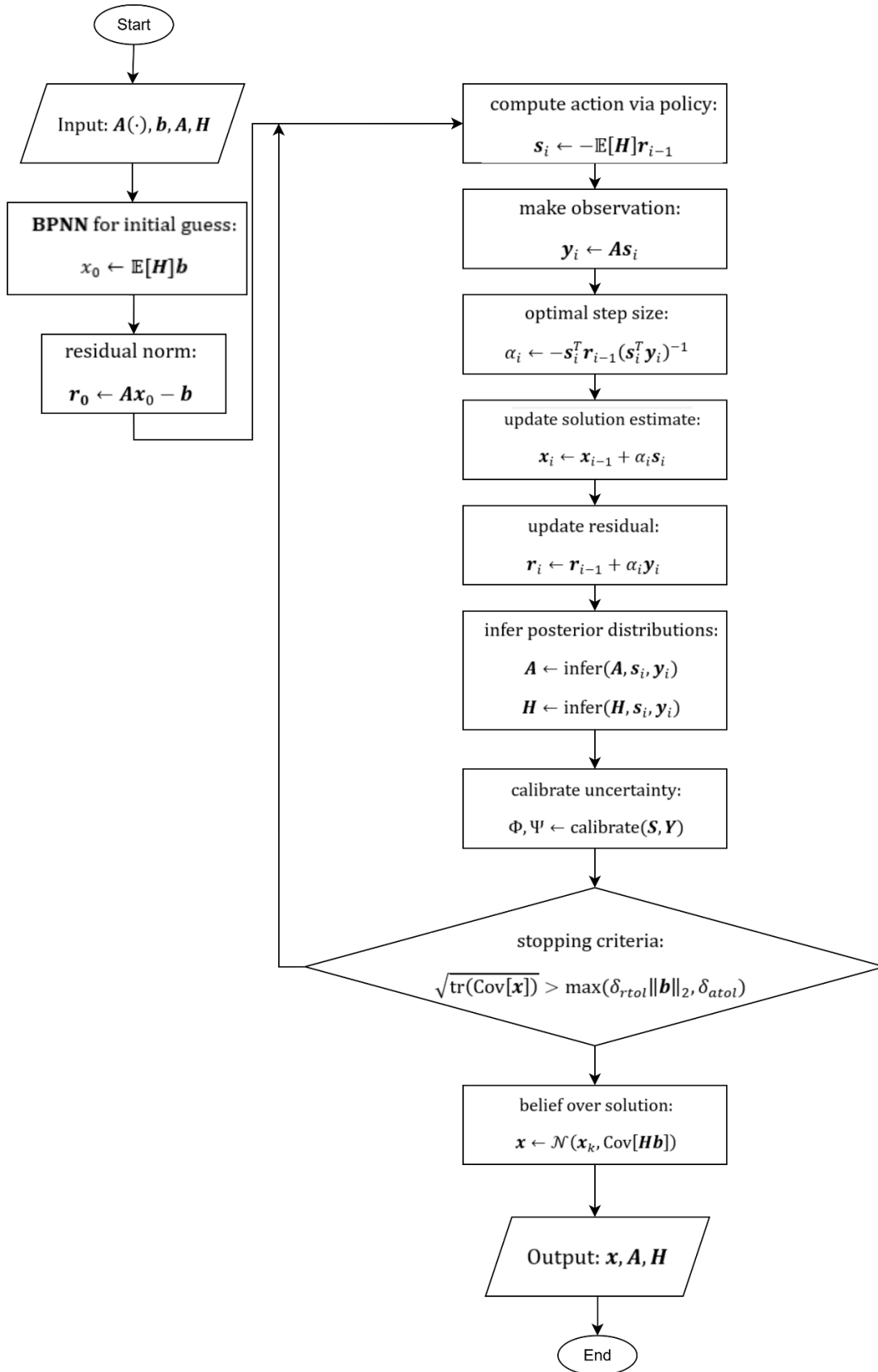


Figure 2. Workflow for solving nonlinear systems in reservoir simulation using hybrid probabilistic-backpropagation neural network (Prob-BPNN) solver.

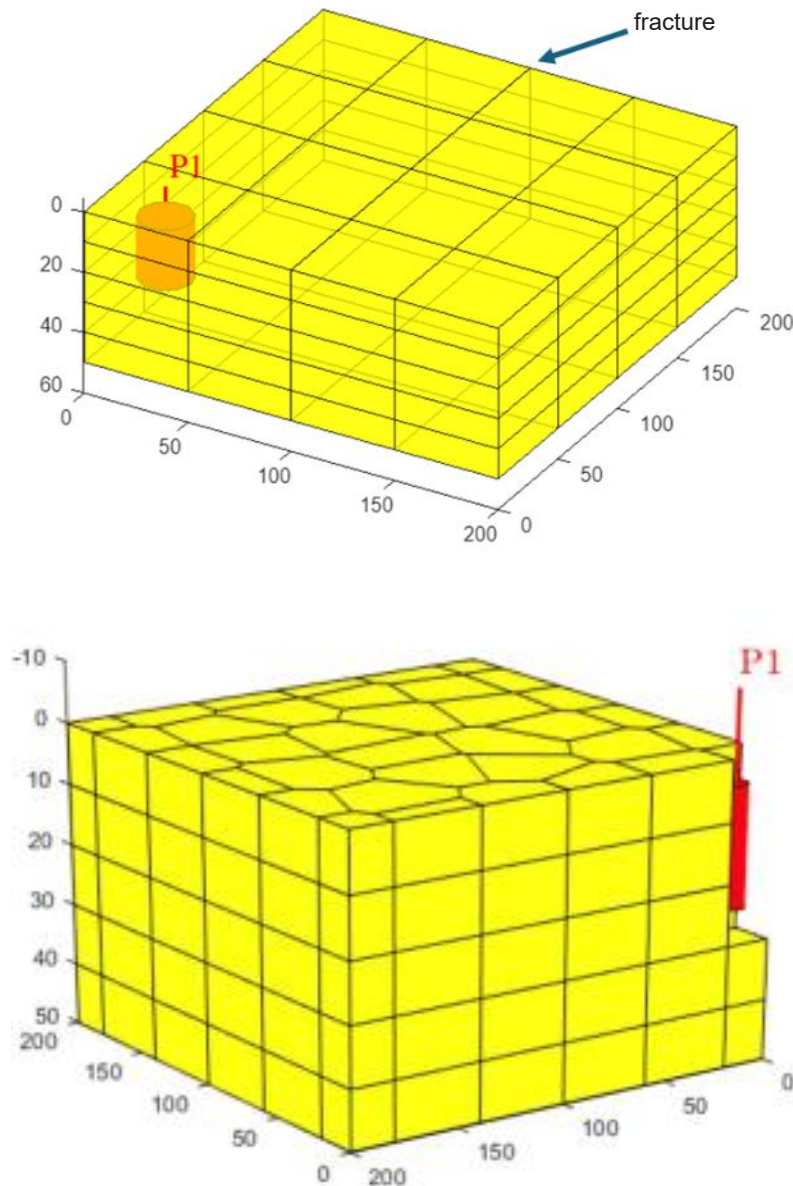


Figure 3. Model geometries for (a). Case 1 and (b). Case 2.

Simulation results

Figure 4 presents the oil production rate profiles for Case 1, comparing the different nonlinear solvers. In this study, the GMRES solution computed with a tolerance of 10^{-10} was used as the reference solution. In Figure 4a, the production trends obtained using the proposed Prob-BPNN are compared with those obtained using the GMRES method. Both methods exhibited a consistent decline in oil production over time, with Prob-BPNN closely following the GMRES curve, thereby demonstrating the robustness and accuracy of the proposed solver in capturing the production behavior.

As shown in Figure 4b, the CG and BiCG-STAB methods failed to reproduce a physically meaningful

production profile, yielding results that deviated substantially from the reference solution. Moreover, the BiCG-STAB method shows large oscillations and occasional negative values, further underscoring the numerical instability. These discrepancies highlight that both CG and BiCG-STAB introduce significant errors when applied to nonlinear reservoir simulations, thereby limiting their reliability compared to Prob-BPNN and GMRES.

Figure 5 illustrates the computational efficiencies of the different nonlinear solvers for Case 1. The Prob-BPNN method consistently requires a higher computational effort than the other methods, with CPU times fluctuating between 0.1–0.7 seconds. This increased cost arises from the additional probabilistic

inference and neural network evaluation steps embedded in the solver. By contrast, GMRES, CG, and BiCG-STAB exhibit markedly lower CPU times, generally remaining below 0.1 seconds throughout the simulation. Among them, the CG achieved the lowest computational demand, reflecting its relatively simple algorithmic structure.

However, as discussed for the production profiles (Figure 4), the computational advantage of CG is offset by its inability to generate physically

meaningful results. Similarly, BiCG-STAB suffers from instability issues in production predictions, despite its competitive efficiency. GMRES offers a more balanced trade-off, maintaining moderate CPU times and reliable accuracy. Overall, although Prob-BPNN incurs a higher computational burden, it provides stable and physically consistent results, highlighting the importance of considering both numerical stability and efficiency when evaluating solver performance.

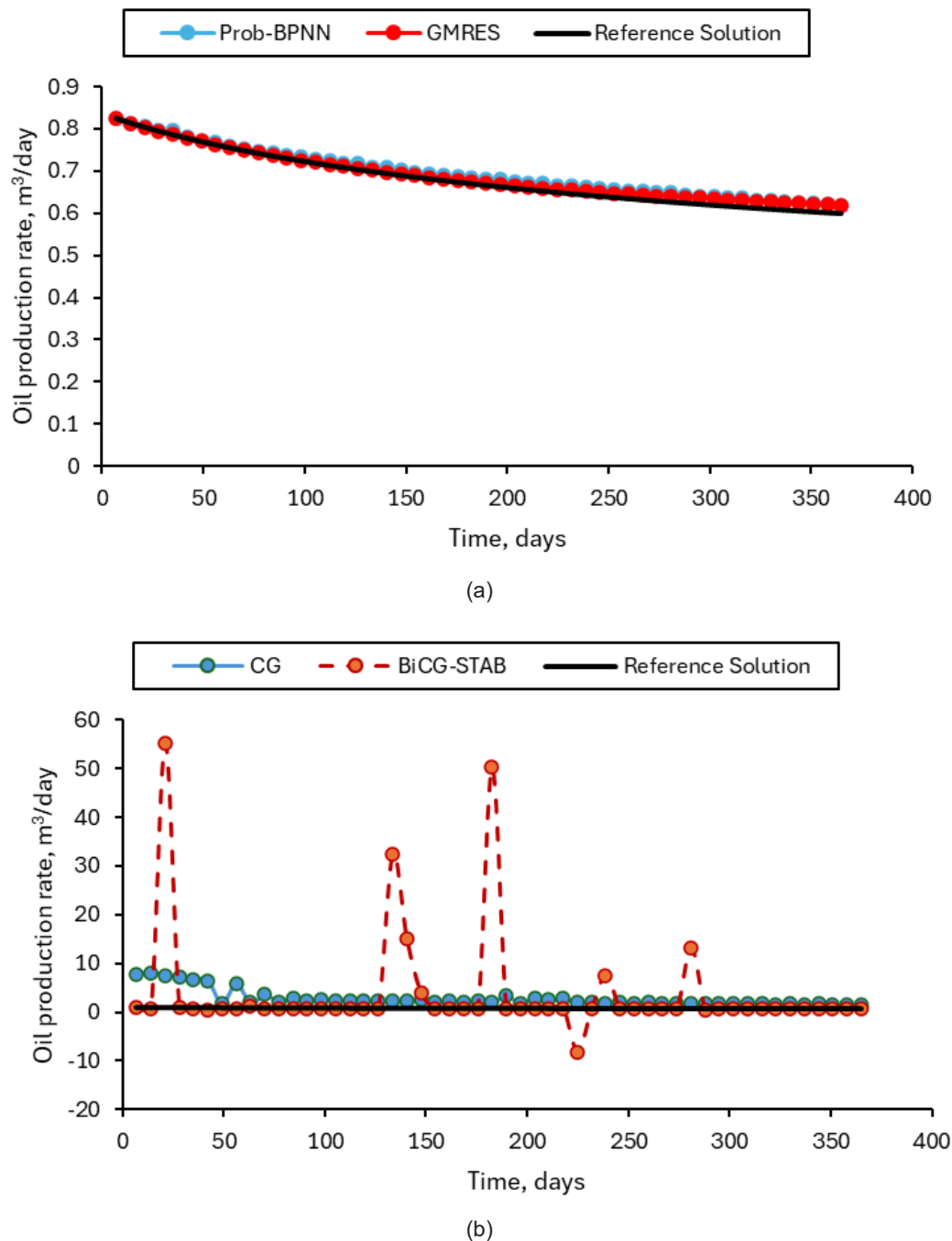


Figure 4. Oil production rate for Case 1, result from (a). Prob-BPNN and GMRES (b). CG and BiCG-STAB.

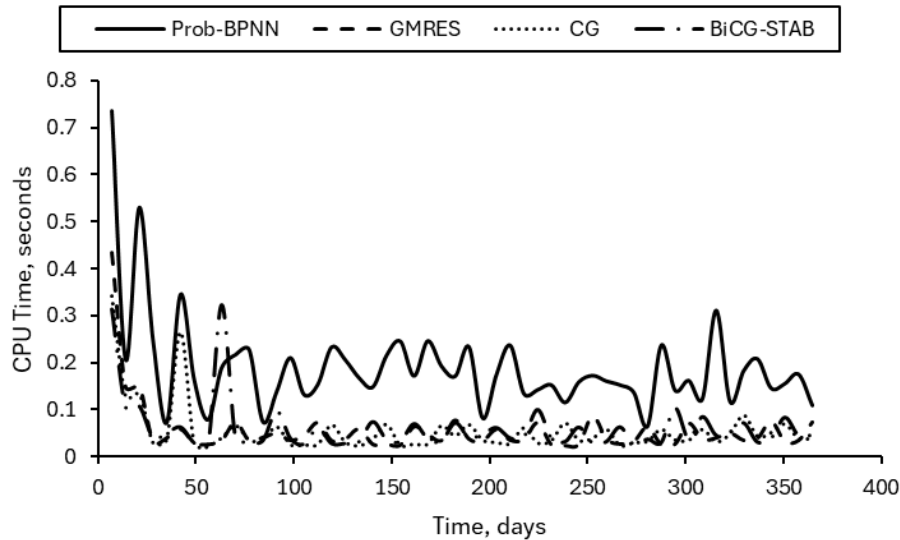


Figure 5. Computational time for Case 1.

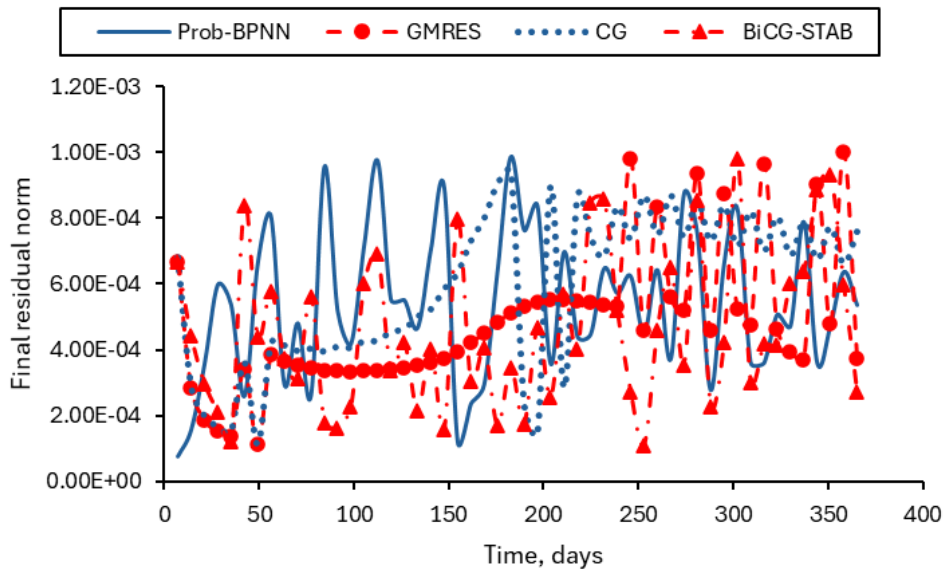


Figure 6. The final residual norm calculated for Case 1. The oscillation of residual values observed in Prob-BPNN, GMRES, CG, and BiCG-STAB indicates potential issues with convergence stability. Among these methods, GMRES demonstrates the highest stability up to approximately $t = 250$ days.

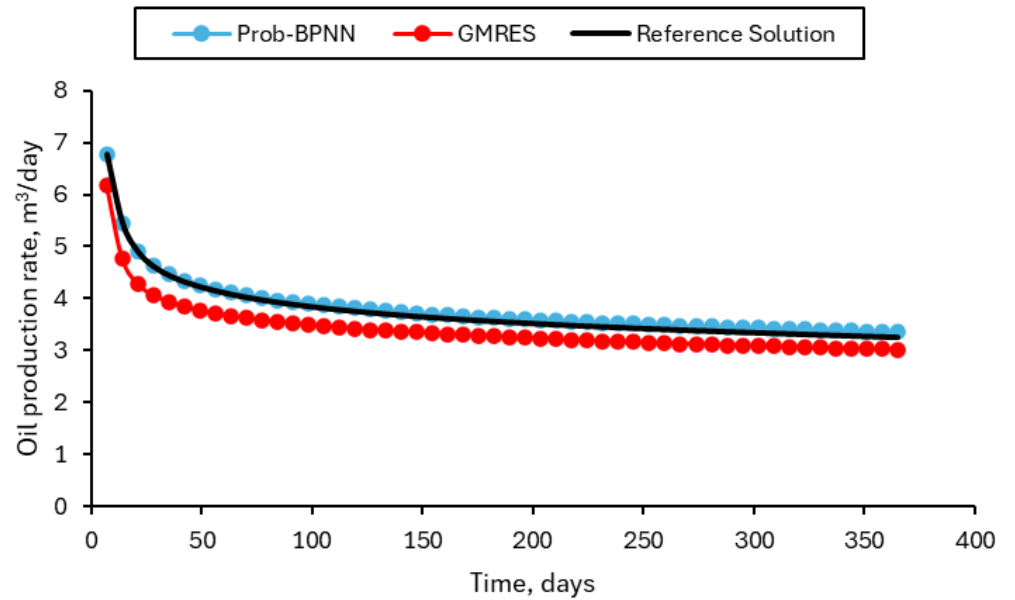
Figure 6 shows the evolution of the final residual norm obtained from different nonlinear solvers throughout the simulation for Case 1. The Prob-BPNN method exhibits oscillatory behavior in the residuals, but remains consistently within the order of 10^{-4} , indicating convergence stability despite fluctuations. GMRES demonstrated an overall smoother trend, although its residuals gradually increased with simulation time, suggesting potential challenges in maintaining accuracy in later stages. In contrast, both CG and BiCG-STAB displayed pronounced instabilities with large variations in the residual norm across the simulation. These erratic behaviors reflect poor convergence characteristics

and highlight their limited robustness when applied to nonlinear reservoir problems. In summary, the results confirm that Prob-BPNN and GMRES offer a more reliable convergence performance, whereas CG and BiCG-STAB suffer from significant instability.

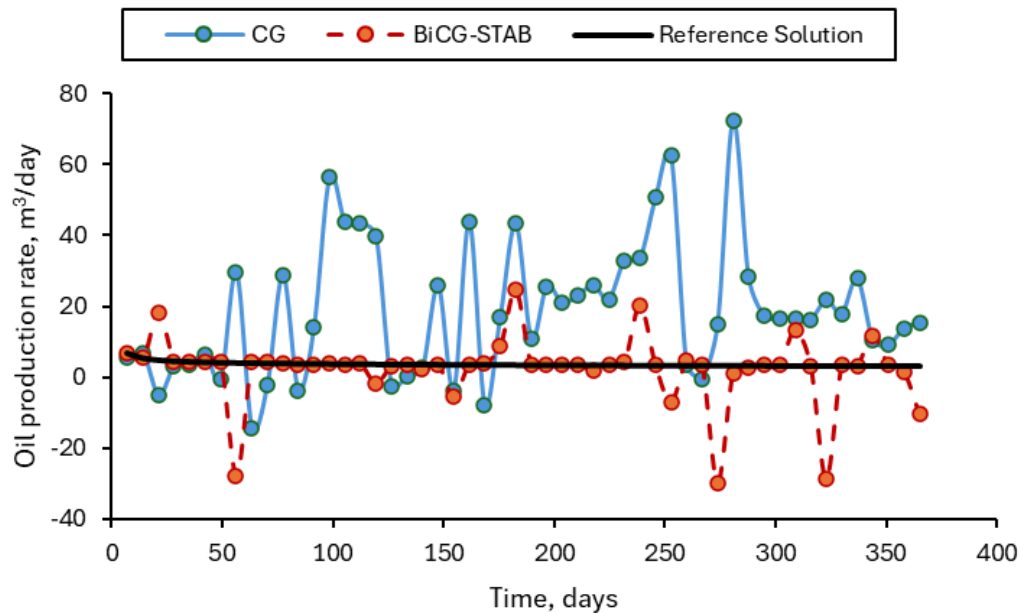
Figure 7 shows the oil production rate profiles for Case 2. In Figure 7a, the proposed Prob-BPNN and GMRES capture the characteristic production decline trend, with Prob-BPNN closely following GMRES. This consistency indicates that Prob-BPNN provides physically meaningful results and maintains robustness throughout the simulation. In contrast, Figure 7b highlights the limitations of CG and BiCG-

STAB. The CG solver produced highly irregular and oscillatory production rates, including physically implausible spikes that deviated significantly from the expected reservoir behavior. Similarly, BiCG-STAB displayed numerical instabilities with abrupt

fluctuations and unrealistic negative production rates. The results confirmed that Prob-BPNN and GMRES yielded stable and physically consistent production trends, whereas CG and BiCG-STAB exhibited convergence and stability issues.



(a)



(b)

Figure 7. Oil production rates for Case 2, (a) Prob-BPNN produces a more accurate solution compared to GMRES, whereas (b) CG and BiCG-STAB yield unrealistic results, exhibiting oscillatory and even negative oil production rates.

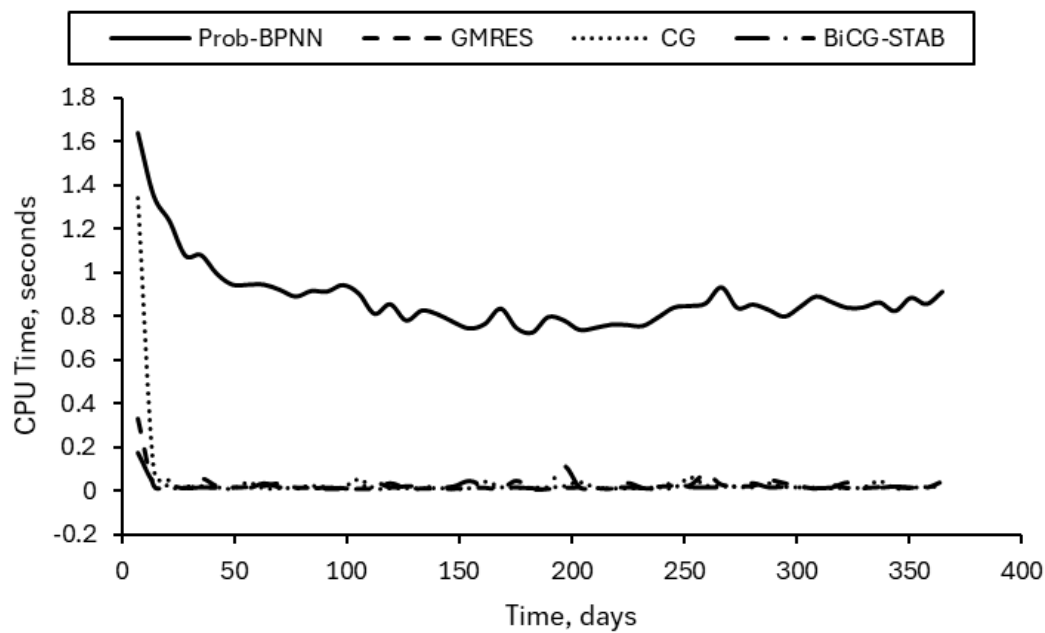


Figure 8. Computational time for Case 2. Prob-BPNN requires a total computation time approximately 19–38 times longer than that of GMRES, CG, and BiCG-STAB. Nevertheless, Prob-BPNN provides the most accurate solution among all methods.

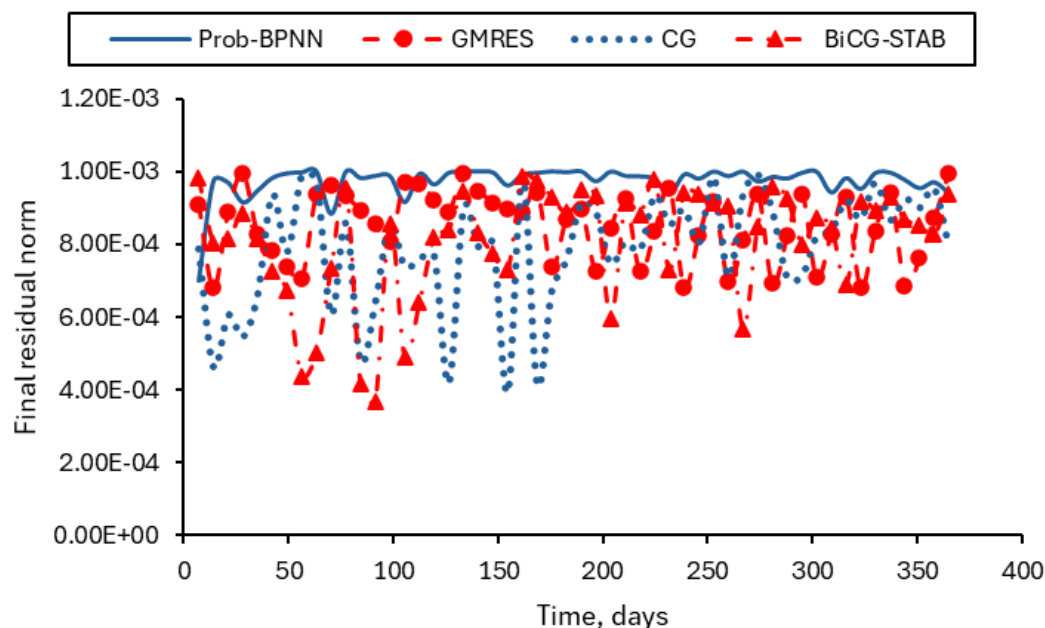


Figure 9. The final residual norm for Case 2. The oscillation of residual values in GMRES, CG, and BiCG-STAB indicates issues with convergence stability, whereas Prob-BPNN exhibits the most stable residual behavior.

Figure 8 compares the computational efficiencies of the nonlinear solvers for Case 2. The Prob-BPNN solver consistently required the highest computational cost, with CPU times ranging from approximately 0.8 to 1.6 seconds. In contrast, GMRES, CG, and BiCG-STAB achieved significantly shorter CPU times. Among these, CG demonstrated the fastest execution, benefiting from its relatively simple

algorithmic structure. However, as highlighted in the production profiles (Figure 7b), the computational advantage of CG is offset by its inability to provide physically meaningful or stable results. Although Prob-BPNN is computationally more demanding, it provides stable and physically consistent outcomes, whereas CG and BiCG-STAB are prone to numerical instabilities.

Figure 9 shows the evolution of the final residual norm for Case 2, comparing the performances of Prob-BPNN, GMRES, CG, and BiCG-STAB. The Prob-BPNN solver maintained residual values consistently below the order of 10^{-3} , with relatively smooth fluctuations over time, indicating convergence stability. GMRES also remained within the same magnitude, although with slightly more scattered variations, suggesting comparable accuracy but with less stability in later simulation stages. In contrast, CG and BiCG-STAB displayed pronounced oscillations throughout the simulation, with sharp drops and spikes in residual values. These erratic patterns are symptomatic of poor convergence behavior, reflecting the limited robustness of these solvers when applied to nonlinear systems in reservoir simulations.

Table 2 summarizes the performance metrics of the nonlinear solvers for Cases 1 and 2. As mentioned previously, we used a GMRES solution with a tight tolerance of 10^{-10} as the reference solution. For Case 1, both Prob-BPNN and GMRES achieved very low errors (MAE < 0.02, RMSE < 0.02, MAPE < 3%) with high R^2 values (0.945 and 0.978, respectively), indicating strong agreement with the reference solution. In contrast, CG and BiCG-STAB exhibited substantial deviations with MAE values of 2.091 and 3.512, RMSE values exceeding 2.7, and extremely high MAPE (292% and 497%). Their negative R^2 values (−1877.487 for CG and −34072.207 for BiCG-STAB) reflect poor correlation with the reference solution. For Case 2, Prob-BPNN maintains accuracy, with MAE = 0.066,

RMSE = 0.071, MAPE $\approx$ 1.88%, and $R^2 = 0.986$, all indicating close agreement with the reference solution. The GMRES remained accurate but showed a moderate increase in error (MAE = 0.329, RMSE = 0.345, $R^2 = 0.677$). Conversely, CG and BiCG-STAB failed to reproduce the reference behavior, with very large errors (MAE > 18, RMSE > 9.5, and MAPE exceeding 100%) and strongly negative R^2 values. These quantitative performance metrics corroborate the oil production rate profile findings. Prob-BPNN consistently achieved accuracy close to that of the reference solution, whereas CG and BiCG-STAB yielded unreliable and physically implausible results.

Table 3 presents a comparison of the solver performances in terms of the iteration count, residual statistics, and total CPU time for Cases 1 and 2. In Case 1, GMRES and CG required significantly fewer iterations (182 and 288, respectively) than Prob-BPNN (16,603) and BiCG-STAB (343). GMRES was the most efficient solver in this case, requiring the shortest CPU time (2.8479 s). In Case 2, the differences were more pronounced. Prob-BPNN performed an extremely large number of iterations (653,345), far exceeding those of GMRES (260), CG (747), and BiCG-STAB (605). Nevertheless, Prob-BPNN achieved a residual of less than 10^{-3} as a Krylov solver, even though it had a substantially higher CPU time (45.8955 s). GMRES demonstrated the most efficient performance, solving the system in only 1.5277 s, whereas CG and BiCG-STAB remained moderately efficient (2.4412 s and 1.3358 s, respectively).

Table 2. Performance metrics of the nonlinear solvers, with the tolerance of 10^{-10} as the reference solution.

	Prob-BPNN	GMRES	CG	BiCG-STAB
Case 1				
MAE	0.013	0.006	2.091	3.512
RMSE	0.014	0.009	2.708	11.533
MAPE (%)	2.036	1.104	292.499	496.716
R^2	0.945	0.978	-1877.487	-34072.207
Case 2				
MAE	0.066	0.329	18.414	4.329
RMSE	0.071	0.345	24.273	9.506
MAPE (%)	1.882	8.639	514.998	121.097
R^2	0.986	0.677	-1588.612	-242.796

Table 3. Iteration counts, residual statistics, and CPU times for different nonlinear solvers, with the tolerance of 10^{-10} as the reference solution.

	Prob-BPNN	GMRES	CG	BiCG-STAB
Case 1				
Total iteration	16603	182	288	343
Minimum residual	7.45E-05	1.12E-04	1.15E-04	1.10E-04
Maximum residual	9.84E-04	9.99E-04	9.42E-04	9.79E-04
Average residual	5.42E-04	4.80E-04	5.90E-04	4.56E-04
Total CPU time (s)	9.958	2.8479	2.8794	3.119
Case 2				
Total iteration	653345	260	747	605
Minimum residual	7.02E-04	6.80E-04	3.94E-04	3.68E-04
Maximum residual	1.00E-03	9.97E-04	9.93E-04	9.86E-04
Average residual	9.73E-04	8.53E-04	7.90E-04	8.12E-04
Total CPU time (s)	45.895	1.527	2.441	1.335

CONCLUSION

This study introduced and evaluated a hybrid probabilistic backpropagation neural network (Prob-BPNN) solver for nonlinear systems in reservoir simulations. The performance of the proposed solver was evaluated using two reservoir model cases against widely used Krylov subspace methods: GMRES, CG, and BiCG-STAB.

The results demonstrate that the Prob-BPNN solver is preferable in scenarios in which accuracy and numerical stability are prioritized over computational efficiency. The solver consistently reproduced physically meaningful production profiles, with quantitative errors ($MAE \leq 0.066$, $RMSE \leq 0.071$, $MAPE \leq 2.04\%$, and $R^2 \geq 0.945$) that were comparable to the reference solution from GMRES at a tight tolerance of 10^{-10} . By contrast, CG and BiCG-STAB often failed to capture the correct reservoir dynamics, yielding very high errors (e.g., MAPE exceeding 292% in Case 1 and 514% in Case 2) and negative R^2 values, which highlight their instability.

In terms of the computational cost, Prob-BPNN incurred significantly higher CPU times than the Krylov solvers. For example, in Case 1, the solver required 9.96 s compared to 2.85 s for GMRES, and in Case 2, it required 45.90 s compared to only 1.53 s for GMRES. These differences are explained by the additional inference steps introduced by the neural network initialization and the probabilistic updates embedded in the algorithm. While GMRES remained the most computationally efficient method, Prob-BPNN achieved residual convergence on the same order of magnitude (below 10^{-3}), demonstrating its

reliability despite higher iteration counts. The findings suggest that Prob-BPNN is particularly suitable when reservoir studies require accuracy and robustness against instability. However, computational overhead is a clear limitation. Therefore, future research should focus on reducing execution time, for example, by leveraging parallelization strategies, GPU acceleration, or hybridization with Krylov methods to improve scalability.

ACKNOWLEDGEMENT

This work was supported and funded by the PPMI FTTM Program 2025 of the Faculty of Mining and Petroleum Engineering at the ITB.

REFERENCES

Alakeely, A., & Horne, R. (2022): Simulating Oil and Water Production in Reservoirs with Generative Deep Learning, SPE Reservoir Evaluation & Engineering, 25(04), 751–773. <https://doi.org/10.2118/206126-PA>.

Aliaga, J. I., Anzt, H., Grützmacher, T., Quintana-Ortí, E. S., and Tomás, A. E. (2023): Compressed basis GMRES on high-performance graphics processing units, The International Journal of High Performance Computing Applications, 37(2), 82–100. <https://doi.org/10.1177/10943420221115140>.

Almajid, M. M., and Abu-Al-Saud, M. O. (2022): Prediction of porous media fluid flow using physics informed neural networks, Journal of

- Petroleum Science and Engineering, 208, 109205. <https://doi.org/10.1016/j.petrol.2021.109205>.
- Alpak, F. O., Jammoul, M., and Wheeler, M. F. (2023): Consistent Discretization Methods for Reservoir Simulation on Cut-Cell Grids, Presented at the SPE Reservoir Simulation Conference, D011S003R001. <https://doi.org/10.2118/212213-MS>.
- Asif, A., Abd, A. S., and Abushaikha, A. (2025): Advanced Linearization Methods for Efficient and Accurate Compositional Reservoir Simulations, *Computation*, 13(8), 191. <https://doi.org/10.3390/computation13080191>.
- Bakhvalov, N. S. (1966): On the convergence of a relaxation method with natural constraints on the elliptic operator, *USSR Computational Mathematics and Mathematical Physics*, 6(5), 101–135. [https://doi.org/10.1016/0041-5553\(66\)90118-2](https://doi.org/10.1016/0041-5553(66)90118-2).
- Benzi, M. (2002): Preconditioning Techniques for Large Linear Systems: A Survey, *Journal of Computational Physics*, 182(2), 418–477. <https://doi.org/10.1006/jcph.2002.7176>.
- Benzi, M., and Tûma, M. (1999): A comparative study of sparse approximate inverse preconditioners, *Applied Numerical Mathematics*, 30(2), 305–340. [https://doi.org/10.1016/S0168-9274\(98\)00118-4](https://doi.org/10.1016/S0168-9274(98)00118-4).
- Bhogeswara, R., and Killough, J. E. (1994): Parallel Linear Solvers for Reservoir Simulation: A Generic Approach for Existing and Emerging Computer Architectures, *SPE Computer Applications*, 6(01), 5–11. <https://doi.org/10.2118/25240-PA>.
- Chen, H., Terada, K., Li, A., and Datta-Gupta, A. (2022): RAmerican petroleum institute d Simulation of Unconventional Reservoirs Using Multi-Domain Multi-Resolution Discretization Based on the Diffusive Time of Flight, Presented at the SPE/AAPG/SEG Unconventional Resources Technology Conference, OnePetro. <https://doi.org/10.15530/urtec-2022-3723026>.
- Duran, A., and Tuncel, M. (2014): Spectral Effects of Large Matrices from Oil Reservoir Simulators on Performance of Scalable Direct Solvers, Presented at the SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference and Exhibition, SPE-172984-MS. <https://doi.org/10.2118/172984-MS>.
- Gasmi, C. F., and Tchelepi, H. (April 22, 2021): Physics Informed Deep Learning for Flow and Transport in Porous Media, arXiv. <https://doi.org/10.48550/arXiv.2104.02629>
- Gasmi, C. F., and Tchelepi, H. (May 19, 2022): Uncertainty Quantification for Transport in Porous media using Parameterized Physics Informed neural Networks, arXiv. <https://doi.org/10.48550/arXiv.2205.12730>.
- Gasparini, L., Rodrigues, J. R. P., Augusto, D. A., Carvalho, L. M., Conopoima, C., Goldfeld, P., Panetta, J., Ramirez, J. P., Souza, M., Figueiredo, M. O., and Leite, V. M. D. M. (2021): Hybrid parallel iterative sparse linear solver framework for reservoir geomechanical and flow simulation, *Journal of Computational Science*, 51, 101330. <https://doi.org/10.1016/j.jocs.2021.101330>.
- Gharieb, A., El Hamady, A., Gad, R., Lairer, B. B., and Rivas-Rivas, S. (2024): Optimizing Field Development in Data-Starved Reservoirs: Mitigating Economic Risks Through Stochastic Production Profiling and Fully Implicit Static-Dynamic Simulation, Presented at the Mediterranean Offshore Conference, D011S007R002. <https://doi.org/10.2118/223239-MS>.
- Goulianas, K., Margaritis, A., Refanidis, I., and Diamantaras, K. (2018): Solving polynomial systems using a fast adaptive back propagation-type neural network algorithm, *European Journal of Applied Mathematics*, 29(2), 301–337. <https://doi.org/10.1017/S0956792517000146>.
- Habib, M., and Joslin, K. (2020): Retrospective Validation of the Robustness of Reservoir Simulation Predictions, Presented at the SPE Canada Heavy Oil Conference, D021S003R003. <https://doi.org/10.2118/200023-MS>.
- Han, J.-X., Xue, L., Wei, Y.-S., Qi, Y.-D., Wang, J.-L., Liu, Y.-T., and Zhang, Y.-Q. (2023): Physics-informed neural network-based petroleum reservoir simulation with sparse data using domain decomposition, *Petroleum Science*, 20(6), 3450–3460. <https://doi.org/10.1016/j.petsci.2023.10.019>.
- He, K., Tan, S. X.-D., Zhao, H., Liu, X.-X., Wang,

- H., and Shi, G. (2016): Parallel GMRES solver for fast analysis of large linear dynamic systems on GPU platforms, *Integration*, 52, 10–22. <https://doi.org/10.1016/j.vlsi.2015.07.005>.
- Hestenes, M. R., and Stiefel, E. (1952): Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards*, 49(6), 409. <https://doi.org/10.6028/jres.049.044>.
- Hørsholt, S., Nick, H., and Jørgensen, J. B. (2019): A Hierarchical Multigrid Method for Oil Production Optimization, 12th IFAC Symposium on Dynamics and Control of Process Systems, Including Biosystems DYCOPS 2019, 52(1), 492–497. <https://doi.org/10.1016/j.ifacol.2019.06.110>.
- Isaiah, J., Schrader, S., Reichhardt, D., and Link, C. (2013): Performing Reservoir Simulation with Neural Network Enhanced Data, SPE Digital Energy Conference, SPE, The Woodlands, Texas, USA, SPE-163691-MS. <https://doi.org/10.2118/163691-MS>.
- Jammoul, M., Alpak, F. O., and Wheeler, M. F. (2023): An Enriched Galerkin Discretization Scheme for Two Phase Flow on Non-Orthogonal Grids, Presented at the SPE Reservoir Simulation Conference, D011S003R002. <https://doi.org/10.2118/212238-MS>.
- Jiang, J., and Pan, H. (2022): Dissipation-Based Nonlinear Solver for Fully Implicit Compositional Simulation, *SPE Journal*, 27(04), 1989–2014. <https://doi.org/10.2118/209233-PA>.
- Kang, Z., Deng, Z., Han, W., and Zhang, D. (2018): Parallel Reservoir Simulation with OpenACC and Domain Decomposition, *Algorithms*, 11(12), 213. <https://doi.org/10.3390/a11120213>.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021): Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 422–440. <https://doi.org/10.1038/s42254-021-00314-5>.
- Khait, M., and Voskov, D. V. (2017): Operator-based linearization for general purpose reservoir simulation, *Journal of Petroleum Science and Engineering*, 157, 990–998. <https://doi.org/10.1016/j.petrol.2017.08.009>.
- Kim, Y., Jang, H., Kim, J., and Lee, J. (2017): Prediction of storage efficiency on CO₂ sequestration in deep saline aquifers using artificial neural network, *Applied Energy*, 185, 916–928.
- Kristanto, D., Hariyadi, H., Pramudyohadi, E. W., Kurniawan, A., Nursidik, U. S., Asmorowati, D., ... & Cahyaningtyas, N. (2025). Optimization of Co₂ Injection Through Cyclic Huff and Puff to Improve Oil Recovery. *Scientific Contributions Oil and Gas*, 48(2), 53-67. <https://doi.org/10.29017/scog.v48i2.1659>.
- Li, L., and Abushaikh, A. (2022): A fully-implicit parallel framework for complex reservoir simulation with mimetic finite difference discretization and operator-based linearization, *Computational Geosciences*, 26(4), 915–931. <https://doi.org/10.1007/s10596-021-10096-5>.
- Lie, K.-A. (2019): An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST), Cambridge University Press, Cambridge. <https://doi.org/10.1017/9781108591416>.
- Mamo, N. B., and Dennis, Y. A. (2020): Artificial neural network based production forecasting for a hydrocarbon reservoir under water injection, *Petroleum Exploration and Development*, 47(2), 383–392. [https://doi.org/10.1016/S1876-3804\(20\)60055-6](https://doi.org/10.1016/S1876-3804(20)60055-6).
- Manea, A. M., Sewall, J. ., and Tchelepi, H. A. (2016): Parallel Multiscale Linear Solver for Highly Detailed Reservoir Models, *SPE Journal*, 21(06), 2062–2078. <https://doi.org/10.2118/173259-PA>.
- Mithani, A. H., Rosland, E. A., Jamaludin, M. A., W Ismail, W. R., Lajawi, M. T., and A Salam, I. H. (2022): Reservoir Souring Simulation and Modelling Study for Field with Long History of Water Injection: History Matching, Prediction, Presented at the SPE Asia Pacific Oil & Gas Conference and Exhibition, D021S010R003. <https://doi.org/10.2118/210778-MS>.
- Raissi, M., Perdikaris, P., and 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. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- Saad, Y., and Schultz, M. H. (1986): GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM Journal on Scientific and Statistical Computing*, 7(3), 856–869. <https://doi.org/10.1137/0907058>
- Stüben, K., Clees, T., Klie, H., Lu, B., and Wheeler, M. F. (2007): Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation, Presented at the SPE Reservoir Simulation Symposium, SPE-105832-MS. <https://doi.org/10.2118/105832-MS>.
- Sugihardjo, S. (2022). CCUS-Aksi Mitigasi Gas Rumah Kaca dan Peningkatan Pengurusan Minyak CO₂-EOR. Lembaran publikasi minyak dan gas bumi, 56 (1), 21–35. <https://doi.org/10.29017/LPMGB.56.1.916>.
- Swadesi, B., Sanmurjana, M., Lubis, M. R. R., Kristanto, D., Cahyaningtyas, N., & Widiyaningsih, I. (2025). A Simulation Study on Polymer Mobility Design Strategies and Their Impact on Oil Recovery Efficiency and Displacement Mechanisms. *Scientific Contributions Oil and Gas*, 48(2), 111-129. <https://doi.org/10.29017/scog.v48i2.1661>.
- Tokuda, T., and Hashimoto, R. (2023): Development of an Efficient Parallelization Scheme for Fully Implicit Discontinuous Deformation Analysis (DDA), Presented at the 15th ISRM Congress, retrieved September 24, 2025 from internet: , ISRM-15CONGRESS-2023-306.
- van der Vorst, H. A. (1992): Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, *SIAM Journal on Scientific and Statistical Computing*, 13(2), 631–644. <https://doi.org/10.1137/0913035>
- Wenger, J., and Hennig, P. (October 2020): Probabilistic Linear Solvers for Machine Learning, *arXiv*. <https://doi.org/10.48550/arXiv.2010.09691>
- Yan, L., Zhou, Z., Li, B., Fang, Q., Kai, Y., and Shi, Z. (2025): A Reservoir Dynamic Prediction Model Based on the REROSIM Method: Digital Twin and Simulation Studies Driven by Digital Transformation, Presented at the Middle East Oil, Gas and Geosciences Show (MEOS GEO), D031S096R002. <https://doi.org/10.2118/227319-MS>.
- Yang, L., Fan, M., Wang, X., Liu, Z., Li, N., Yu, W., and Wang, Y. (2025): Efficiency Comparison of Direct and Krylov Subspace Iterative Parallel Solvers for Hydraulic Fracture Propagation Simulations in Shale Reservoirs, Presented at the 59th U.S. Rock Mechanics/Geomechanics Symposium, D031S025R002. <https://doi.org/10.56952/ARMA-2025-0403>.
- Zhang, J., Braga-Neto, U., and Gildin, E. (2024): Physics-Informed Neural Networks for Multiphase Flow in Porous Media Considering Dual Shocks and Interphase Solubility, *Energy & Fuels*, 38(18), 17781–17795. <https://doi.org/10.1021/acs.energyfuels.4c02888>.
- Zhao, Y., Fukaya, T., Zhang, L., and Iwashita, T. (2022): Numerical Investigation into the Mixed Precision GMRES(m) Method Using FP64 and FP32, *Journal of Information Processing*, 30(0), 525–537. <https://doi.org/10.2197/ipsjip.30.525>
- Zubarev, D. I. (2009): Pros and Cons of Applying Proxy-Models as a Substitute for Full Reservoir Simulations, Presented at the SPE Annual Technical Conference and Exhibition, SPE-124815-MS. <https://doi.org/10.2118/124815-MS>.