
History-Matching of Imbibition Flow in Multiscale Fractured Porous Media Using Physics-Informed Neural Networks (PINNs)

Jassem Abbasi*, Pål Østebo Andersen & Aksel Hiorth

Department of Energy Resources, University of Stavanger
Stavanger, Rogaland, Norway

{jassem.abbasi, pal.andersen, aksel.hiorth}@uis.no

Ben Moseley

Seminar for Applied Mathematics, ETH Zürich,
Zürich, Switzerland

{benjamin.moseley}@ai.ethz.ch

Ameya D. Jagtap

Aerospace Engineering Department, Worcester Polytechnic Institute,
Worcester, MA 01609, USA

{ajagtap}@wpi.edu

Takeshi Kurotori & Anthony R. Kovscek

Department of Energy Science & Engineering, Stanford University
Stanford, CA, USA

{kurotori, kovscek}@stanford.edu

Abstract

We present a workflow based on physics-informed neural networks (PINNs) to model multiphase flow in fractured porous media. After validating the method on a synthetic problem, it was applied to an experimental dataset where brine was injected into a CO₂-saturated fractured shale core, with the fractures and fluid distribution imaged using CT scans. A domain decomposition approach was used for matrix and fractures, with a multi-network architecture to separately compute the state variables of each domain. A novel hybrid pre-training strategy was applied to enable the model to capture the system's complex multiscale attributes. By history matching of multi-fidelity observations, flow parameters were determined, with multiple initializations performed to assess uncertainty and uniqueness. The workflow showed high precision in retrieving key flow characteristics and accounted for multiscale effects. Additionally, the accuracy and computational efficiency of the proposed approach were compared with an existing method, numerical simulation, demonstrating superiority by order(s) of magnitude in both aspects. To our knowledge, this is the first workflow to efficiently solve inverse modeling of multiphase flow in fractured porous media using noisy, multifidelity real-world data.

*Corresponding Author

1 Introduction

Interpretation of multiphase flow phenomena in fractured porous media is challenging, because of their intrinsic multiscale nature (1). Almost all natural porous media are to some degree fractured; however, an inaccurate description of their influence on flow phenomena can lead to substantial errors (2). Various numerical methods based on finite-element and finite-volume approaches have been recommended to solve the problem more efficiently (3; 4). However, specially for inverse problems, they typically face challenges such as high computational requirements and low convergence reliability (5). Therefore, there is still a demand for more accurate and efficient prediction methods.

In this work, we have implemented a multiscale workflow based on Physics-Informed Neural Networks (PINNs) (6) for the forward and inverse simulation of (3+1)D multiphase flow in fractured porous media. This PINNs-based approach is proposed for reliable and computationally efficient inverse modeling of multiphase flow in fractured porous media, suitable for history-matching noisy and multifidelity experimental measurements. The robustness of the computational framework was first demonstrated through validation using a synthetic benchmark problem, and the performance was compared against a finite-difference (FD) based numerical simulator, as a common approach in the community. Then we applied the workflow to model and interpret a complex experiment of water injection into CO₂-saturated fractured shale rock (7). We incorporated a large multi-fidelity observational dataset, including 3D in situ saturation data measured using a high-resolution computed tomography (CT) scan setup, in the process of inverse calculations.

Implications. Solving the inverse problem in this study is essential for supporting reliable decision-making in real-world geoscientific applications, such as geological carbon storage and hydrocarbon reservoir management. The results of the studied inverse problem can, for example, enable more accurate and informed field-scale subsurface flow simulations.

Hypothesis. The hypothesis being tested is that PINNs, combined with a multiscale domain decomposition approach, can effectively solve the inverse problem of multiphase flow in fractured porous media by capturing complex dynamics and leveraging multi-fidelity observations to reduce uncertainties, even in the presence of noisy data. Additionally, the study evaluates the impact of coupled matrix-fracture interactions on the inverse-calculated coefficients of multiphase flow.

2 Methodology

Experimental Data. The availability and fidelity of experimental observations depend on laboratory technologies and prior knowledge of system dynamics and geometry, though uncertainties can arise if this knowledge is incomplete. Basic data include recovery factor curves and pressure measurements, while advanced tools like dynamic CT-scan imaging can provide in-situ saturation data but may not always be accessible. This study is motivated by the need for a workflow that efficiently can evaluate and history-match an experimental dataset injecting brine into a CO₂-saturated naturally fractured shale rock core, as detailed in (7). The core was initially saturated with CO₂, then brine was injected. Dynamic CT images were captured at intervals, recording in-situ saturation, phase recovery. Different fractures were identified from the CT scan images (8). Considering various scenarios, we utilized either *phase recovery vs. time* or *in-situ saturation data* as observational inputs, in addition to pressure and injection rate constraints.

Physics. Due to the complexities in geometry, we decomposed the domain into two subdomains, the matrix and fracture systems. We refer to the low-permeability non-fractured domain as matrix (Ω_M), and the high-permeability fractured domain as fracture (Ω_F). The matrix domain occupies the majority of the pore volume while the fracture domain has much higher permeability than the matrix. A schematic of the geometry of the core and the fractures is visualized in Fig. 1a. The Darcy velocity (u_i) of each phase is determined by Darcy's law (9), $u_i = -\lambda_i \nabla p_i$, $\lambda_i = K k_{r_i} / \mu_i$, where λ_i is mobility, p_i is the phase pressure, K and k_{r_i} are absolute and relative permeabilities, respectively, and μ_i is the phase viscosity. The index i determines wetting (w), and non-wetting (nw) phases. The effects of gravity have been ignored. The capillary pressure function, $p_c = p_{nw} - p_w$, relates the phase pressures. k_r and p_c are functions of phase saturation, s_i , where $s_w + s_{nw} = 1$. The conservation law for the spatiotemporal mass transport of phases in the matrix and fracture domains

is defined as

$$\phi \partial_t (\rho_i s_i) = -\nabla (\rho_i u_i) + q_i^\perp, \quad (i = w, nw), \quad \in \Omega_M \quad (1)$$

$$\phi^F \partial_t (\rho_i s_i^F) = -\nabla^\parallel (\rho_i u_i^\parallel) - q_i^\perp, \quad (i = w, nw), \quad \in \Omega_F \quad (2)$$

where q_i^\perp is matrix-fracture mass flux term and is a function of the local pressure difference between the two media, and the matrix flow properties (2). The unknown functions in this system are the k_r and p_c functions, which for a capillary dominated system (the matrix), appear mainly in the capillary diffusion coefficient, i.e., $\Lambda = \mu_m \lambda_{nw} f_w (-dJ/ds_w)$ (in scaled form). μ_m is the geometric mean of the phase viscosities ($\sqrt{\mu_w \mu_{nw}}$), and f_w is defined as the wetting phase relative mobility. Also, p_c and J are related via Leverett scaling (10), $p_c = \sigma \sqrt{\phi/K} J$. Here, ϕ is porosity and σ is surface tension. To solve the above coupled equations, we propose a PINNs-based workflow based on a multiscale neural networks architecture, as explained below.

Model architecture and training. We defined neural networks (separately for matrix (\mathcal{N}_m) and fracture (\mathcal{N}_f) domains), to emulate the functional dependency between the independent and dependent variables

$$s_w, p_{nw} = \mathcal{N}^M(x, y, z, t, \theta, \eta) \in \Omega_M \quad (3)$$

$$s_w^f, p_{nw}^f = \mathcal{N}^F(x_f, y_f, z_f, t, \theta_f, \eta) \in \Omega_F \quad (4)$$

where, $[x, y, z]$ and $[x_f, y_f, z_f]$ are the spatial coordinates corresponding to the collocation points of matrix and fracture, respectively (Fig. 1b). θ and θ_f are the network trainable parameters. The goal is to solve an optimization problem where $\theta^{m,f}, \eta = \arg \min(\mathcal{L}_t)$. Here, η is a set of inverse parameters calculated during the inverse calculations. Also, \mathcal{L}_t is the total loss term, defined to minimize the errors in both the physical flow equations in the matrix and fracture, their underlying initial/boundary conditions, as well as the available observation data. The outlet of networks are the state variables of the system, that are the s_w , and p_{nw} . The total loss term is a weighted combination of matrix and fracture losses, as well as losses corresponding to observational data, i.e., $\mathcal{L}_t = \mathcal{L}_t^M + \mathcal{L}_t^F + \mathcal{L}_t^D$, where \mathcal{L}_t^M and \mathcal{L}_t^F are the total loss terms corresponding to the initial/boundary conditions, as well as the PDE residuals for matrix and fracture domains, respectively. Minimizing these terms ensures the PINNs solution respects the physical constraints within their domain. Furthermore, \mathcal{L}_t^D represents the total loss term corresponding to the errors in the predictions of PINNs compared to the observational data. The complete explanation of the PINNs implementation and the underlying loss terms is provided in Appendix B. To address the inverse problem, we adopted an ansatz approach for representing the k_r and p_c curves. This means a trainable vector was defined, where each element corresponded to a parameter within the chosen correlations (eqs. (19), (20) and (31)). In the NNs, we used the following sequence of operators: a normalization function, an encoder, a Fourier transformer, a latent multilayer perceptron (MLP) network, an inverse Fourier transformer, a decoder network, and a denormalization function (as shown in Fig. 1c). Fourier transformers were used to make the networks capable of capturing high-frequency or multiscale behaviour (11) in the frontal regions of the saturation profiles. The network was trained using an Adam optimizer, with resampling of collocation points, and incorporating a gradually decreasing learning rate, from 0.0003, to 0.0001.

Pre-training. In the training, we started the computations with a pre-training strategy, then shifted to fully-coupled modeling strategy (Fig. 1f). The pre-training helps stabilize computations by first solving for individual domains independently. The physical implication of the pre-training assumption is that the main matrix and fracture interaction mechanisms is capillary imbibition

Implementation. The workflow ensures efficient training by managing collocation points and dimensionless loss terms. Collocation points were selected separately for the matrix and fracture domains, leveraging system geometry and CT scan data, such as fracture locations. Boundary collocation points were placed to capture inlet, outlet, and no-flow boundaries, while temporal points were distributed based on a \sqrt{t} scheme to emphasize critical early-time imbibition effects. The loss function integrates PDE residuals, boundary conditions, and observational data through dynamically adjusted, dimensionless weights. These weights, normalized to ensure comparability across terms with differing magnitudes, prevent any single component—such as high-gradient PDE residuals or observational mismatches—from dominating the training. This normalization maintains numerical stability and allows for balanced optimization across all data fidelities and physical constraints. More information regarding the implementation of the workflow is provided in Appendices.

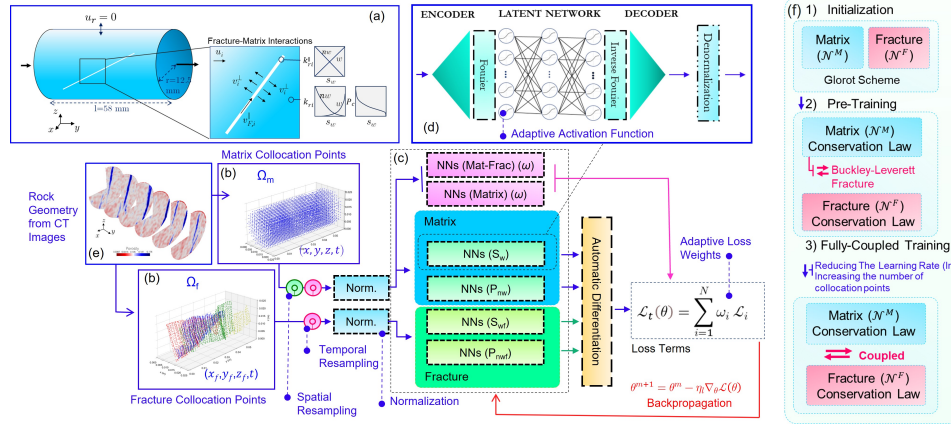


Figure 1: **A schematic review of the proposed PINNs-based computational workflow.** **a)** The simplified geometry of the matrix-fracture system and the impacting boundary conditions, **b)** The spatial distribution of the matrix and fracture collocation points, **c)** Separate networks for the state variables of matrix and fracture systems, **d)** The architecture of each NN, and **e)** the cross-sectional view of the porosity distribution in the experimental data, **f)** the proposed step-wise learning strategy.

Assumptions. Although the average permeability of the shale rock was known, an accurate measurement of the matrix permeability (K) was unavailable. The fracture permeability (K_F) was estimated from flow rates derived during history matching. Linear relative permeability curves and zero capillary pressure were assumed for fractures (12). These assumptions do not compromise the generality of the methodology or the effective flow properties. Furthermore, while demonstrated on a Brine- CO_2 system, the approach is adaptable to other incompressible fluids, with potential extensions to include phenomena like fluid compressibility, viscosity changes, varying rock/fracture properties and gravity effects.

3 Results

At the beginning, we validated the proposed workflow against a synthetic scenario of imbibition flow in a fractured porous media, with a vertical fracture parallel to the direction of the brine injection. Fig. 2 shows the results, in which the proposed workflow could successfully regenerate the phase and pressure distributions in porous media, compared to FD simulations (Fig. 2a). Also, it could greatly calculate the inverse parameters (Λ curve) by history matching of the observation data, as well as the total injection rate of brine (Fig. 2b), by history matching of the observations. For comparison, we also attempted to history-match the problem using a coupled FD and Nelder-Mead (NM) optimizer. In the case of inverse calculations, the workflow performed the calculations order(s) of magnitude faster and more accurately than FD-based simulation methods (see Fig. 2c).

Then, we applied the framework to interpret and analyse a multi-fidelity experimental dataset for brine injection in a CO_2 saturated fractured shale rock. The studied experimental problem is characterized by a confluence of complexities, e.g., significant permeability contrasts, unpatterned distribution of fractures, and highly unfavorable mobility ratios. The results demonstrated the effectiveness of the workflow in capturing the multiscale flow dynamics, as well as the phase and pressure distributions in the system (Fig. 3a). The implemented uncertainty quantification analysis yielded valuable insights into the unique and non-unique characteristics of the obtained solutions (Fig. 3b). Our comparison of inverse calculations with and without fractures demonstrates the pivotal role of multiscale phenomena in interpreting the experiments of imbibition in fractured porous media, as the presence of fractures introduces a hierarchy of scales in the system. These multiscale effects, while commonly overlooked in existing procedures, can significantly change the overall system behavior, influencing the scale of inverse calculated parameters such as K , k_r , and p_c relationships, as well as the interplay between different flow mechanisms in the system (Fig. 3c). Furthermore, we demonstrated the applicability of the proposed workflow for history-matching noisy high-fidelity observational data (dynamic high-resolution 3D CT images) and quantifying uncertainties in key system characteristics, such as fracture

geometry, as illustrated in Fig. 3d and e. Considering these demonstrations, the proposed approach offers valuable opportunities to identify and address uncertainties in the presumed mathematical models and system geometries, ensuring a more accurate representation of the physical system.

Limitations. Despite the superiority of the proposed approach in the examples studied, increasingly complex fracture distributions likely require even more complex neural networks architectures. This highlights the need for developing more flexible network architectures and optimisers capable of handling the significant non-linearities exist in such complex systems.

Summary. We propose a workflow based on PINNs for the modeling of (3+1)D two-phase flow in fractured porous media. The results demonstrate the accuracy and computational efficiency of the workflow, providing a reliable and efficient solution for inverse modeling of multiphase flow in fractured porous media, achieved through history matching of real-world experiments with noisy, multi-fidelity experimental measurements.

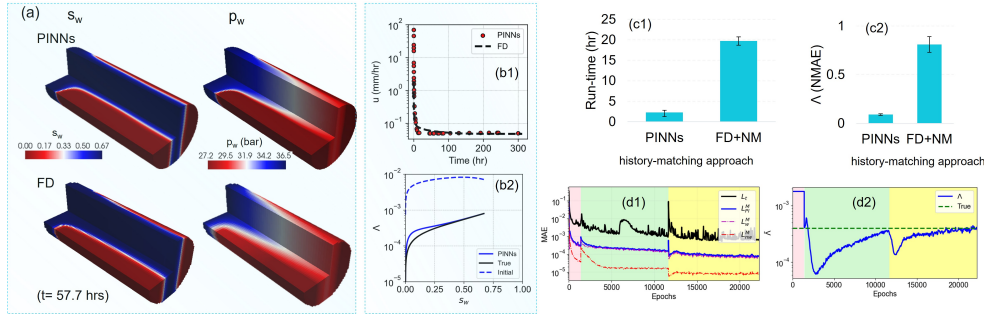


Figure 2: **The results of inverse calculation for the synthetic benchmark model, based on history matching of the RF curve vs. time.** **a)** the saturation and pressure profiles compared to FD simulation, **b)** The inverse calculated injection rate and capillary diffusion coefficient Λ , compared to the true values, **c)** TA comparison of the run-time and accuracy of two history-matching approaches in the estimation of the Λ curve: PINNs, and FD + Nelder-Mead (NM) optimizer, **d)** the training trend, 1) The total loss term, \mathcal{L}_t , and loss terms corresponding to the PDE residuals for matrix, i.e., \mathcal{L}_{PI}^M , \mathcal{L}_{PIw}^M , and \mathcal{L}_{PINw}^M , i.e., physics informed loss terms in matrix for w and nw phases, 2) $\bar{\Lambda}$ (the area under Λ).

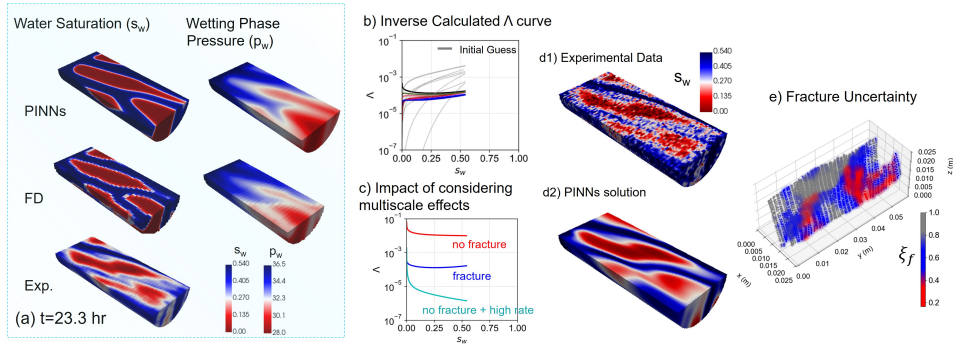


Figure 3: **The results for inverse calculation of matrix and fracture flow parameters by history matching of the real experimental data.** **a)** Comparing water saturation (s_w) and wetting phase pressure (p_w) distributions in the porous media at $t = 23.2$ hr, **b)** The results of inverse calculations for an ensemble of models with different initialization values. **c)** The impact of multiscale simulations on the inverse calculated flow parameters of porous media, **d)** The PINNs solution for inverse calculations when the in-situ water saturation data have been used as the observations **e)** The results of fracture uncertainty quantification; $\xi_f = 1$ (gray) indicates minimal uncertainty, $\xi_f = 0.6$ (blue) represents medium uncertainty, and $\xi_f = 0.2$ (red) represents high uncertainty.

References

- [1] L. S. Fung, “Simulation of Block-to-Block Processes in Naturally Fractured Reservoirs,” *SPE Reservoir Engineering*, vol. 6, no. 04, pp. 477–484, 11 1991. [Online]. Available: <https://dx.doi.org/10.2118/20019-PA>
- [2] I. Berre, F. Doster, and E. Keilegavlen, “Flow in Fractured Porous Media: A Review of Conceptual Models and Discretization Approaches,” *Transport in Porous Media*, vol. 130, no. 1, pp. 215–236, 10 2019. [Online]. Available: <https://link.springer.com/article/10.1007/s11242-018-1171-6>
- [3] M. Komijani, P. Wriggers, and T. Goudarzi, “An enriched mixed finite element model for the simulation of microseismic and acoustic emissions in fractured porous media with multi-phase flow and thermal coupling,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 47, no. 16, pp. 2968–3004, 11 2023. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/nag.3608><https://onlinelibrary.wiley.com/doi/abs/10.1002/nag.3608><https://onlinelibrary.wiley.com/doi/10.1002/nag.3608>
- [4] O. Møyner and K.-A. Lie, “A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,” *Journal of Computational Physics*, vol. 304, pp. 46–71, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999115006725>
- [5] X. Yang, H. Sun, Y. Yang, Y. Liu, and X. Li, “Recent progress in multi-scale modeling and simulation of flow and solute transport in porous media,” *Wiley Interdisciplinary Reviews: Water*, vol. 8, no. 6, p. e1561, 11 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/wat2.1561><https://onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1561><https://wires.onlinelibrary.wiley.com/doi/10.1002/wat2.1561>
- [6] 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, 2 2019.
- [7] T. Kurotori, M. P. Murugesu, C. Zahasky, B. Vega, J. L. Druhan, S. M. Benson, and A. R. Kovscek, “Mixed imbibition controls the advance of wetting fluid in multiscale geological media,” *Advances in Water Resources*, vol. 175, p. 104429, 5 2023.
- [8] M. P. Murugesu, B. Vega, C. M. Ross, T. Kurotori, J. L. Druhan, and A. R. Kovscek, “Coupled Transport, Reactivity, and Mechanics in Fractured Shale Caprocks,” *Water Resources Research*, vol. 60, no. 1, p. e2023WR035482, 1 2024. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1029/2023WR035482><https://onlinelibrary.wiley.com/doi/abs/10.1029/2023WR035482><https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023WR035482>
- [9] D. B. D. B. Das and S. M. Hassanizadeh, “Upscaling Multiphase Flow in Porous Media : From Pore to Core and Beyond,” 2005. [Online]. Available: <https://www.ebay.co.uk/itm/364905098551>
- [10] M. C. Leverett, J. Member, W. B. Lewist, and M. E. True, “Dimensional-model Studies of Oil-field Behavior,” *Transactions of the AIME*, vol. 146, no. 01, pp. 175–193, 12 1942. [Online]. Available: <https://dx.doi.org/10.2118/942175-G>
- [11] S. Wang, H. Wang, and P. Perdikaris, “On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113938, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045782521002759>
- [12] J. J. De La Porte, C. A. Kossack, and R. W. Zimmerman, “The Effect of Fracture Relative Permeabilities and Capillary Pressures on the Numerical Simulation of Naturally Fractured Reservoirs,” *Proceedings - SPE Annual Technical Conference and Exhibition*, pp. 191–199, 10 2005. [Online]. Available: <https://dx.doi.org/10.2118/95241-MS>

- [13] W. Anderson, “Wettability Literature Survey- Part 4: Effects of Wettability on Capillary Pressure,” *Journal of Petroleum Technology*, vol. 39, no. 10, pp. 1283–1300, 1987. [Online]. Available: <http://www.onepetro.org/doi/10.2118/15271-PA>
- [14] Y.-S. Wu, “Multiphase Flow in Fractured Porous Media,” *Multiphase Fluid Flow in Porous and Fractured Reservoirs*, pp. 207–250, 1 2016.
- [15] V. Martin, J. Jaffré, and J. E. Roberts, “Modeling Fractures and Barriers as Interfaces for Flow in Porous Media,” *SIAM Journal on Scientific Computing*, vol. 26, no. 5, pp. 1667–1691, 1 2005. [Online]. Available: <https://doi.org/10.1137/S1064827503429363>
- [16] A. Corey, “The interrelation between gas and oil relative permeabilities,” *Producers Monthly*, vol. 19, no. 1, pp. 38–41, 1954.
- [17] R. G. Bentsen and J. Anli, “Using Parameter Estimation Techniques To Convert Centrifuge Data Into a Capillary-Pressure Curve,” *Society of Petroleum Engineers Journal*, vol. 17, no. 01, pp. 57–64, 2 1977. [Online]. Available: <https://dx.doi.org/10.2118/5026-PA>
- [18] P. Andersen, “Early-and late-time prediction of counter-current spontaneous imbibition, scaling analysis and estimation of the capillary diffusion coefficient,” *Transport in Porous Media*, vol. 147, no. 3, pp. 573–604, 2023.
- [19] M. H. Krause and S. M. Benson, “Accurate determination of characteristic relative permeability curves,” *Advances in Water Resources*, vol. 83, pp. 376–388, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.advwatres.2015.07.009>
- [20] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” pp. 249–256, 3 2010. [Online]. Available: <https://proceedings.mlr.press/v9/glorot10a.html>
- [21] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 548–26 560, 2021.
- [22] E. T. Chung, Y. Efendiev, T. Leung, and M. Vasilyeva, “Coupling of multiscale and multi-continuum approaches,” *GEM - International Journal on Geomathematics*, vol. 8, no. 1, pp. 9–41, 2 2017. [Online]. Available: <https://arxiv.org/abs/1702.07095v1>
- [23] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh, “Analyses of internal structures and defects in materials using physics-informed neural networks,” *Science Advances*, vol. 8, no. 7, p. 644, 2 2022. [Online]. Available: <https://www.science.org/doi/10.1126/sciadv.abk0644>
- [24] J. Abbasi and P. Andersen, “Application of Physics-Informed Neural Networks for Estimation of Saturation Functions from Countercurrent Spontaneous Imbibition Tests,” *SPE Journal*, vol. 1, pp. 1–20, 1 2024. [Online]. Available: <https://dx.doi.org/10.2118/218402-PA>
- [25] W. M. Haynes, *CRC handbook of chemistry and physics*. CRC press, 2016.
- [26] A. Fenghour, W. A. Wakeham, V. Vesovic, and W. A. Wakeham, “The Transport Properties of Carbon Dioxide,” *Carbon Dioxide Journal of Physical and Chemical Reference Data*, vol. 27, p. 13107, 1998.
- [27] M. Shiga, T. Morishita, and M. Sorai, “Interfacial tension of carbon dioxide - water under conditions of CO2 geological storage and enhanced geothermal systems: A molecular dynamics study on the effect of temperature,” *Fuel*, vol. 337, p. 127219, 4 2023.
- [28] M. Houben, R. van Geijn, M. van Essen, Z. Borneman, and K. Nijmeijer, “Supercritical CO2 permeation in glassy polyimide membranes,” *Journal of Membrane Science*, vol. 620, p. 118922, 2 2021.
- [29] J. Abbasi and P. Andersen, “Simulation and Prediction of Countercurrent Spontaneous Imbibition at Early and Late Time Using Physics-Informed Neural Networks,” *Energy & Fuels*, vol. 0, no. 0, 9 2023. [Online]. Available: <https://pubs.acs.org/doi/10.1021/acs.energyfuels.3c02271>

- [30] O. Fuks and H. A. Tchelepi, “LIMITATIONS OF PHYSICS INFORMED MACHINE LEARNING FOR NONLINEAR TWO-PHASE TRANSPORT IN POROUS MEDIA,” Tech. Rep. 1, 2020. [Online]. Available: www.begellhouse.com
- [31] Z. Xiang, W. Peng, X. Liu, and W. Yao, “Self-adaptive loss balanced Physics-informed neural networks,” *Neurocomputing*, vol. 496, pp. 11–34, 7 2022.
- [32] S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopoulos, and G. E. Karniadakis, “Residual-based attention and connection to information bottleneck theory in PINNs,” 2023.
- [33] G. Coulaud and R. Duvigneau, “Physics-Informed Neural Networks for Multiphysics Coupling: Application to Conjugate Heat Transfer,” 2023. [Online]. Available: <https://inria.hal.science/hal-04225990https://inria.hal.science/hal-04225990/document>
- [34] S. Fridovich-Keil, R. Gontijo-Lopes, and R. Roelofs, “Spectral Bias in Practice: The Role of Function Frequency in Generalization,” *Advances in Neural Information Processing Systems*, vol. 35, 10 2021. [Online]. Available: <https://arxiv.org/abs/2110.02424v4>
- [35] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, “An Expert’s Guide to Training Physics-informed Neural Networks,” 8 2023. [Online]. Available: <https://arxiv.org/abs/2308.08468v1>
- [36] S. Buckley and M. Leverett, “Mechanism of Fluid Displacement in Sands,” *Transactions of the AIME*, vol. 146, no. 01, pp. 107–116, 1942. [Online]. Available: <http://www.onepetro.org/doi/10.2118/942107-G>
- [37] L. W. Lake, “Enhanced oil recovery,” 1989.
- [38] H. J. Welge, “A simplified method for computing oil recovery by gas or water drive,” *Journal of Petroleum Technology*, vol. 4, no. 04, pp. 91–98, 1952.
- [39] A. Lohne, O. Nødland, A. Stavland, and A. Hiorth, “A model for non-Newtonian flow in porous media at different flow regimes,” *Comput Geosci*, vol. 21, pp. 1289–1312, 2017.
- [40] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1 1965. [Online]. Available: <https://dx.doi.org/10.1093/comjnl/7.4.308>
- [41] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G. L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature Methods* 2020 17:3, vol. 17, no. 3, pp. 261–272, 2 2020. [Online]. Available: <https://www.nature.com/articles/s41592-019-0686-2>

Appendices

A Methods

A.1 Mathematics of Flow in Fractured Porous Media

We approach a problem of 3D multiphase flow in a fractured porous media at core scale. Our focus is specifically on interpreting data from a cylindrical core-plug (a shale rock taken from the Wolfcamp formation) (7). We refer to the low-permeability non-fractured zone as a matrix, and the high-permeability fractured zone as a fracture. The model is inferred from the definition that the matrix domain (Ω_M) occupies the majority of the pore volume and that fractures domain (Ω_F) have a much higher permeability than the matrix. The geometric structures of the core and the fractures are given in (7). A simplified schematic of the sample set-up is visualized in Fig. 1a. The core is placed horizontally, and the fractures are distributed in the core with different orientations. The radial surface of the core at r_c is closed (no-flow boundary). The inlet and outlet faces are at $y = 0$ and $y = L$, respectively. Water and CO₂ are considered to be the wetting (w) and non-wetting (nw) phases, respectively. In line with the experimental conditions, we neglected both the compressibility and the solubility of the phases. The system is initially saturated with the nw phase. The definition of the system implies that the flow phenomena are governed by mixed-imbibition mechanisms, where the fracture flow is driven by forced imbibition, and the matrix flow is governed by spontaneous imbibition between the fracture(s) and the matrix. Forced imbibition happens when an external force drives the wetting phase fluid through the porous media, overcoming any resistance from the medium and the non-wetting phase. Spontaneous imbibition, on the other hand, is the process in which the wetting phase fluid flows into a porous media via capillary forces without the need for external pressure (13). In the following, the mathematics of flow in matrix and fracture domains are discussed.

Matrix flow. The Darcy flow velocity (u_i) of each phase in the matrix is determined by Darcy's law

$$u_i = -\lambda_i \nabla p_i, \quad \lambda_i = \frac{K k_{ri}}{\mu_i}, \quad (i = w, nw) \quad (5)$$

where λ_i is mobility, p_i is the phase pressure, K and k_{ri} are absolute permeability and relative permeability, respectively, and μ_i is the phase viscosity. The index i determines (w), and (nw) phases. The effects of gravity have been ignored. The conservation law for the spatiotemporal mass transportation of phases in a matrix zone is defined as

$$\phi \partial_t (\rho_i s_i) = -\nabla (\rho_i u_i), \quad (i = w, nw), \quad \in \Omega_M \quad (6)$$

Given that $s_{wc} = 0$, we define the initial conditions as

$$\begin{cases} s_w(x, y, z, t = 0) = 0 \\ p_{nw}(x, y, z, t = 0) = p_i \end{cases}, \quad \in \Omega_M \quad (7)$$

At the inlet face ($y = 0$), the wetting phase is injected under a constant pressure p_{in} over time. At this face, it is assumed the wetting phase takes the highest possible mobility

$$\begin{cases} s_w(x, y = 0, z, t) = 1 - s_{nw} \\ p_w(x, y = 0, z, t) = p_{in} \end{cases} \quad (8)$$

The production from the outlet face ($y = l$) occurs at a constant pressure, p_{out}

$$p_{nw}(x, y = L, z, t) = p_{out} \quad (9)$$

At the outlet face of the core, no explicit constraints are imposed on the saturation of the fluid phases. The radial faces ($r = r_c$) are sealed from flow in radial directions

$$\frac{\partial p}{\partial r} \Big|_{r=r_c} \approx \left| \frac{\partial p}{\partial x} \right|_{r=r_c} + \left| \frac{\partial p}{\partial z} \right|_{r=r_c} = 0 \quad (10)$$

The assumed matrix properties state that the flow in the matrix is governed by viscous and capillary forces (gravity effects are assumed to be negligible).

Fracture flow. A fracture is typically characterized as a narrow, and high permeability zone enclosed by a less permeability matrix zone. This contrast in properties leads to highly heterogeneous flow behavior across the scales. The aperture between the fracture surfaces is generally small and is normally occupied by filling materials, that affects its effective porosity and permeability values (14).

Although fractures are discontinuities in the porous medium, they are considered as a separate porous medium domain (Ω_F). This assumption enables us to apply Darcy's law. Consequently, we assume fractures to be high permeability porous zones. This makes the equations in the two media consistent, eliminating the need for unnecessary complexities in the calculations (15). So, the average flux of each flowing phase in a fracture is written by modification of Darcy's law in eq.(5) as

$$v_i^{\parallel} = -\lambda_i^{\parallel} \nabla^{\parallel} p_i, \quad \lambda_i^{\parallel} = \frac{K^F k_{ri}^{\parallel}}{\mu_i}, \quad (i = w, nw). \quad (11)$$

In this equation, K_F is the fracture permeability, and k_{ri}^{\parallel} is the relative permeability of the fluids in the fracture. The value of the K_F is dependant on the fracture aperture, and also the extent of the filling materials in the fracture.

Matrix-fracture interactions. The accurate treatment of multiphase flow in a fractured porous media depends on the modeling of the dynamic matrix-fracture coupled interactions. Utilizing Darcy's law (2), the matrix-fracture flux is approximated by

$$v_i^{\perp} = -2K^{\perp} \lambda_i^{\perp} \left[\frac{p_{M,i} - p_{F,i}}{e_V} \right], \quad \lambda_i^{\perp} \simeq \frac{k_{ri}}{\mu_i}, \quad (i = w, nw), \quad (12)$$

where K^{\perp} refers to the matrix-fracture permeability normal to the fracture surface, and e_V is the fracture average aperture. As an ideal case, if one assumes a fracture as an open space between two parallel surfaces, then the aperture is called the hydraulic aperture, e_h . In practice, the value of e_V is much larger than e_h , i.e., $e_V \gg e_h$, as the fractures are normally filled with materials. The value of K_F is theoretically correlated to e_V . In this study, the fracture aperture, e_V , is assumed to be 0.001 m for all the fractures, and kept constant in all the evaluations. The cumulative value $V_i^{\perp}(t)$ of a rate $v_i^{\perp}(t)$ over time is given by the following integral

$$V_i^{\perp}(t) = \int_0^t v_i^{\perp}(\tau) d\tau, \quad (i = w, nw). \quad (13)$$

We can rewrite eq.(12) to obtain the mass transfer rate [$\text{kg/m}^3/\text{s}$] between the two media as

$$q_i^{\perp} = \frac{2\rho_i}{e_V} v_i^{\perp}, \quad (i = w, nw). \quad (14)$$

The factor 2 denotes the flow across both sides of fracture to matrix, assuming the equal flow characteristics across the two surfaces of the fracture. Then, the cumulative injected mass [kg/m^3] is

$$Q_i^{\perp}(t) = \int_0^t q_i^{\perp}(\tau) d\tau, \quad (i = w, nw). \quad (15)$$

The eq.of mass conservation in the fracture is the same as in the matrix, with an additional term representing the mass from the fracture to the matrix q_i^{\perp} . So, by modifying the matrix mass balance equation (eq. (6)), we define the mass balance in the fracture system as follows

$$\phi^F \partial_t (\rho_i s_i^F) = -\nabla^{\parallel} (\rho_i u_i^{\parallel}) - q_i^{\perp}, \quad (i = w, nw), \quad \in \Omega_F \quad (16)$$

At the contact point of the matrix and fractures, the continuity equation should also be solved in the matrix domain. So, we may rewrite eq. (6) as

$$\phi \partial_t (\rho_i s_i) = -\nabla (\rho_i u_i) + q_i^{\perp}, \quad (i = w, nw), \quad \in \Omega_M \quad (17)$$

The fraction of the pore volume of the matrix [dimensionless] that has been transferred between the matrix and fracture can also be calculated using the following equation

$$V_i^{\perp}(t) = \frac{Q_i^{\perp}(t)}{\phi \rho_i}, \quad (i = w, nw), \quad (18)$$

Closure equations. Capillary pressure and relative permeability curves are crucial input to model multiphase flow in porous media, e.g., in eqs. (6), (16) and (17). These curves are normally defined using theoretical correlations, with a few matching parameters. In this work, we apply an extended version of the Corey function (16) to model relative permeability curves

$$k_{rw}(S_w) = k_{rw}^{max}(S_w)^{n_w}; \quad n_w = n_{w1}S_w + n_{w2}(1 - S_w) \quad (19)$$

$$k_{rnw}(S_w) = k_{rnw}^{max}(1 - S_w)^{n_{nw}}; \quad n_{nw} = n_{nw1}S_w + n_{nw2}(1 - S_w) \quad (20)$$

In the extended version, n_w and n_{nw} are defined as saturation exponents that are weighted linearly with saturation S_w . S_w is the normalized water saturation and is calculated as

$$S_w = \frac{s_w - s_{wc}}{1 - s_{nwr} - s_{wc}}; \quad (21)$$

The capillary pressure curve $P_c(S_w)$ [pa] is calculated via Leverett scaling (10)

$$P_c = \sigma \sqrt{\frac{\phi}{K}} J \quad (22)$$

In this context, $J(S_w)$ represents the Leverett J-function, a dimensionless function that describes the shape of the capillary pressure curve. We define J-function via the modified (17) correlation

$$J(S_w) = -J_1 \ln\left(\frac{S_w}{S_{eq}}\right) + J_2 \ln\left(\frac{1 - S_w}{1 - S_{eq}}\right) \quad (23)$$

The equation is hard constrained in a way that that $J(S_{eq}) = 0$ and that dJ/dS_w becomes infinite at endpoints.

However, previous research by (18) demonstrated that capillary-dominated two-phase flow in porous media (spontaneous imbibition) can be uniquely characterized by the capillary diffusion coefficient (CDC) curve. This curve represents a mixed combination of flow properties of both phases

$$\Lambda = \mu_m \lambda_{nw} f_w \left(-\frac{dJ}{dS_w} \right) \quad (24)$$

In this equation, μ_m is the geometric mean of the phase viscosities ($\sqrt{\mu_w \mu_{nw}}$). The CDC curve, as well as the dynamics of multiphase flow in porous media is mainly controlled by the relative permeability of the phases and the capillary pressure curves. Measurement of these curves is challenging and typically requires indirect methods, such as history matching using numerical simulations (19). However, history matching in complex geometries, such as fractured porous media, is a challenging task that involves significant computational complexities. The challenge becomes more serious when high-fidelity datasets, such as CT-scan images, must be matched. In this work, we propose the application of PINNs as a reliable framework for history matching and characterisation of such problems.

Known and unknown parameters. To characterize fully the discussed system of equations, it is crucial to determine and understand the properties of both the matrix and the fractures. The determination of fluids properties, or average porosity and permeability of rock can be done using simpler methods. However, determination of the multiphase flow properties of the system - i.e., the relative permeability (k_r) and capillary pressure (p_c) of the matrix - is essential for accurately understanding the governing mechanisms, and this primarily requires multiphase experimental core flooding tests, in which the cylindrical rock is first saturated by a phase, then injected by the second phase at controlled pressures and flow rates. The k_r curve can be fully addressed via six parameters related to the Corey model (eqs. (19) and (20)): k_{rw}^{max} , k_{rnw}^{max} , n_{w1} and n_{w2} , n_{nw1} and n_{nw2} . Also, the p_c curve can be characterized via J_1 and J_2 parameters, based on eq. (23).

Due to the small pore volume of the fractures, we can reasonably assume a constant fracture porosity. The fracture permeability (K_F), which significantly impacts the overall flow rate during the experiments, needs to be calculated. Additionally, for simplicity, we assumed linear relative permeability curves for fractures (12).

A.2 Methodology

Physics-Informed Neural Networks (PINNs) were introduced by (6) as an efficient method for solving differential equations by combining neural networks with physics-based principles. In this study, we defined neural networks (separately for matrix (\mathcal{N}_m) and fracture (\mathcal{N}_f) domains), to emulate the functional dependency between the independent and dependent variables

$$s_w, p_{nw} = \mathcal{N}^M(x, y, z, t, \theta, \eta) \quad (25)$$

$$s_w^f, p_{nw}^f = \mathcal{N}^F(x_f, y_f, z_f, t, \theta_f, \eta) \quad (26)$$

where, $[x, y, z]$ and $[x_f, y_f, z_f]$ are the spatial coordinates corresponding to the collocation points of matrix and fracture, respectively. θ and θ_f are the network trainable parameters. The goal is to solve an optimization problem where

$$\theta^{m,f}, \eta = \arg \min(\mathcal{L}_t) \quad (27)$$

where η is a set of inverse parameters calculated during the inverse calculations. Also, \mathcal{L}_t is the total loss term, defined to minimize the errors in both the physical flow equations in the matrix and fracture, as well as the available observation data. The outlet of networks are the state variables of the system, that are the s_w , and p_{nw} . The total loss term is a weighted combination of matrix and fracture losses, as well as losses corresponding to observational data

$$\mathcal{L}_t = \mathcal{L}_t^M + \mathcal{L}_t^F + \mathcal{L}_t^D \quad (28)$$

where \mathcal{L}_t^M and \mathcal{L}_t^F are the total loss terms corresponding to the initial/boundary conditions, as well as the PDE residuals for matrix and fracture domains, respectively. Minimizing these terms ensures the PINNs solution respects the physical constraints within their domain. Furthermore, \mathcal{L}_t^D represents the total loss term corresponding to the errors in the predictions of PINNs compared to the observational data. The complete explanation of the PINNs implementation and the underlying loss terms is provided in Appendix B.

To address the inverse problem, we adopted an ansatz approach for representing the relative permeability and capillary pressure curves. This means a trainable vector was defined, where each element corresponded to a parameter within the chosen correlations for these curves (equations (19), (20) and (31)). More information regarding the initialization and definition of the inverse variables is provided in Appendix D.

A.3 Model Architecture

The architecture of the applied PINNs model is shown in Fig. 1. Due to the complexities in the geometry, and non-linearities in flow profiles, we considered the matrix and fracture systems as decomposed domains. Then, separate networks were allocated to each domain, to increase the flexibility of the networks in capturing the high-frequency trends in the solution of equations. Also, for each domain, the system state variables, i.e., pressure (p_{nw}) and water saturation (s_w) had separate networks (Fig. 1b). Before passing the data to the network, the input data was normalised to the approximate range of (-1,1). The (de)normalization technique is discussed in Appendix D. In the NNs corresponding to the S_w values (for both matrix and fracture), we used the following sequence of operators: an encoder, a Fourier transformer, a latent multilayer perceptron (MLP) network, an inverse Fourier transformer, and a decoder network (as shown in Fig. 1c). The outlet of the NNs is then denormalized to the range of actual values. The Fourier transformers were used to make the networks able to capture the high-frequency or multiscale behavior (11) in the frontal regions of the saturation profiles. The NNs for the calculation of p_{nw} were similar, except that the Fourier and inverse Fourier transformations were not applied. The MLP network, which had a depth of five and a width of 80 neurons, was activated by an adaptive \tanh activation function. The model was initialized using the Glorot initialization scheme (20). Separate MLP networks were used to address the self-adaptive local weighting of the errors in PDE residuals for matrix and fracture domains. See Appendices B and D for more details.

Training: The network was trained utilizing an Adam optimizer employing a full-batch approach and incorporating a gradually decreasing learning rate, from 0.0003, to 0.0001. We have applied a weight decay value of 0.0001. The number of gradient descent training steps (epochs) depended on the problem's complexity. Typically, 15,000 to 25,000 epochs were needed for stable error values.

Table 1: The network properties for the synthetic benchmark model

Property	Value	Property	Value
\mathcal{N}^M Width	80	\mathcal{N}^M Depth	8
\mathcal{N}^F Width	60	\mathcal{N}^F Depth	6
Activation Function	Adaptive <i>tanh</i>	Optimizer	Adam
Learning Rate (lr)	2e-4	Weight Decay	1e-4
Batch size	36000	Fourier Transform	Active for s_w

A.4 The Solution Approach

The investigated system of equations governing the matrix and fracture domains exhibits a complex interdependency. In this context, training the model using a fully-coupled approach generates chaotic computational behavior as the interaction terms between the matrix and fracture domains are highly sensitive to the accuracy of the calculations performed in each domain individually. On the other hand, the invasion of fluids from the fracture into the matrix domain is a strong function of the values of the interaction terms. Crucially, the differing loss landscapes of the matrix and fracture PINNs models result in distinct convergence behaviors. To mitigate these issues, alternative preconditioning techniques are necessary (21). In this work, we suggested to apply a sequence of pre-training training steps before performing the fully-coupled simulations. The details of the applied approach is visualized in Fig. 1e. In this process, at first, the matrix and fracture networks are trained individually, without utilizing any coupled inter dependencies. Then, we shift to the fully-coupled training technique.

However, the utilisation of the pre-training technique, although considers many of the possible flow mechanisms, neglects the impact of mechanisms such as viscous flow interactions between the matrix and fracture. The difference gets more critical when the permeability and capillary variation in matrix and fracture media is less significant. The mentioned mechanisms are naturally considered in the fully coupled approach. So, after pre-training of the networks, and reaching an equilibrium in the training process, we switch to the fully coupled technique. The mathematics behind coupling the matrix and fracture is previously explained in section A.1.

Pre-training: This stage leverages the model reduction technique (22) as a means to tackle multiscale problems. To do so, we applied an independent training strategy for matrix and fracture at the beginning of training, before shifting to the fully-coupled training strategy. In this approach, instead of defining the matrix-fracture transfer to modify the PDEs of flow, we explicitly applied their impacts as the boundary conditions in the matrix as follows.

- Matrix: The multiphase flow in the matrix is analyzed by assuming that the s_w and the p_{nw} in the matrix at points of intersection with fractures are equal to the corresponding values in the fractures

$$s_w = s_w^{F,BL}, \quad \in \Omega_{MF} \quad (29)$$

$$p_{nw} = p_{nw}^{F,BL}, \quad \in \Omega_{MF} \quad (30)$$

$s_w^{F,BL}$ is calculated using the 1D Buckley-Leverett (BL) theory, and $p_{nw}^{F,BL}$ is calculated using linear pressure distribution assumption. The Buckley-Leverett theory offers a fundamental analytical approach for forecasting the saturation profile and front velocity of the displacing fluid in two-phase flow in porous media. The BL theory is explained in Appendix E.

- Fracture: The PINNs model corresponding to the PDE of multiphase flow in fractures is solved by ignoring the interactions with matrix.

In the case of inverse calculations, before starting the optimization of inverse variables, it is essential to partially train the networks so that they can learn the main characteristics of the problem. This approach has been successfully applied in previous studies, such as those by (23) and (24). At this step, the trainable inverse parameters are kept frozen for some limited epochs.

Table 2: The network properties applied for the experimental problem.

Property	Value	Property	Value
\mathcal{N}^M Width	80	\mathcal{N}^M Depth	8
\mathcal{N}^F Width	60	\mathcal{N}^F Depth	6
Activation Function	Adaptive <i>tanh</i>	Optimizer	Adam
Learning Rate (lr)	0.10	Weight Decay	1e-4
Batch size	46000	Fourier Transform	Active for s_w
Resampling Sequence	10	Temporal Sampling	Cartesian

A.5 Experimental Data

This study is motivated by the need for a workflow that efficiently can evaluate and history-match an experimental dataset injecting brine into a CO₂-saturated naturally fractured shale rock core, as detailed in (7). Shales are considered tight media with very low permeability and have only the recent decades been developed for hydrocarbon production due to advances in hydraulic fracturing and drilling technology. The affinity of shale to adsorb CO₂ (e.g. over CH₄, shale gas) has also resulted in opportunities to sequester carbon in shales. The examined cylindrical Wolfcamp shale core had a length of 5.8 cm, diameters of 2.5 cm, an average permeability of $1.97e - 17 \text{ m}^2$ (0.020 mD) and porosity of 0.102. The reported permeability represents the average permeability of both the matrix and fractures, with the matrix permeability expected to be significantly lower. The core was initially saturated with CO₂, then brine was injected at fixed pressure drop (530 psi injection pressure and 460 psi back-pressure) for 311 hours while confined at 700 psi. Dynamic computed tomography (CT) images were captured at intervals, recording in-situ saturation, phase recovery, and injected fluid volume over time. Different fractures were identified from the CT scan images (8). History matching and inverse calculations in this study relied on several key experimental measurements:

- **Recovery Factor (RF):** The volume fraction of the pore space displaced by the injected fluid over time. We calculated the RF curve by averaging the saturation of the wetting phase fluid obtained from CT scan images.
- **Boundary Conditions:** The constant pressures at the inlet and outlet boundaries were known based on the experimental setup yielding a fixed pressure drop over the core.
- **Injection Volume:** The cumulative volume of water injected as function of time was measured.
- **Spatial Saturation Data:** Spatial distributions of fluid saturation within the core sample at different times, obtained through CT scans (19 snapshots).

Regarding the fluid properties, by considering the laboratory conditions, the viscosities of water and CO₂ are assumed to be 0.89 cP (25) and 0.0157 cP (26), respectively, with their interfacial tension (IFT) assumed to be 0.04 *N/m*. (27). Water and CO₂ density also were set to be 998.7 and 78.9 *kg/m³*, respectively (28).

A.6 Collocation Points

As it is visualized in Fig. 1, collocation points were extracted separately for the matrix and fracture domains based on the CT/micro-CT scan coordinates. 23500 matrix spatial collocation points were used in the core cylinder domain at specified x, y, z resolutions, excluding points in superposition with fractures. Boundary collocation points were collected for the cylinder ends and radius. Fracture collocation points were manually extracted from micro-CT scans. Temporal collocation points were randomly selected based on the \sqrt{t} distribution intervals, as it is expected that the system is mainly controlled by spontaneous imbibition mechanism. (29) have shown how the selection of temporal points based on \sqrt{t} distribution intervals helps in improving PINNs solutions. More detailed information about the experimental data and the collocation points are provided in Appendix C.

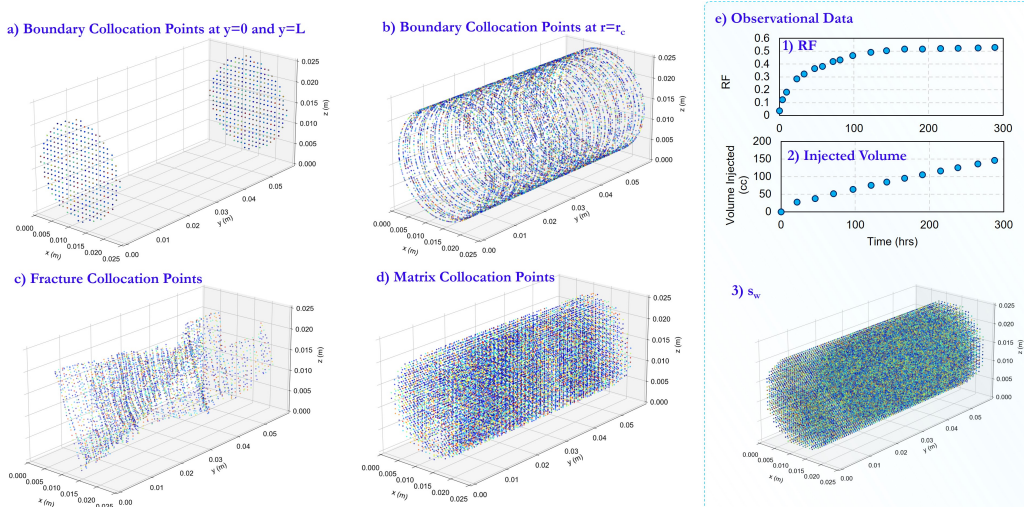


Figure 1: **The collocation points have been used for solving the problem under study.** The color of the points demonstrate the corresponding time for each point in the range $t = 1 - 10^6$ sec. **a)** The collocation points at the boundary conditions of inlet ($y=0$) and outlet ($y=L$) faces (940 spatial points), **b)** The collocation points corresponding to the non-flowing boundary condition at $r = r_c$ (8630 spatial points), **c)** The collocation points corresponding to the fracture (4103 spatial points), **d)** The collocation points corresponding to the matrix (23500 spatial points). The matrix collocation points with distance less than 0.0006 m from the fracture collocation points have been removed. **e)** the observational data points used in the data loss: 1) RF points, 2) points with the measured injected volume, and 3) spatiotemporal points of the in-situ s_w measured using CT-scan.

B Physics-Informed Neural Networks: Implementation

Physics-Informed Neural Networks (PINNs) is an approach that is introduced by (6) for more efficient solving of differential equations, by combining neural networks with physics-based principles. In this study, let NNs \mathcal{N}_m and \mathcal{N}_f emulate the functional dependency between the independent and dependent variables

$$s_w, p_{nw} = \mathcal{N}^M(x, y, z, t, \theta, \eta) \quad (31)$$

$$s_w^f, p_{nw}^f = \mathcal{N}^F(x_f, y_f, z_f, t, \theta_f, \eta) \quad (32)$$

The goal is to solve an optimization problem in which

$$\theta^{m,f}, \eta = \operatorname{argmin}(\mathcal{L}_t) \quad (33)$$

In this equation, \mathcal{L}_t represents the weighted summation of all of the loss terms corresponding to matrix and fracture, as well as the observational data

$$\mathcal{L}_t = \mathcal{L}_t^M + \mathcal{L}_t^F + \mathcal{L}_t^D \quad (34)$$

where \mathcal{L}_t^M and \mathcal{L}_t^F are the total loss terms corresponding to the initial/boundary conditions, as well as the PDE residuals for matrix and fracture domains, respectively. Furthermore, \mathcal{L}_t^D represents the total loss term corresponding to the errors compared to the observational data values. We will now explore the specific formulations of each loss term in more detail.

As the scales of pressure data and water saturation data differ significantly, we needed to ensure they are comparable by bringing them to similar scales. To achieve this, we normalized the pressure loss terms by dividing all related terms by κ_p , defined as

$$\kappa_p = (p_{in} + p_{out})/2 \quad (35)$$

Furthermore, since the scale of loss terms related to PDE residuals also deviates significantly from the [0,1] range, we defined another PDE normalization multiplier as

$$\kappa_r = \frac{t_{max}}{(\rho_w + \rho_{nw})/2} \quad (36)$$

where t_{max} is the time related to the end of experiment, which is almost 1e6 sec (277.78 hrs).

B.1 Flow in Matrix Domain

The solution for the governing equations within the matrix domain is achieved by minimizing a weighted combination of loss terms, denoted as \mathcal{L}_t^M . These loss terms account for the residuals of the PDEs, the initial conditions, and the boundary conditions applied at different boundaries.

PDE residuals: To solve the flow in the matrix, we must simultaneously address the conservation laws for both the wetting and non-wetting phases. Thus, we define different loss terms according to equation (17)

$$\mathcal{L}_w^M = \frac{1}{\kappa_r} \text{MAE}(\phi \partial_t (\rho_w s_w) + \nabla (\rho_w u_w), 0.0), \in \Omega_M \quad (37)$$

$$\mathcal{L}_{nw}^M = \frac{1}{\kappa_r} \text{MAE}(\phi \partial_t (\rho_{nw} s_{nw}) + \nabla (\rho_w u_{nw}), 0.0), \in \Omega_M \quad (38)$$

Here, MAE is the Mean Absolute Error, a metric used to measure the errors. It represents the average absolute difference between the predicted values and the actual values:

$$\text{MAE}(\hat{y}_i, y_i) = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (39)$$

where n is the number of observations, y_i is the actual value, and \hat{y}_i is the predicted value. Then, the total loss corresponding to the residual of equations of flow in matrix is defined as

$$\mathcal{L}_{PI}^M = (\mathcal{L}_w^M + \mathcal{L}_{nw}^M) \cdot e^{\omega^M}, \in \Omega_M \quad (40)$$

where ω^M represents the local weight modifier obtained by self-attention mechanism as outlined in Appendix D. A loss term is defined to constraint the values of the self-adaptive modifier for

$$\mathcal{L}_\omega^M = \text{MAE}(\omega^M, 0.0), \in \Omega_F \quad (41)$$

Matrix-fracture mass transfer: For the matrix collocation points, where mass transfer occurs between the matrix and the superimposed fracture network, we formulate the residual term of the governing partial differential equation as follows

$$\mathcal{L}_w^{MF} = \frac{1}{\kappa_r} \text{MAE}(\phi \partial_t (\rho_w s_w) + \nabla (\rho_w u_w) - \frac{\rho_w}{e_V} (2v_w^\perp), 0.0), \in \Omega_{MF} \quad (42)$$

$$\mathcal{L}_{nw}^{MF} = \frac{1}{\kappa_r} \text{MAE}(\phi \partial_t (\rho_{nw} s_{nw}) + \nabla (\rho_w u_{nw}) - \frac{\rho_{nw}}{e_V} (2v_{nw}^\perp), 0.0), \in \Omega_{MF} \quad (43)$$

The interaction term is governed by the pressure differential between the matrix and fracture and the local transmissibility of the matrix. In the fully-coupled training stage, we then define

$$\mathcal{L}_{PI}^{MF} = (\mathcal{L}_w^{MF} + \mathcal{L}_{nw}^{MF}) \cdot e^{\omega^{MF}}, \in \Omega_{MF} \quad (44)$$

Here, ω^{MF} is defined as a local self-adaptive weight modifier. A loss term is defined to constraint the values of the self-adaptive modifier for

$$\mathcal{L}_\omega^{MF} = \text{MAE}(\omega^{MF}, 0.0), \in \Omega_{MF} \quad (45)$$

Pre-training stage: In the pre-training stage, we assumed that the matrix collocation points corresponding to the fractured zones are acting as the boundary conditions of the system, following the pressure and saturation trends of the fractures. The matrix and fracture points, however, was calculated via Buckley-Leverett technique. We directly assign the fracture saturation and pressure properties to the collocated matrix points

$$\mathcal{L}_{PT,s_w}^{MF} = \text{MAE}((s_w - s_w^F)\xi_f, 0.0), \quad \in \Omega_{MF} \quad (46)$$

$$\mathcal{L}_{PT,p_{nw}}^{MF} = \frac{1}{\kappa_p} \text{MAE}((p_{nw} - p_{nw}^F)\xi_f, 0.0), \quad \in \Omega_{MF} \quad (47)$$

where, s_w^F and p_{nw}^F are the fracture state variables calculated via Buckley-Leverett and linear pressure drop equation, explained in Appendix E. Also, ξ_f is defined as a self-adaptive multiplier acts as the fracture uncertainty identifier. Technically, it adjusts the local matrix-fracture interactions. ξ_f is a trainable vector defined for all fracture collocation points, and initialized to be equal to one. Then, a restricting loss terms is defined to balance the ξ_f impacts during the training

$$\mathcal{L}_{\xi_f}^{MF} = \text{MAE}(\xi_f, 1.0), \quad \in \Omega_{MF} \quad (48)$$

In trying to minimize the trade-offs between the loss terms comparing the PINNs predictions vs. CT-scan images, as well as the matrix-fracture physical constraints, the values of ξ_f is adjusted in a way that the lowest discrepancies was obtained. In scenarios where uncertainty quantification is not a focus, setting $\xi_f = 1.0$ for all points simplifies the optimization process.

Initial conditions: With the initial state of the system fully defined, we can now express the corresponding loss terms as

$$\mathcal{L}_{IC,s_w}^M = \text{MAE}(s_w(x, y, z, t = 0), s_{wc}), \quad \in \Omega_M \quad (49)$$

$$\mathcal{L}_{IC,p_{nw}}^M = \frac{1}{\kappa_p} \text{MAE}(p_{nw}(x, y, z, t = 0), p_i), \quad \in \Omega_M \quad (50)$$

Boundary conditions at inlet: At the inlet, the core is exposed to water under a constant injection pressure, allowing us to express the loss terms as

$$\mathcal{L}_{BC0,s_w}^M = \text{MAE}(s_w(x, y = 0, z, t), s_{w,max}), \quad \in \Omega_M \quad (51)$$

$$\mathcal{L}_{BC0,p_{nw}}^M = \frac{1}{\kappa_p} \text{MAE}(p_{nw}(x, y = 0, z, t), p_{in}), \quad \in \Omega_M \quad (52)$$

Boundary conditions at outlet: To ensure proper flow characteristics, the outlet should be modeled with a constant pressure boundary condition

$$\mathcal{L}_{BC1,p_{nw}}^M = \frac{1}{\kappa_p} \text{MAE}(p_{nw}(x, y = L, z, t), p_{out}), \quad \in \Omega_M \quad (53)$$

Boundary conditions at radial surfaces: The radial (r_c) boundary condition can be enforced by adding a term to the loss function that penalizes the deviation from zero of the sum of the partial derivatives with respect to x and z at $r = r_c$. The loss function term can be written as

$$\mathcal{L}_{BCr}^M = \frac{1}{\kappa_p} \text{MAE} \left(\left. \frac{\partial p}{\partial x} \right|_{r=r_c} + \left. \frac{\partial p}{\partial z} \right|_{r=r_c} \right), \quad \in \Omega_M \quad (54)$$

Pre-training: In the pretraining stage, we assumed that the matrix collocation points corresponding to the fractured zones are acting as the boundary conditions of the system, following the pressure and saturation trends of the fractures. The matrix and fracture points, however, was calculated via Buckley-Leverett technique. The

B.2 Flow in Fracture Domain

Same as the matrix domain, we define the fracture total loss term, \mathcal{L}_t^F .

PDE residuals: The conservation of mass in fracture media should also be solved in parallel with the matrix equations. By utilizing equation (16), we write

$$\mathcal{L}_w^F = \frac{1}{\kappa_r} \text{MAE}(\phi^{\parallel} \partial_t (\rho_w s_w^F) + \nabla^{\parallel} (\rho_w u_w^{\parallel}) + \frac{\rho_w}{e_V} (2v_w^{\perp})), \quad \in \Omega_F \quad (55)$$

$$\mathcal{L}_{nw}^F = \frac{1}{\kappa_r} \text{MAE}(\phi^{\parallel} \partial_t (\rho_{nw} s_{nw}^F) + \nabla^{\parallel} (\rho_{nw} u_{nw}^{\parallel}) + \frac{\rho_w}{e_V} (2v_{nw}^{\perp})), \quad \in \Omega_F \quad (56)$$

Our approach to address the reported limitations of PINNs in capturing shock fronts for fully-viscous problems (30) involved incorporating a small capillary pressure term. This term introduces a diffusive effect that mitigates the overly sharp transitions typically observed in the displacement front of the replacing fluid. Then, the total loss value for the residual of PDE in the fracture domain is calculated by

$$\mathcal{L}_{PI}^F = (\mathcal{L}_w^F + \mathcal{L}_{nw}^F) \cdot e^{\omega^F}, \quad \in \Omega_F \quad (57)$$

Here, e^{ω^F} is defined as a local self-adaptive weight modifier (see Appendix D). A loss term is defined to constrain the values of the self-adaptive modifier

$$\mathcal{L}_{\omega}^F = \text{MAE}(\omega^F, 0.0), \quad \in \Omega_F \quad (58)$$

Initial conditions: A similar initial condition is also applied to the fracture collocation points.

$$\mathcal{L}_{IC,s_w}^F = \text{MAE}(s_w^F(x, y, z, t = 0), s_{wc}), \quad \in \Omega_F \quad (59)$$

$$\mathcal{L}_{IC,p_{nw}}^F = \frac{1}{\kappa_p} \text{MAE}(p_{nw}^F(x, y, z, t = 0), p_i), \quad \in \Omega_F \quad (60)$$

Boundary conditions at inlet: For the fractures with collocation points at the inlet boundary, we may also define a similar set of boundary conditions

$$\mathcal{L}_{BC0F,s_w}^F = \text{MAE}(s_w^F(x, y = 0, z, t), s_{w,max}), \quad \in \Omega_F \quad (61)$$

$$\mathcal{L}_{BC0F,p_{nw}}^F = \frac{1}{\kappa_p} \text{MAE}(p_{nw}^F(x, y = 0, z, t), p_{in}), \quad \in \Omega_F \quad (62)$$

Boundary conditions at outlet: Similarly, for fractures with collocation points at the outlet boundary, we define a similar pressure boundary condition to satisfy the downstream pressure of the system in the fracture system.

$$\mathcal{L}_{BC1,p_{nw}}^F = \frac{1}{\kappa_p} \text{MAE}(p_{nw}^F(x, y = L, z, t), p_{out}), \quad \in \Omega_F \quad (63)$$

B.3 Observations

During the inverse calculations, ensuring the PINNs predictions closely align with experimental observations is crucial. To achieve this, we define a data loss term, denoted by \mathcal{L}_t^D , which represents the sum of one or more individual loss terms detailed below:

Recovery factor (RF): As matching the average recovered fluids at all times is time-consuming and computationally intensive, we matched the RF at one randomly chosen time point during each epoch. This approach helps the network learn from the observation data without imposing a high computational burden on the optimizer. So, consider t_r as the time corresponding to one of the available CT-scan images, selected randomly, we can define \bar{s}_w as:

$$RF = \frac{1}{N_x N_y N_z} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} s_w(x_i, y_j, z_k, t = t_r), \quad \in \Omega_M \quad (64)$$

In the above equation, it is assumed that the core is fully-saturated by the non-wetting phase at the beginning of the test. Then, the corresponding loss term will be:

$$\mathcal{L}_{RF} = \text{MAE}(RF, \bar{s}_w^{\text{obs}}(t = t_r)), \quad \in \Omega_M \quad (65)$$

Here, \bar{s}_w^{obs} is the observed \bar{s}_w at time t_r , obtained by averaging the water saturation from the CT-scan images.

In-situ water saturation: To incorporate the measured in situ saturation data, a loss term is defined that minimises errors in local saturation values:

$$\mathcal{L}_{s_w} = \text{MAE}(s_w(x, y, z, t), s_w^{\text{obs}}(x, y, z, t)), \quad \text{for } (x, y, z, t) \in \Omega_M \quad (66)$$

where s_w^{obs} is the measured in-situ water saturation data. Given access to approximately one million observation points in the spatiotemporal dimensions, we randomly selected a small fraction of them (25000 points) at each epoch. In this work, the selected points was resampled (randomly) at each epoch.

Inflow rate constraint: As an observational data, we may have access to the total inflow rate versus time. To match the inflow rate with the observational data, we write it as:

$$\begin{aligned} \mathcal{L}_{Q_{inj}} = & \text{MAE}(A_m \iint u_w M(y=0) dx dy + \dots \\ & A_f \iint u_w^F(y^F=0) dx^F dy^F, Q_{inj}^{\text{obs}}(t)), \quad \in (\Omega_M \cup \Omega_F) \end{aligned} \quad (67)$$

We denote the cross-sectional areas of the matrix and fracture as A_m and A_f , respectively. To estimate A_f , we sum the areas of all fracture collocation points at the $y = 0$ plane, excluding overlaps. The area of each individual fracture collocation point is calculated using $\pi \left(\frac{e_v}{2}\right)^2$, where e_v represents the fracture aperture. Finally, we determine A_m based on the core radius (r_c) and the calculated A_f using the formula $A_m = \pi(r_c)^2 - A_f$.

C Experimental Data/Collocation Points

C.1 Experimental Data

In this study, we evaluated an experimental dataset corresponding to the injection of brine into a fully CO₂-saturated fractured shale rock, as detailed in (7). The experiment was conducted using a core-flooding apparatus integrated with an X-ray μ CT imaging setup. Fig. 1e illustrates a map of the porosity distribution. In the test, a cylindrical Wolfcamp shale core with a mean permeability (K_{exp}) of $1.97 \times 10^{-17} \text{ m}^2$ (0.02 mD) and an average porosity (ϕ) of 0.102 was used. The reported permeability represents the average permeability of both the matrix and fractures, with the matrix permeability expected to be significantly lower. The viscosities of water (25) and CO₂ (26) are assumed to be 0.89 cP and 0.0157 cP, respectively, with their interfacial tension (IFT) assumed to be 0.04 N/m. (27). Water and CO₂ density also were set to be 998.7 and 78.9 kg/m³, respectively (28).

The core was initially saturated with CO₂, and subsequently, water (brine) was injected at room temperature with an injection pressure (p_{in}) of 530 psi and a backpressure (p_{out}) of 460 psi. Throughout the test, the core was confined at a pressure of 700 psi, ensuring no lateral flow out of the core. The experiment lasted 311 hours, during which dynamic CT images were captured at various intervals. The in-situ saturation data and phase recovery recorded during the injection process have been processed. In addition, the volume of the injected fluid (water) versus time has been recorded and utilised in the computations.

The collocation points corresponding to the matrix and fractures are extracted from the clinical CT and micro-CT images. Each snapshot of the CT-scan had dimensions of [120, 87, 120], with the voxel resolution of 0.024 mm³. We had access to 19 snapshots through the text from 0.1 to 311.405 hrs after starting of the injection. In total, we identified and extracted six fractures from the CT-scan observations, ensuring none of them intersected with each other. The spatial geometry of the fractures is shown in Fig. 1b. We neglected the impacts of micro-fractures however, although they were observed in the micro-CT scan images.

Collocation points: We need to collect collocation points separately for matrix and fracture domains. The spatial coordinates of the collocation points were extracted on the basis of the coordinates of the points in the CT-scan images. The spatial points have been chosen with the resolution of 30 points in x and z ($dx = dz = 0.81$ mm) dimensions, and 60 points in y dimension ($dy = 0.96$ mm). The collocation points close to the fracture collocation points have been removed from the list of spatial points. In total, 73500 spatial points have been used in the cylindrical domain of the core. Refer to Fig. 1 for a visualization of these collocation points. To address the system's boundary conditions, we've also collected the requisite boundary collocation points. Fig. 1 shows the collected collocation points for the boundary conditions at $y=0$, $y=L$ and $r=r_c$. The radial collocation points have been chosen via $x_r = r_c \cdot \cos(\beta)$ and $z_r = r_c \cdot \sin(\beta)$, where β represents the angular coordinate of the point, randomly generated within the range of $(0, 2\pi)$. The corresponding y values of the points was extracted from the main collocation points.

Fracture collocation points: The fracture collocation points were manually extracted from microCT scans, along with porosity and saturation images. Although some of the fractures could be viewed as separate smaller fractures, we merged the fractures that aligned with each other, reflecting their consistent behaviour in passing the fluids during the injection process.

Temporal points: The temporal values assigned to the collocation points were randomly selected and subsequently updated during the resampling process. Anticipating spontaneous imbibition as the dominant mechanism, we selected collocation points at equal intervals relative to their square root of time values, ensuring a balanced distribution across the temporal dimension. (29) have previously shown that the spontaneous imbibition occurs with a constant rate at the square root of time coordinate, and it is advantageous to distribute the collocation points accordingly during training of PINNs.

D PINNs Computational Strategies

Adaptive loss weights: In the training of PINNs, especially in complex processes, simultaneous minimization of various terms often requires addressing conflicting objectives. Numerous studies have demonstrated that employing adaptive weights for these loss terms can enhance the performance of PINNs (31). In this study, we applied these techniques by employing self-adaptive weight values for both the loss terms and the calculated PDE residuals at different domains: matrix (Ω_M), fracture (Ω_F), and matrix-fracture (Ω_{MF}). This adaptive weighting strategy enables more effective training by dynamically adjusting the contribution of different terms, leading to more reliable convergence.

In the context of adaptive-weighting of the PDE residuals at domain collocation points, using trainable vectors was impractical due to the high number of collocation points and the random selection of the temporal points. Therefore, we defined different MLP networks - with spatiotemporal vectors as inputs, and adaptive weights at outputs - to provide self-adaptive weights for PDE residuals at various collocation points, as suggested by (32):

$$\omega^M = \mathcal{N}_\omega^M(x, y, z, t, \theta) \quad (68)$$

$$\omega^F = \mathcal{N}_\omega^F(x_f, y_f, z_f, t_f, \theta) \quad (69)$$

$$\omega^{MF} = \mathcal{N}_\omega^{MF}(x_f, y_f, z_f, t_f, \theta) \quad (70)$$

The MLP networks had a depth of 4 layers, each with a width of 40 neurons, activated by *tanh* activation functions. The weights of these networks were updated during each backpropagation gradient descent step.

Resampling of collocation points: The efficient simulation of the processes in 4D spatio-temporal spaces requires a significant number of collocation points to be chosen and computed during the training. These calculations create a significant computational load that makes the computations time-consuming. Various strategies, including resampling, have been proposed to address this challenge (33). In our study, we implemented a resampling strategy for the collocation points, introducing randomness at various unique periods during training. Specifically, resampling for the matrix collocation points occurred every 10 epochs, with approximately 33% of the spatial points being selected for resampling each time. The uniform sampling strategy inherently leads to a set of collocation points with statistical homogeneity. We have not applied the resampling strategy for the spatial points of the fractures.

Furthermore, the temporal coordinates corresponding to the points of the matrix and fractures were randomly resampled using a non-uniform square root of time (\sqrt{t}) distancing approach in the temporal range of [1 - 10^6] seconds. For the temporal points also the resampling period was 10 epochs.

Loss term (error) thresholding: To enhance the training efficiency and mitigate the risk of overfitting, we have incorporated an error thresholding technique into the PINNs model. This approach involves neglecting the impact of loss terms that fall below certain preset threshold values. By introducing these threshold limits, we effectively disregard the contributions of loss terms that are deemed insignificant or negligible, thereby simplifying the training landscape.

$$e = \max(0, e - \tau) \times w \quad (71)$$

In this equation, e is the value of the loss term, and τ is the threshold value. We applied this technique to modify the values of boundary and initial conditions for both saturation and pressure terms. The value of $\tau = 0.003$ for the saturation and pressure of the phases under initial or boundary conditions

Mathematics of the inverse calculations: A critical strategy in inverse computations using PINNs is the pre-training stage, at the beginning of the calculations. In the stage, the inverse parameters will be frozen for some epochs - about 2000 epochs in this work. This technique allows the model to learn the basic mathematical features of the system, before looking for the values of the inverse parameters. The length of pre-training stage depends on the complexity of the evaluating process.

After the freezing step has been ended, the values of the inverse parameters have been updated at the beginning of each epoch. The value of the inverse parameter γ have been calculated based on

$$\gamma = \gamma_i e^{\kappa \theta^*} \quad (72)$$

where γ_i is the initial value of the inverse parameter. Also, θ^* is the trainable parameter in neural networks, which have been initialised to be zero at the beginning of the training. κ is a regularisation term that is responsible for the training rate of the parameter. We have found that there should be a balance between the model training speed and the inverse parameter. The large values of κ makes the calculations unstable and prone to converge to the wrong values. The exponential form of the equation allows the algorithm to vary in scales of magnitude. Also, it prevents the model to reach to the nonphysical negative values.

To consider the uncertainties in the calculations, we defined a strategy to initialize the γ_i values randomly. It helps us in reaching a wide range of possible solutions for the problem, not only finding same solutions. Considering an initial value for the parameter of interest as γ_i^* , we write

$$\gamma_i = \gamma_i^* * 10^{\xi \xi_m} \quad (73)$$

A random number, denoted by ξ , is drawn from a uniform distribution between -1 and 1. This randomness is scaled by a model hyperparameter, ξ_m , which controls the overall level of randomness injected into the system. In this case, ξ_m was set to 0.6.

Convolutional kriging: To overcome with the measurement noises in the measured in-situ saturation data, we applied 3D denoising operations on the dataset. We employed a 3D convolutional kriging approach to effectively interpolate between the cell properties and their neighbour property

values. Convolutional denoising involves convolving a noisy data with a weighting kernel that performs smoothing to reduce the noises in the observations, while preserving the main features of the data. The kernel is an important part of the denoising. In this work, we utilized a semi-Gaussian weighted kernel that is shown in Fig. 2a. The algorithm of the computations is shown in Algorithm D. We have used similar kernels for all three dimensions. Finally, a comparison of the saturation distribution in the rock after some time is shown at two different experiment times is shown in 2b. We can see that the approach could effectively capture the main features of the data while removing unnecessary noise.

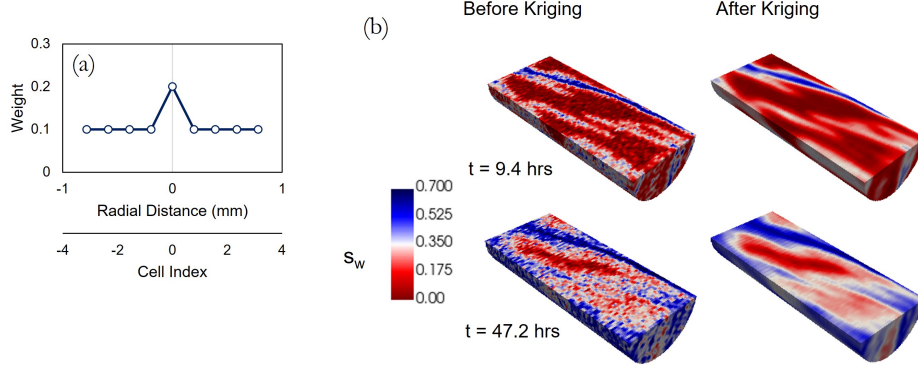


Figure 2: The impact of 3D convolutional kriging on the in-situ water saturation data. a) The applied weighting kernel, b) A comparisons of the saturation distribution before and after kriging at two different times

3D Convolutional Kriging [1] *inp*: 3D array of input data $k_1, k_2, k_3 \leftarrow (1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 1 \ 1 \ 1)$ the kernel along the dimensions $k_1, k_2, k_3 \leftarrow \text{normalize}(k_1, k_2, k_3)$ Normalize the kernels to ensure the mass balance is attained. *out* \leftarrow *inp* Initialize output array as input i in range(0, 3) *out* \leftarrow *convolve*(*out*, k_i , axis = i) Convolve over dimension i *out*

Normalization and denormalization: An essential pre/post processing step in training of PINNs is normalization and then denormalization of the inlet and outlet variables. By standardizing the input/output data into a common scale and distribution, it can greatly enhance the model’s stability and performance. In this work, we have utilized a Z-score normalization scheme. We normalized the PINNs inlets as

$$x_n = \frac{x - \bar{x}}{\sigma} \quad (74)$$

And then the outlets denormalized via

$$x = x_n \cdot \sigma + \bar{x} \quad (75)$$

In above equations \bar{x} and σ are the mean and standard deviation of the variable x in the domain under study, respectively.

Fast Fourier transformation (FFT): In previous studies, various works have highlighted the challenges of PINNs in solving high-frequency solutions due to an issue known as spectral bias (34). This bias causes the network to preferentially learn solutions with lower frequencies, not necessarily the most true solutions. A prominent strategy to address this involves incorporating Fast Fourier Transformation (FFT) within the network architecture (35; 29; 11). In this work, we also utilized a FFT operator, same as (29), to make the network able to capture the high-frequency characteristics of the solution. As it is visualized in Fig. 1, the normalized inlets are first encoded, then transformed to the frequency domain using an FFT operation. After processing by the main Multi-Layer Perceptron (MLP) network, the information is returned to the spatial domain via an inverse FFT operator. Only the real parts of the solution are retained, discarding the imaginary components. Table 3 compares

the performance of the workflow for solving the benchmark model, when different strategies of FFT transformation was applied. It confirms the effectiveness of the applied strategy in improving the performance of the model.

Table 3: The impact of using different Fourier Transformation strategies on the accuracy of the final results for the forward solution of the benchmark problem

Case	Lt (MAE)	RF (MAE)
without FT	2.51e-3	6.6e-2
with FT (imaginary)	5.30e-4	2.4e-2
with FT (real)	4.43e-4	2.2e-2

E Buckley-Leverett equation

At the first stage of the PINNs calculations, we have assumed the flow in fractures is governed by the Buckley-Leverett assumptions, meaning that flow is assumed to be a sole function of viscous forces, while capillary and gravity forces are neglected.

As a basic solution to this problem, we may follow the solution recommended by (36), called the Buckley-Leverett (BL) equation. The solution assumes that two fluids are incompressible and that the flow occurs unidirectionally. Rock properties (e.g., porosity and permeability) is considered homogeneous and constant. Also, the problem is considered isothermal with no interphase mass transfer between fluids. The mass conservation equation may be written as (37):

(76)

Rewriting the equation for a 1D problem, and assuming incompressibility of rock and fluids, leads to a further simplification as:

(77)

where, u_i is the Darcy flow velocity of each phase and is determined by Darcy's law:

(78)

Equation (78) can be rewritten by eliminating pressure in the equation and summing the phase equations of two phases:

(79)

In this equation, f_i is the fractional flow of phase water, and is written as:

(80)

We then employ the Welge tangent method (38) to calculate the local water saturation at each time step, as shown in the following equation

$$x = \frac{u_t}{\phi} \frac{\partial f_w}{\partial s_w} t \quad (81)$$

here, q_t is the total velocity of the fluids, which we can calculate it via

$$u_t = \frac{k k_{rw}^{max}}{\mu_w} \frac{(p_{in} - p_{out})}{l} \quad (82)$$

F Appendix: Computational Resources

In this study, we compared two simulation approaches: Finite Difference (FD) and PINNs. We utilized an in-house FD-based numerical simulator (39) as a benchmark tool. The FD numerical simulator were developed in C++ and run on an Intel i9-11950H CPU (2.60 GHz) with 32 GB RAM.

Individual forward simulations for a 27,000 cell model ranged from 30 minutes to 2 hours, depending on the complexity of effective parameters (e.g., flow parameter shapes and matrix-fracture property contrasts). The simulator were designed to effectively capture the experimental boundary conditions.

In contrast, PINNs computations were conducted using a single NVIDIA RTX A2000 GPU with the PyTorch 1.13.1+cu117 framework. Notably, PINNs simulations exhibited almost similar runtimes for both forward and inverse problems. For a standard inverse computation scenario, with a typical network size and number of collocation points as reported in Table 2, the inverse computations typically required nearly 1.5 - 2.5 hours.

G Appendix: History Matching

The PINNs history-matching computations were performed using Adam optimizer on the PyTorch computational framework. To compare with PINNs, we coupled the Finite-Difference (FD) simulator with a Nelder-Mead (NM) simplex optimization algorithm (40). The optimizor was deployed in SciPy library (41). We used the default meta-parameters.

For the synthetic problem, the static model of the system kept constant for all the cases. The goal was only history-matching of the RF curve. To achieve faster results while accepting a certain level of error, we utilized a static model with 15,625 cells, which is less than the original synthetic model. A summary of the optimization is shown in Fig. 3. After approximately 100 iterations, the errors in the objective function stabilized. As seen in the figure, the history-matching computations took around 20 hours to stabilize. Even after stabilization, the error in the estimated $\bar{\Lambda}$ curve was significant, with a normalized mean absolute error (NMAE) of 0.81. Fig. 3d shows the clear advantage of PINNs-based history-matching compared to FD-NM history-matching tool.

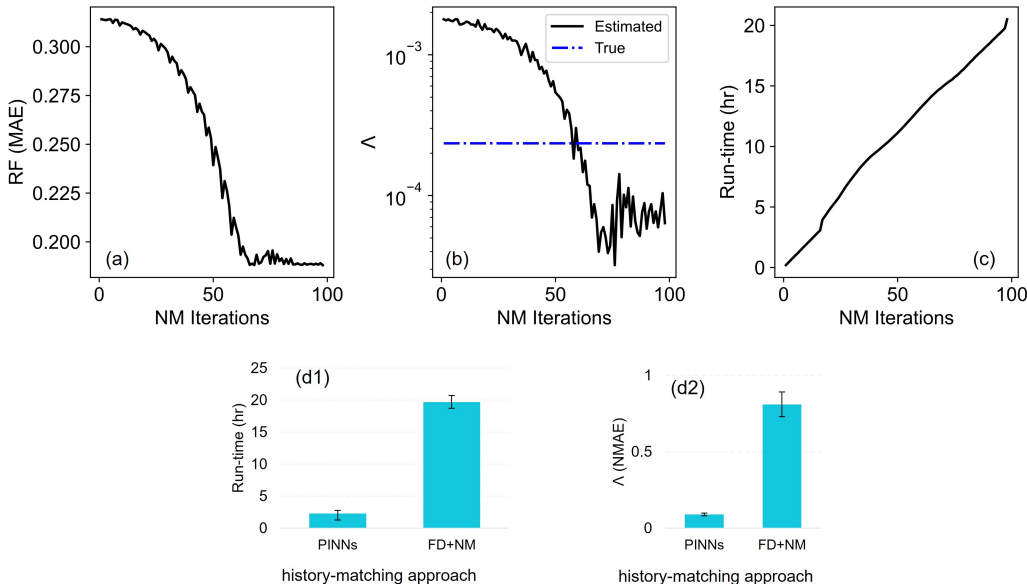


Figure 3: **The results of history-matching using coupled finite-difference numerical simulations, and Nelder-Mead (NM) optimization.** The history-matching continued for around 100 NM iterations. a) The MAE in the matched RF, b) The estimated $\bar{\Lambda}$ curve compare to the expected true value, c) The total run-time versus NM iterations, d) the performance (1: run-time, 2: NMAE in the estimations) of PINNs, and FD-NM history-matching tools are compared.

H Appendix: Sensitivity Analysis

Training Dynamics. The dynamics of training of PINNs is visualized in Fig. 4. It's evident that the precision of PINNs computations is enhanced as the training advances. Initially, the PINNs

predictions encapsulate the high-dimensional attributes of the flow within the matrix. Following this, the interactions between matrix and fracture flows have been learnt.

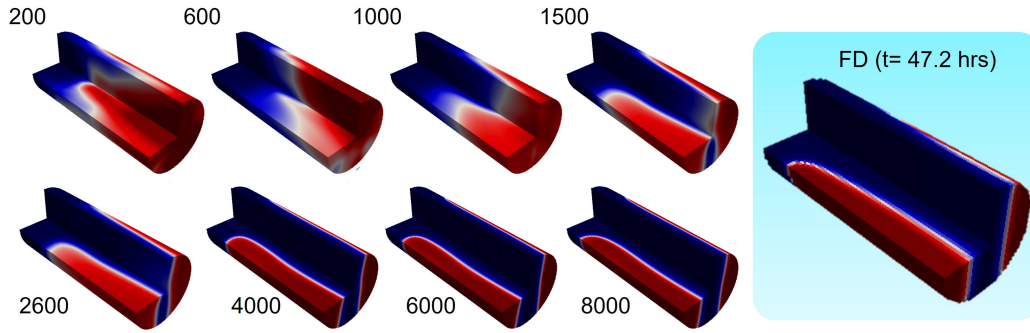


Figure 4: The dynamics of training PINNs at different epochs (200-8000) compared to the FD simulation result at same time ($t=47.2$ hrs).

Resampling of collocation points. As it is demonstrated in Fig. 5, utilizing resampling techniques significantly reduces errors while accelerating the computations. Resampling, applicable for both spatial and temporal domains, helps the network avoid getting stuck on local regions and steers it towards the optimal general solution of the system in the domain. This is particularly advantageous for large-scale problems requiring a substantial number of collocation points.

As Fig. 5b illustrates, the effectiveness of the resampling strategy depends on the frequency of resampling, or how often it's applied during training epochs. Resampling with very low frequency can not significantly help the optimization. Conversely, excessively frequent resampling prevents the system from learning the selected points effectively. In this context, a frequency between 0.03 and 0.1 (corresponding to resampling every 10-30 epochs) appears to be the optimal choice.

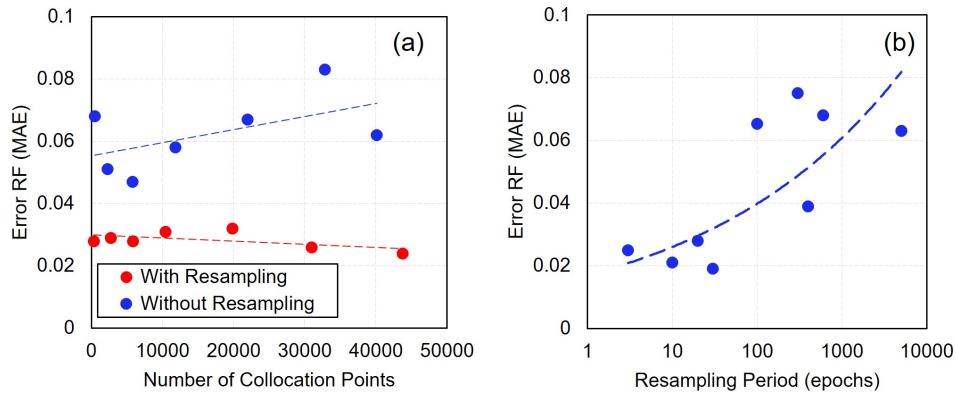


Figure 5: The sensitivity analysis on the number of collocation points in the matrix. All the models were trained for 20000 epochs. a) A comparison of the impact of the number of collocation points on the MAE in the predicted RF curve, b) The impact of resampling period on the accuracy of the predictions.