Application of Physics-Informed Neural Networks to Immiscible Compositional Problems

Hadjisotiriou, G.; Voskov, D.

DOI
10.3997/2214-4609.202437113

Publication date
2024

Document Version
Final published version

Published in
European Conference on the Mathematics of Geological Reservoirs, ECMOR 2024

ECMOR 2024
European Conference on the Mathematics of Geological Reservoirs

EAGE

Application of Physics-Informed Neural Networks to Immiscible Compositional Problems.

G. Hadjisotiriou[1], D. Voskov[1]

[1] Delft University of Technology

Summary

Carbon capture and storage is an essential technology to mitigate anthropogenic CO2 emissions from carbon-intensive industries. To model CO2 injection, physics-based numerical methods are computationally intensive due to the nonlinear nature of the governing equations. Therefore, several data-driven deep learning methods have been developed to serve as proxies and replace numerical simulations. These proxies have demonstrated significantly faster runtimes while maintaining comparable accuracy to numerical simulations. This makes them suitable for high-fidelity models and ensemble-based techniques that require a large number of forward runs. Our method utilizes physics-informed neural networks (PINNs) to parameterize the solution space of immiscible compositional problems. The PINN parameterizes the forward solution of the compositional problem based on the composition of the upstream grid block at the updated time step, the composition of the current grid block at the current time step and the total velocity at their interface. The neural network is trained in the entire solution space and is used in a sequential, cascading solver. In this approach, we obtain the pressure solution first before solving for transport by treating the reservoir as a series of two-cell problems. The resulting transport solver is applicable to all problems with different initial/injection conditions and different heterogeneous reservoirs. We demonstrate our approach for binary and multicomponent problems and furthermore use multilinear interpolation to compare and validate the solution method.

Introduction

Geological Carbon Storage (GCS) is a collection of technologies that capture CO_2 emissions from carbon-intensive industries for long-term geological storage in the subsurface. To safely, and permanently store CO_2, a geological trap consisting of a reservoir and an overlying impermeable caprock is required. Consequently, potential geologic storage sites for CO_2 include depleted oil and gas fields and deep saline aquifers. These reservoirs have stored buoyant hydrocarbons over geological time and additionally, existing data and infrastructure can be reused to manage project development's technical and financial risks.

Compositional models are used to capture the complex physical processes associated with CO_2 injection into depleted reservoirs. Multiphase multi-component compositional modeling of CO_2 injection models the transport of various components that are present in each phase. Solving this large, highly nonlinear system of equations is computationally expensive for large reservoirs and ensemble-based techniques e.g. uncertainty quantification, optimization problems, history matching (Müller et al., 2016).

Proxy modeling of CO_2 injection attempts to capture the behavior of the full-physics, high-fidelity model at a reduced cost. A wide range of proxy modeling methods exist. In particular, the most widely known and understood are upscaling techniques. During upscaling, a coarse scale model is made to mimic the behavior of the high-fidelity model. In the case of compositional models, single-phase flow, multiphase effects, and compositional solubilities are usually upscaled one by one using upscaling procedures. Multiphase and composition functions (also known as pseudo-functions) are defined by observations in the fine-scale model and subsequently transplanted in the coarse-scale model (Iranshahr et al., 2014). However, upscaling functions need to be defined at every coarse-scale interface and necessarily have to be redefined when boundary conditions change. Additionally, coarse-scale models that employ upscaling functions are susceptible to convergence issues. Barker and Thibeau (1997) give a review on the use of pseudo-functions for multiphase models.

An alternative proxy modeling method is to use a machine learning-based proxy. These proxies can replace the numerical simulation in part or entirely and have solution times that are orders of magnitude faster. Generally, two approaches can be identified within machine learning-based proxy modeling. The first predicts scalar values such as net present value and replaces the numerical simulator with the proxy to evaluate the objective function. The second maps the distribution of state variables in space and time with permeability and/or porosity maps as inputs. In the former, convolutional neural networks are the most explored algorithm (Wang et al., 2021a; Zhu and Zabaras, 2018; Zhao et al., 2023). Recently, Fourier Neural Operators (FNO) have also shown much promise in modeling CO_2 storage (Chu et al., 2022; Yan et al., 2021). Typically, these methods are data-intensive, sensitive to boundary conditions, and limited to two-dimensional models with reduced physics.

Physics-informed machine learning methods integrate data and physical models to manage the cost of data acquisition, facilitate learning of underlying governing equations, and provide solutions to partial differential equations (PDE). Physics-informed neural networks (PINNs) encode the residual form of the PDE in the loss function as a regularization term, thereby forcing the neural network to satisfy the PDE (Raissi et al., 2019). This enables training when data is sparse, uncertain, or high-dimensional (Karniadakis et al., 2021; Chen et al., 2021). Raissi et al. (2019) employ PINNs to provide solutions to Burgers' equations and the Korteweg-de Vries equation. However, it is found that PINNs are unable to resolve the defining feature of hyperbolic PDEs. Specifically, the location of the shocks in space and time (Fuks and Tchelepi, 2020). These shocks are sharp discontinuities in the composition or saturation profile of the solution. For hyperbolic PDEs, it was found that including diffusion and solving the parabolic form of the Buckley-Leverett problem provides an accurate approximation of the solution. Rodriguez-Torrado et al. (2022) design a more effective attention-based neural network to successfully solve the Buckley-Leverett problem. PINNs are an active field of research and improvements in the performance of PINNs focus on network architecture, sampling schemes, weighting schemes, and different training strategies. In a comprehensive review, Wang et al. (2023a) examine the training of PINNs.

The solution of highly nonlinear PDEs required for modeling CGS applications. Operator Based Linearization (OBL) is a simulation method that factorized governing conservation equation into operator form (Voskov, 2017). The operators are parameterized with respect to the state variables (pressure, saturation, etc.) of the model. Thereafter, an interpolator is used to evaluate operators and their partial derivatives during simulation. This linearizes the conservation equations and furthermore enables robust, flexible computation of derivatives. OBL is used in the academic simulator, open Delft Advanced Research Terra Simulator (Voskov et al., 2024). The OBL approach allows the simulation of various energy transition applications (GCS, geothermal, etc.) within the same framework (Wang et al., 2021b; Lyu et al., 2021; Wang et al., 2023b; Lyu and Voskov, 2023).

In this work, we parameterize the transport solution for immiscible compositional problems with observations from the solution of a nonlinear model. Olawoyin and Younis (2023) parameterize the single-cell equation for two-phase advection-diffusion problems and apply their parameterization with a neural network as a preconditioning strategy. Here we parameterize the solution space in control volume with respect to its current state, the state of the upstream grid block at the updated time step, and the velocity at their interface. Observations are used to train a deep neural network to predict the transport solution for a multi-component compositional simulation. The discretized form of the residual equation is included in the loss function and provides an additional training signal to both expedite training and minimize the reliance on labeled data in higher dimensional parameter space. In addition, multilinear interpolation is used as a benchmark and alternative to the neural network. The resulting parameterization is utilized to provide the transport solution within a sequential, cascading solver. The resulting simulation framework is equally applicable to different reservoirs with different heterogeneities, and spatial and temporal discretizations. Furthermore, once trained, the neural network does not need to be retrained for different input parameters including distribution of permeability, porosity, and initial and injection compositions.

In the next section, the elements of the proposed simulation framework are discussed. In the section after, examples of binary and multi-component solutions are shown. We conclude the paper with discussion on future development of the proposed technique.

Simulation framework

In this section, the simulation framework is described with details on parametrization, training, and following cascading solver.

Compositional simulation

For simplicity, the capillarity, gravity and diffusion are neglected in the formulation of the conservation equations. Then, the general mass conservation equation for component c is defined as follows:

$$\frac{\partial}{\partial t}(\phi \sum_{j}^{N_p} x_{cj}\rho_j s_j) + \nabla \sum_{j}^{N_p} x_{cj}\rho_j \mathbf{u_j} = 0, \quad c = 1,...,N_c. \tag{1}$$

In the mass conservation equation, t is time, ϕ porosity, x_{cj} molar fraction of component c in phase j, ρ_j phase density and s_j the phase saturation. The Darcy velocity, $\mathbf{u_j}$, is calculated according to,

$$\mathbf{u}_j = -\mathbf{K}\frac{k_{rj}}{\mu_j}\nabla p. \tag{2}$$

In the Darcy equation $\mathbf{K}$ is absolute permeability, k_{rj} relative permeability, μ_j phase viscosity and p pressure.

For an incompressible, one-dimensional reservoir, the governing equations are simplified to,

$$\phi\frac{\partial z_c}{\partial t} + u_t\frac{\partial F_c}{\partial x}, \quad c = 1,...,N_c. \tag{3}$$

The overall composition, z_c, is given by,

$$z_c = \sum_{j}^{N_p} x_{cj} v_j, \tag{4}$$

and is calculated from the partitioning coefficients and phase fractions, v_j. The fractional flow per component, F_c, is calculated as,

$$F_c = \sum_{j}^{N_p} x_{cj} f_j, \tag{5}$$

and is a function of the phase fractional flow f_j,

$$f_j = \frac{s_j^2}{s_j^2 + M(1-s_j)^2}, \tag{6}$$

where M is the mobility ratio μ_g/μ_o. The thermodynamic equilibrium relations between liquid and vapor close the system and are expressed by,

$$\hat{f}_{cl}(p,T,\mathbf{x_l}) = \hat{f}_{cv}(p,T,\mathbf{x_v}), \tag{7}$$

where $\hat{f}_{cl}$ and $\hat{f}_{cv}$ are the liquid and vapour fugacities. The phase fraction v_j and partitioning coefficients x_{cj} are determined with the Rachford Rice equation and constant K-values. The Rachford Rice equations reads as follows:

$$h(v) = 0 = \sum_{c}^{N_c} \frac{z_c(K_c - 1)}{v_i(K_c - 1) + 1}. \tag{8}$$

Single-cell parameterization

Observations from a two-cell model are used to parameterize the forward transport solution. The conservation equation for incompressible isothermal flow is discretized implicitly with backward Euler approximation in time and forward Euler approximation in space. For a model with two cells the residual equation for a given component reads,

$$r = z_2^{n+1} - z_2^n + \theta(F(z_2^{n+1}) - F(z_1^{n+1})), \tag{9}$$

where $n+1$ and n correspond to the next and current time step. The Newton-Rhapson method is applied to solve the equations to a tight tolerance. Consequently the forward solution z_i^{n+1} is parameterized,

$$z_i^{n+1} = N(\theta, z_i^n, z_{i-1}^{n+1}), \tag{10}$$

with respect to the the state of the current grid block i at the current time step n, the state of the adjacent/upstream grid block $i-1$ at the updated time step $n+1$, and the total velocity at the grid block interface included in the quantity θ, defined as:

$$\theta = \frac{u_t dt}{\phi dx}. \tag{11}$$

Training procedure

Data is generated from observations from the two-cell model and are used as supporting data for multilinear interpolation or as training data for a deep neural network. Neural networks are well suited to higher dimensional regression problems and are therefore colloquially referred to as universal approximation functions for their theoretical ability to approximate any function. In this case, the dimensionality of the problem is equal to $2 \times N_c - 1$, since the sum of compositions is equal to 1 and we include θ. The input vector of the neural network is defined as,

$$\mathbf{x} = \{\theta, \mathbf{z}_i^n, \mathbf{z}_{i-1}^{n+1}\}, \tag{12}$$

where i refers to the grid block, n the time step and $\mathbf{z}$ the vector of compositions. The neural network predicts the vector of compositions at the updated time step $\hat{\mathbf{z}}_i^{n+1}$ and is defined recursively by,

$$a_l(\mathbf{x}) = \sigma(\mathbf{W}_l x + \mathbf{b}_l), \tag{13}$$

and,

$$\hat{\mathbf{z}}_i^{n+1} = a_L(a_{L-1}(a_{L-2}(...(a_1(\mathbf{x}))))), \tag{14}$$

where $\mathbf{W}^l$ is the weight matrix of the l-th layer, $\mathbf{b}^l$ is the bias vector and σ is the element wise activation function. The resulting parametrization is equally applicable to different initial and injection conditions since the forward solution is parameterized with respect to the entire parameter space. In addition since θ is included the time step, grid resolution and permeability can be changed without having to retrain the neural network.

Many successful applications of deep learning based proxies within reservoir simulation are data-driven. These supervised algorithms typically require more data in order to increase or maintain accuracy for higher-dimensional problems. However, methods developed for unsupervised learning, specifically explicit regularization techniques that penalize the loss function by including prior knowledge in the loss function, have a proven ability to make an underdetermined problem determined. This is particularly advantageous when the data is scarce or when there is uncertainty regarding the data points. In this case, regularization can increase the capabilities of the neural network beyond what can be offered by just the labelled data. For these reasons, we propose a physics-informed neural network where the right-hand side of the loss function consists of two terms: a regression term and a residual term,

$$L = \frac{1}{N}\sum_{j}^{N} |\mathbf{z}_2^{n+1} - \hat{\mathbf{z}}_2^{n+1}|_j^2 + \frac{1}{N}\sum_{j}^{N} |\hat{\mathbf{z}}_2^{n+1} - \mathbf{z}_2^n + \theta(F(\hat{\mathbf{z}}_2^{n+1}) - F(\mathbf{z}_1^{n+1}))|_j^2. \tag{15}$$

In the regressive term the mean squared error of the prediction denoted by $\hat{\mathbf{z}}_2^{n+1}$, and the label denoted by, $\mathbf{z}_2^{n+1}$, is calculated. In the residual term, the discretized form of the residual is included. The calculation of $F(\mathbf{z})$ is carried out with OBL. Two types of data points are used that are either uniformly or randomly sampled from the parameter space. The labels for the data points distributed uniformly are known from observations of the two-cell model, as described in eq. (9). The random collocation points are labeled by interpolating with the uniformly distributed points. This approach cheaply and effectively augments the data set. The gradient of the loss function is computed by back-propagation and the adaptive moment estimation (Adam) optimization algorithm is used with exponential decay.

Cascading solver

The parameterized solution space is used to solve for transport in a sequential cascading solver. In this method pressure and transport are decoupled and pressure is solved first. Thereafter, based on the obtained pressure and corresponding total velocity field, the gridcells are reordered according to their potential from high to low. This defines the order of the cascade sweep. In the cascade sweep compositions are calculated one cell at a time starting with the cell with the highest potential. This is valid assuming that cells with higher potential are treated first and that there is no counter current flow (Appleyard and Cheshire, 1982). With the algorithm below, we solve a one-dimensional incompressible system with injection boundary condition on the left and pressure boundary condition on the right.

There are three things of note to consider with the proposed approach. Firstly, the solution time is proportional to the reservoir size as is the case with established numerical methods as well. Secondly, error accumulates with every time step. Thirdly, a sequential solution method introduces so-called splitting errors. Kwok and Tchelepi (2007) use a potential based ordering scheme for two- and three-phase black

Algorithm 1: Cascade method

```
**Pressure step**: compute pressure update
Calculate $\theta$ at the grid block interfaces
**Potential-orderding**: reorder cells such that $p_{i-1} > p_i$
**Transport step**: solve sequence of single-cell problems for transport update
**for** $i = 1, \ldots, n_b$ **do**
    $z_i^{n+1} = N(\theta, z_{i-1}^{n+1}, z_i^n)$
```

oil models in a reduced-order Newton method. It is applicable to multidimensional models containing countercurrent flow and capillarity and effectively speeds up solution time by allowing greater time-steps. Their approach is further extended by Hamon and Tchelepi (2016) and applied to immiscible compositional problems.

Numerical results

The transport solution of a binary and four component compositional system is parameterized and deployed within a cascading solver. In our examples, the pressure solution is obtained from DARTS before solving for transport. For comparison, the transport solution is obtained either by interpolation or with the PINN. The obtained solution, $\hat{z}$, is compared to the reference solution, z. The error maps shown compute the error as the absolute error,

$$E = |z - \hat{z}|. \tag{16}$$

We present compositional models with different injection/initial conditions and furthermore apply our simulation framework to heterogeneous 2D and 3D models.

An illustrative example for a binary system

It is found that the effectiveness of the solution strategy depends largely on the resolution of the supporting points (e.g. training data) used and the degree of non-linearity contained within the problem. Fig. 1, illustrates the solution space of a binary system with small and large θ values. For larger time steps and finer space discretization the θ values are larger and the non-linearity of the solution space increases. In a sparse data regime, the accuracy of the solution obtained with interpolation deteriorates. Fig. 2 estimates the residual for a random set of collocation points from low- and high-resolution data and further illustrates the dependency on the underlying data. Correspondingly, in the example shown in Fig. 3 the number of points is limited to 25 for the entire solution space. At this resolution of training data it is evident that interpolating is inadequate as the location of the leading shock is underestimated. Conversely, the physics-informed neural network is able to resolve the location of the shocks.

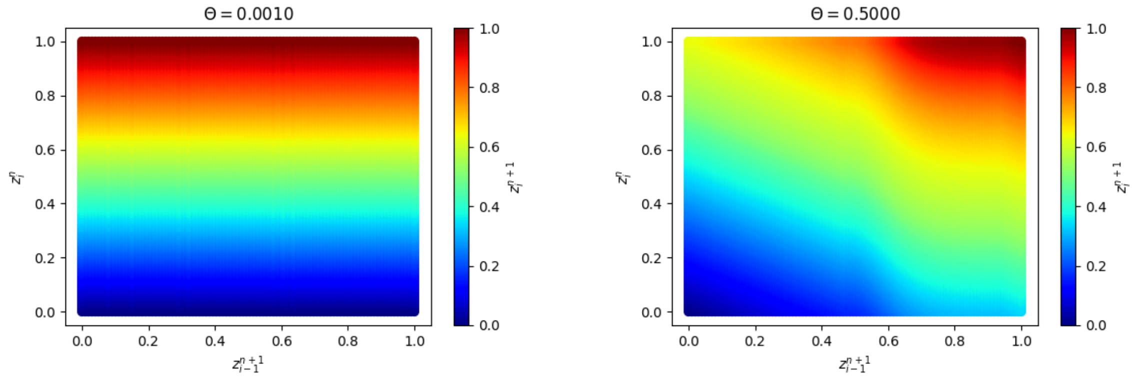


Figure 1: Solution space of a binary problem for different θ. K-values of each component are equal to 2.0 and 0.1.

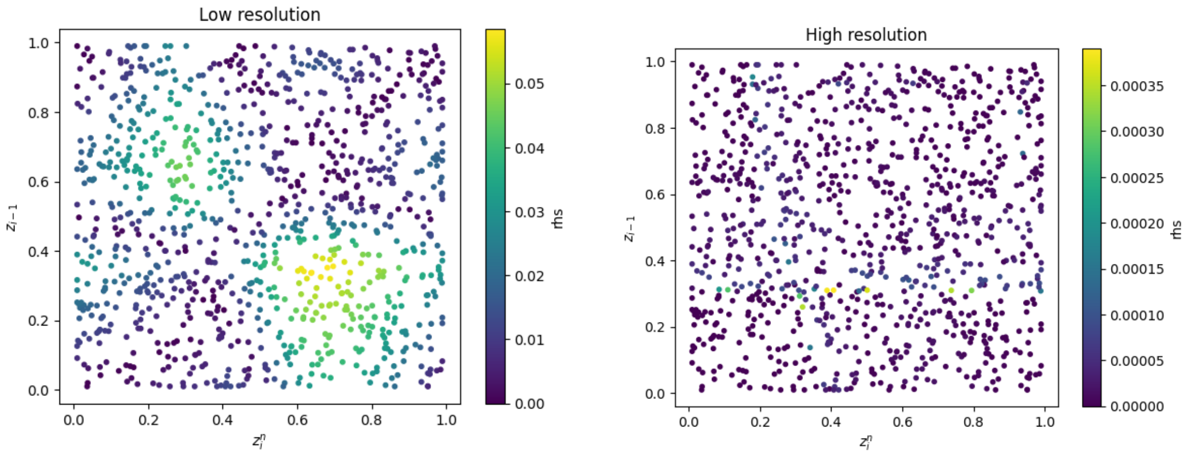


Figure 2: Right hand side of the provided solution for a set of random points with low and high resolution supporting data.

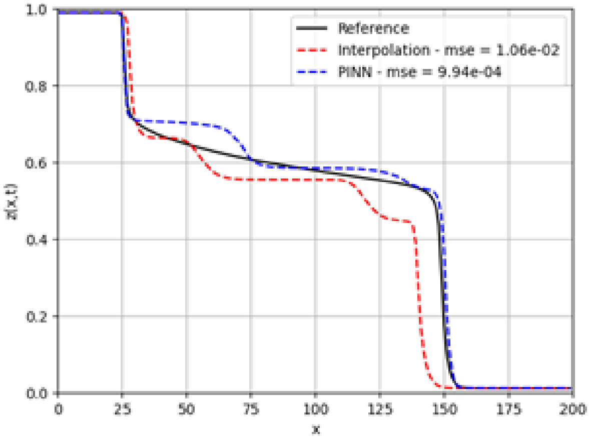
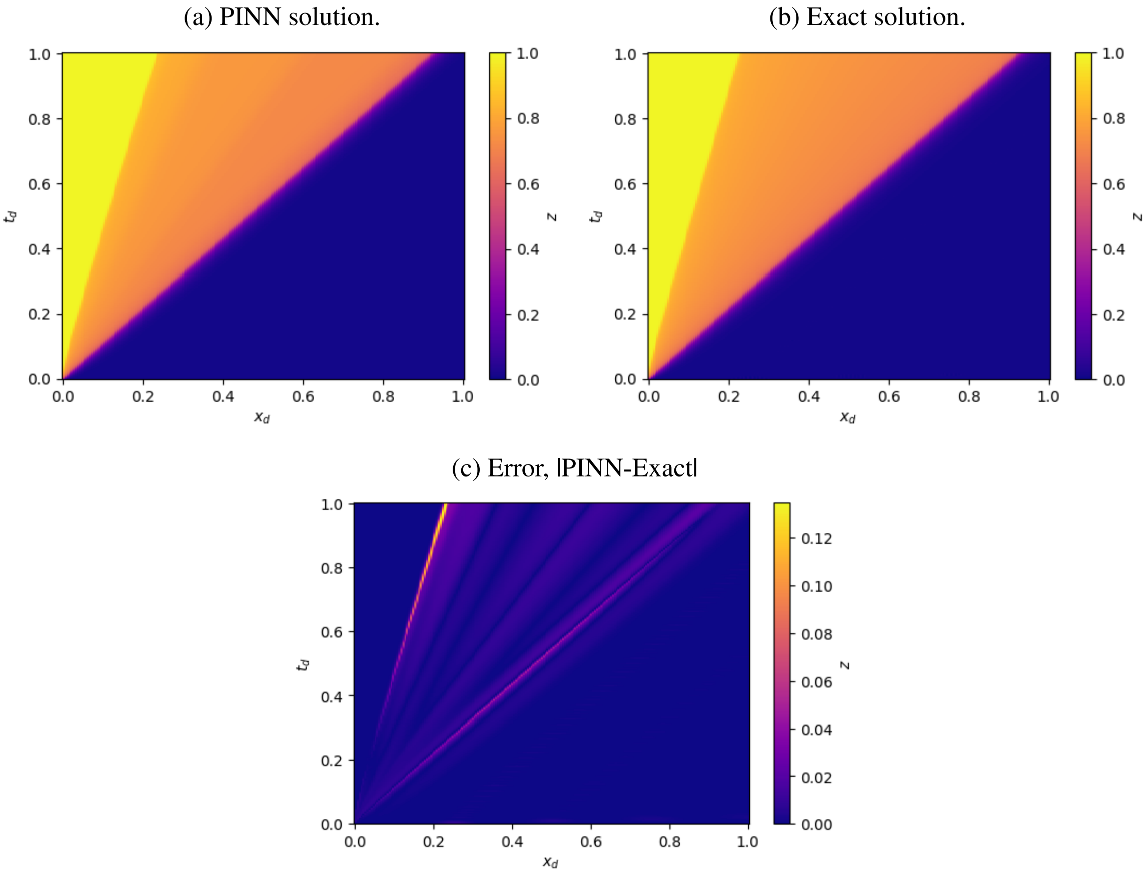


Figure 3: Compositional profile the PINN versus interpolation and reference solution. When operating in a sparse data regime the PINN is able to outperform the interpolation.

A standard fully connected neural network is trained for a binary system. The K-values of each component are 2.0 and 0.1. The network architecture has a depth of 3 hidden layers and 125 neurons per layer. An equal number of training data and collocation points are used. The training data set contains 1000000 data points and is sampled uniformly from the parameter space with a minimum value for θ of 0.001 and maximum of 0.5. Fig. 4 plots the solution in space and time of the PINN versus that of the numerical simulation. Theta in this example is equal to 0.5. The PINN accurately approximates the exact numerical solution. Late in the simulation there is a small misfit of the trailing shock.

Two-dimensional example for binary system

We apply our simulation framework with interpolator to two different layers representing the near shore environment of SPE10 problem (Christie and Blunt, 2001a), (case 1, Fig. 5a) and fluvial environment (case 2, Fig. 5b). In these examples, the total velocity and thus θ is calculated per interface and provided as an input to compute the transport solution. The model is discretized by $100 \times 20 \times 1$ cells with $dx = 10m$, $dy = 5m$ and $dz = 1m$. The porosity is assumed to be constant and equal to 0.3. The initial composition of the reservoir is equal to 0.01 CO_2 and 0.99 CO_2 is injected. In case 1, CO_2 is injected in the middle of the model at a constant rate. Producers are located in each corner of the model operating at fixed bottom hole pressure. In case 2, pressure drives the flow from the left edge to the right edge. The left edge of the model has a constant injection boundary and the right edge a constant pressure boundary. The provided solution, $\hat{z}$, is shown in Figs. 5c and 5d is very close to the reference solution, z, in both



(a) PINN solution.

(b) Exact solution.

(c) Error, |PINN-Exact|

Figure 4: Solution in space and time for a one-dimensional simulation with θ constant and equal to 0.5.

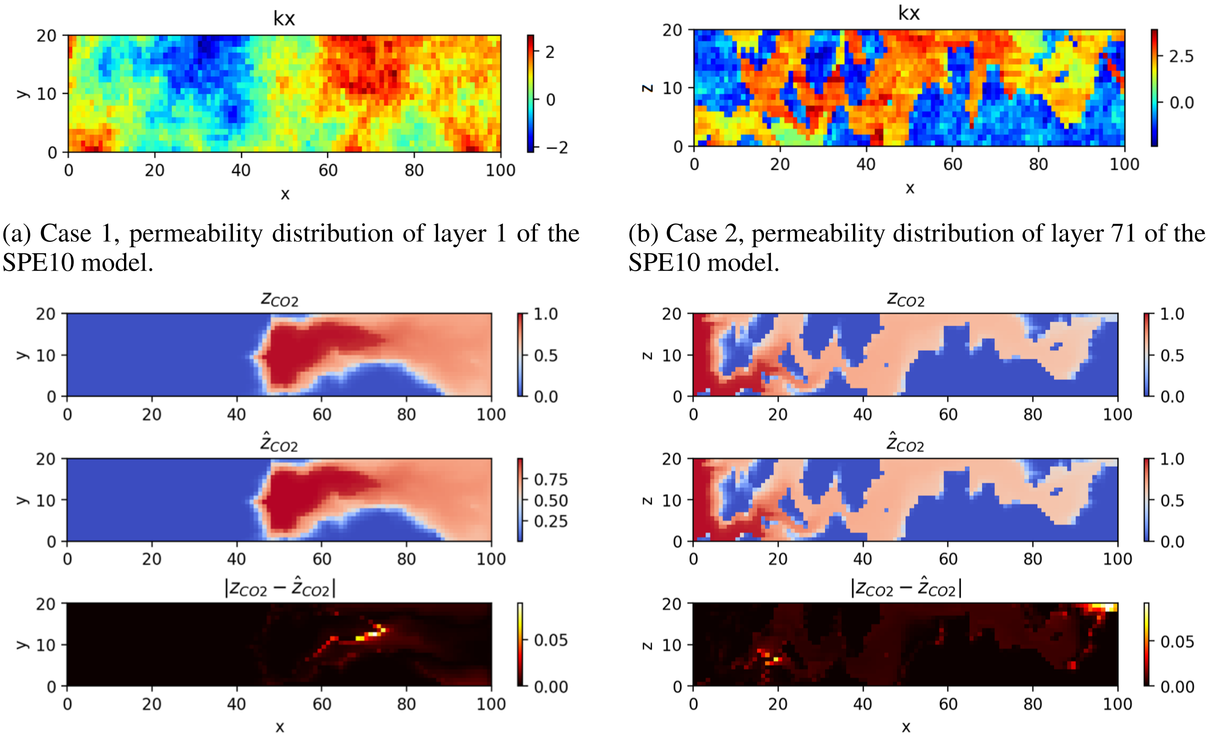
cases.

Three-dimensional example for binary system

CO_2 is injected in the top 20 layers of the SPE10 model. The SPE 10 model is cropped and discretized by $100 \times 20 \times 20$ cells. The grid cell dimensions are $5m \times 5m \times 1m$. An injection well injects in the middle of the reservoir and perforates the entire interval. Production wells are located in the corners of the reservoir and operator with fixed bottom hole pressure. The initial composition is $0.01CO_2$ and injection composition $0.99CO_2$. The simulation is run for $10000 days$ for a total of $0.1PV$ injected. Figure 6 shows the obtained composition and the corresponding error.

Simulations for four component systems

Here, we illustrate how interpolation can be adapted for modeling of multicomponent systems. The adaptive multilinear interpolator of DARTS is applied to various four component problems. The adaptive interpolator computes two-cell solutions as necessary which act as supporting points for interpolation. In the following examples, the minimum and maximum per axis is tailored to the problem and a resolution of 30 points per dimension, excluding θ, is used. Note that fixing θ is equivalent to fixing the injection rate of the model. In Fig. 7 CO_2 is injected in a depleted reservoir. In Fig. 8 and 9 additional challenging test cases from Orr (2007) are simulated. In all of the shown cases it is found that the interpolator is able to resolve all of the features of the compositional profile.



(a) Case 1, permeability distribution of layer 1 of the SPE10 model.

(b) Case 2, permeability distribution of layer 71 of the SPE10 model.

(c) Case 1, overall compostion of CO_2. Comparison of the fine-scale reference solution, z, and proxy solution $\hat{z}$.

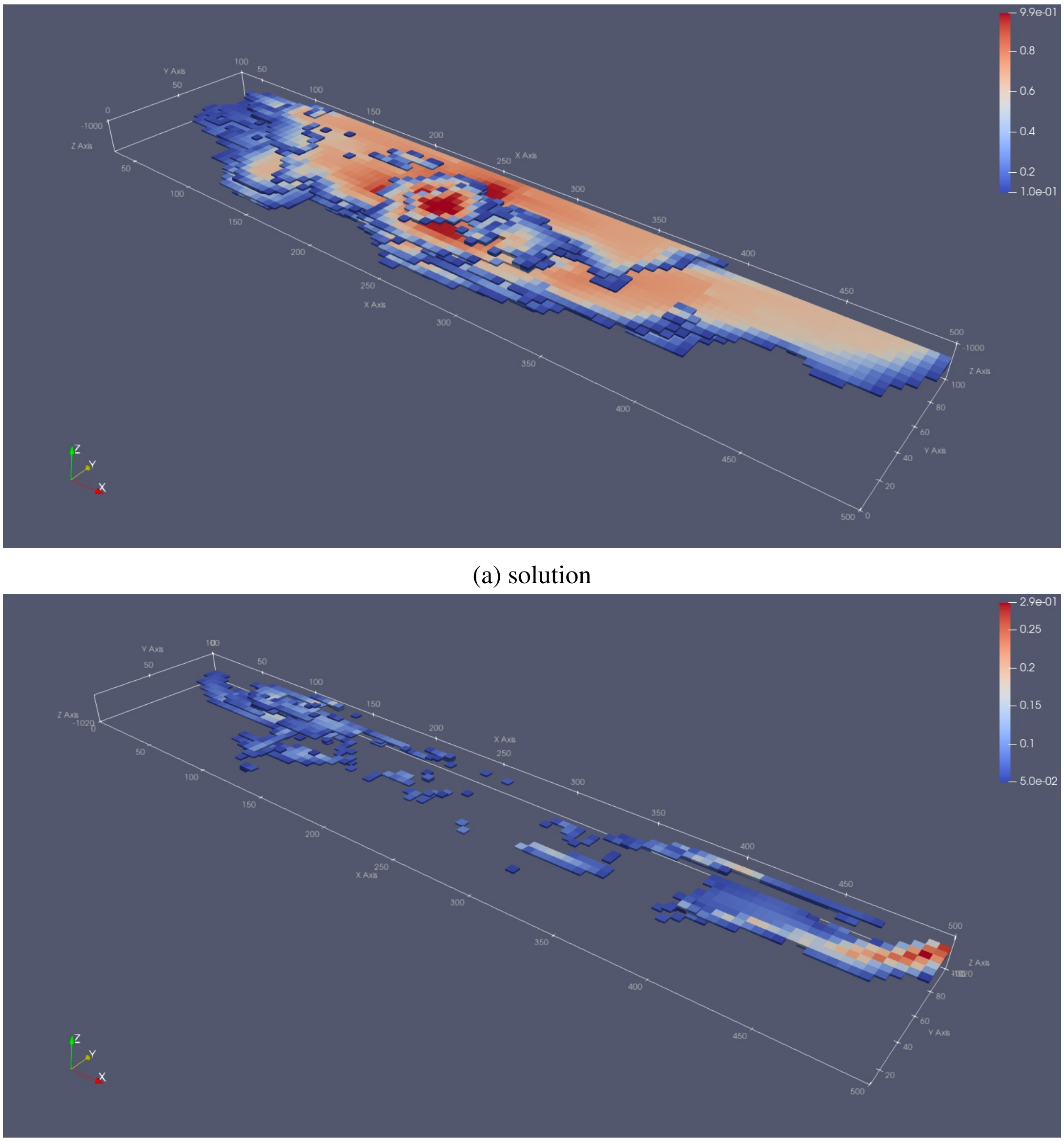
(d) Case 2, overall composition of CO_2. Comparison of the fine-scale reference solution, z, and proxy solution $\hat{z}$.

Figure 5: Two-dimensional test cases.

PINN for four component system

Parameterizing and interpolation in the entire solution space for higher dimensional problems is difficult as the amount of data points increases exponentially. In the case of four components and a uniform sampling of 30 points per dimension parameterizing the entire solution space amounts to a data set containing billions of data samples. For this reason a PINN is trained globally for a 4-component system containing four components: $CH4$, CO_2, $C4$ and $C10$ and a constant θ equal to 0.2. The training data set consists of 944784 data points sampled uniformly in the entire parameter space. Additionally, the residual is calculated for an equal number of random collocation points. The neural network has a depth of 8 layers and width of 80 neurons per layer.

The trained neural network is applied to various challenging gas injection problems. Fig. 10 shows the results for the vaporizing and condensing gas drive treated previously with the interpolator in Fig. 9 and 8. The PINN accurately approximates the solution for each of these cases with an accurate estimation of the shocks as well as a good approximation of the composition in the two-phase zone located between. The neural network is trained in the entire solution space and is therefore equally applicable to various problems with different initial and injection composition conditions. CO_2 is injected into a reservoir with various initial compositions listed in table 1. The solutions are shown in Fig. 11. The neural network does not perform equally well for each case. In particular, in some of the cases, the composition in the unswept part of the reservoir deviates from the initial composition.



(a) solution

(b) error

Figure 6: CO_2 concentration after 10000 days. The initial composition is 0.01 CO_2 and injection composition 0.99 CO_2.

Table 1: Initial compositions

	CH4	CO_2	C4	C10
A	0.01	0.01	0.39	0.59
B	0.10	0.18	0.37	0.35
C	0.10	0.00	0.20	0.70
D	0.30	0.00	0.16	0.54

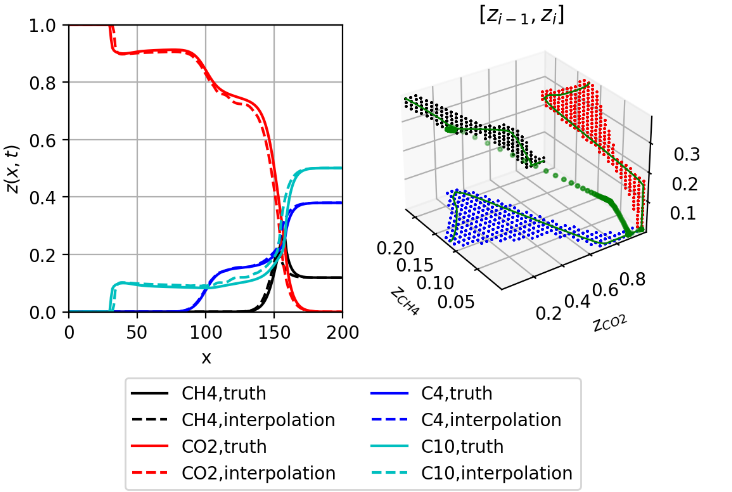


Figure 7: Injection of pure CO_2 in a depleted reservoir containing 0.1192 $CH4$, 0.3798 $C4$, and 0.5010 $C10$. K-values are constant and equal to $K_{CO_2} = 1.5$, $K_{CH4} = 2.5$, $K_{C4} = 0.5$ and $K_{C10} = 0.05$.

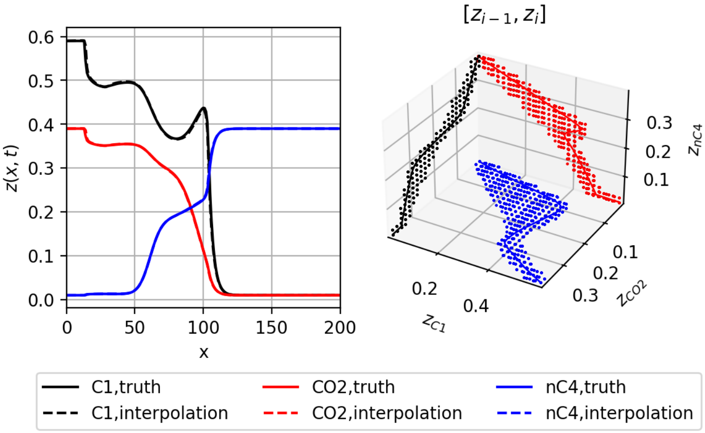


Figure 8: Condensing gas drive - Injection of 0.59 $C1$ and 0.39 CO_2. Initial composition is 0.01 $C1$, 0.01 CO_2, 0.39 $C4$ and 0.59 $C10$. K-values are constant and equal to $K_{CO_2} = 1.5$, $K_{C1} = 2.5$, $K_{C4} = 0.5$ and $K_{C10} = 0.05$.

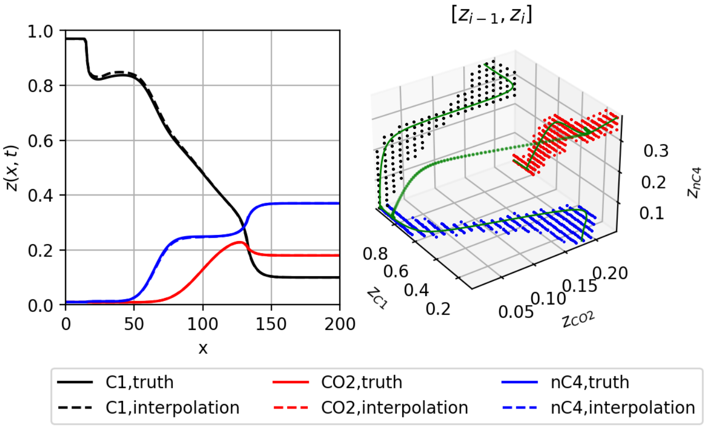


Figure 9: Vaporizing gas drive - Injection of 0.97 $CH4$. Initial composition is equal to 0.10 $CH4$, 0.18 CO_2, 0.37 $C4$ and 0.35 $C10$. K-values are constant and equal to $K_{CO_2} = 1.5$, $K_{CH4} = 2.5$, $K_{C4} = 0.5$ and $K_{C10} = 0.05$.

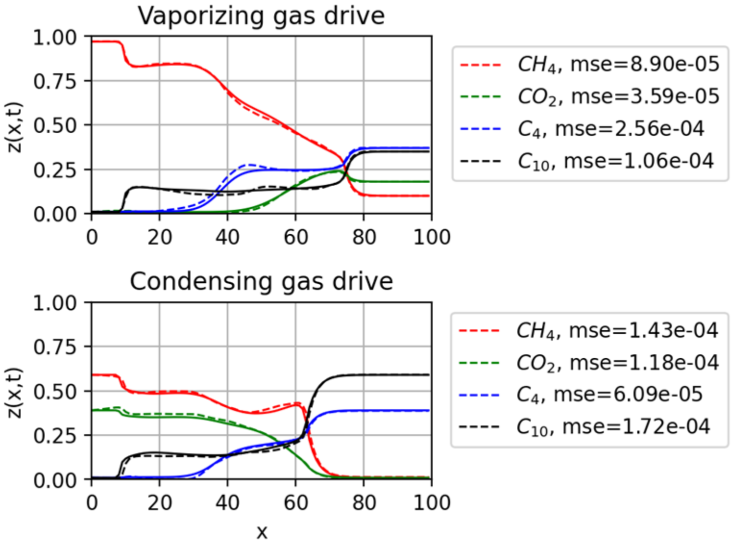
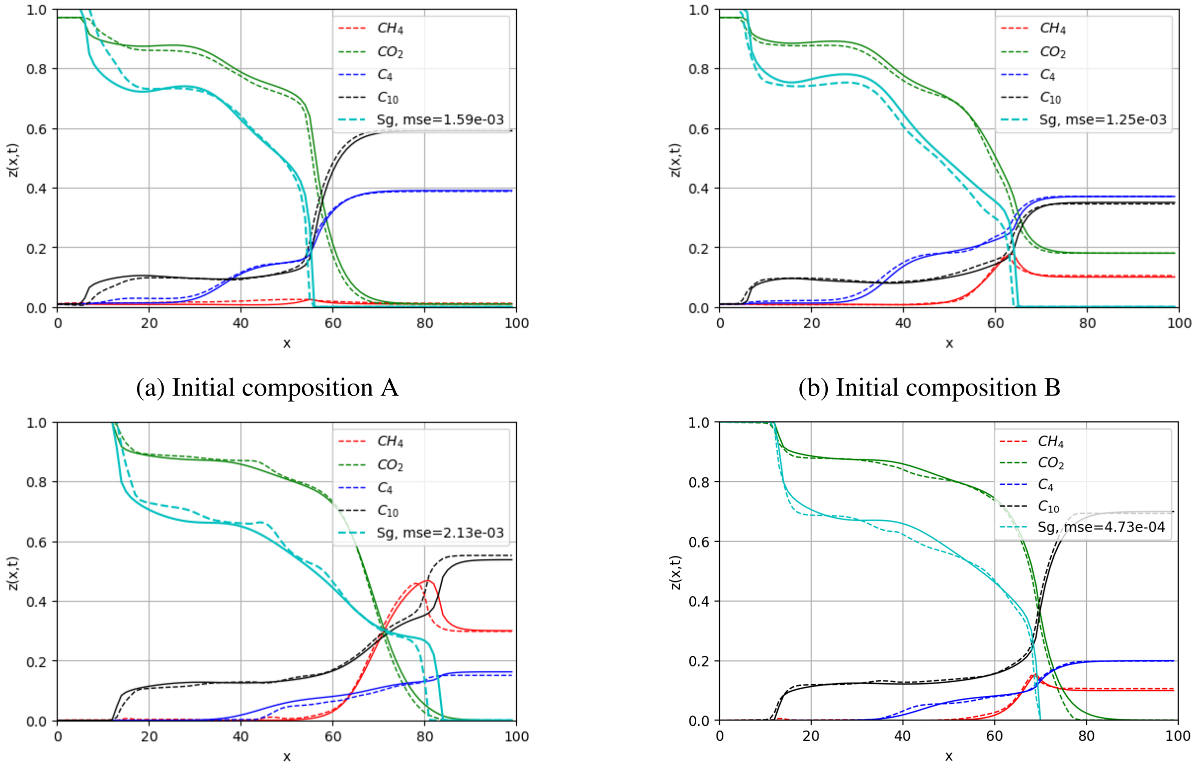


Figure 10: PINN solution (dashed line) versus reference solution for condensing and vaporizing gas drive. K-values are constant and equal to $K_{CO_2} = 1.5$, $K_{C1} = 2.5$, $K_{C4} = 0.5$ and $K_{C10} = 0.05$

.



(a) Initial composition A

(b) Initial composition B

(c) Initial composition C

(d) Initial composition D

Figure 11: PINN solution (dashed line) versus reference solution for various CO_2 injection problems with different initial compositions. The initial compositions are shown in table 1. K-values are constant and equal to $K_{CO_2} = 1.5$, $K_{C1} = 2.5$, $K_{C4} = 0.5$ and $K_{C10} = 0.05$

.

Conclusion and discussion

The solution space for multicomponent, immiscible compositional problems is parameterized and integrated into a sequential, cascading solver. The transport solution at the updated time step is provided by physics-informed neural networks or multilinear interpolation. This solution is derived from the current composition, the composition of the upstream grid block at the updated time step, and the total velocity at the grid block interface. Interpolation serves as a baseline estimate of performance. Multilinear interpolation is capable of reconstructing the reference solution when provided with high-resolution interpolation tables. Nevertheless, in high-dimensional, multicomponent solution spaces, a physics-informed solution strategy is proposed as a practical alternative. The physics-informed neural network is trained with data points sampled uniformly and randomly from the entire parameter space. This approach ensures that the neural network does not require retraining for different initial/injection compositions. The neural network-provided transport solution is capable of accurately capturing the shocks and connecting waves of the compositional profile of the reference solution. Furthermore, the proposed simulation framework is capable of treating reservoirs with different spatial and temporal discretizations and heterogeneities. This is accomplished by incorporating the coefficient θ into the parameterization of the solution space. Theta is defined as the ratio of $(u_t\Delta t)/(\phi\Delta x)$ and is analogous to the Courant–Friedrichs–Lewy (CFL) number.

Discussion

The proposed simulation framework presents a step towards utilizing PINNs for multidimensional heterogeneous reservoirs. However, identically to standard numerical methods, the simulation proceeds iteratively and requires a mesh. This is in stark contrast with the PINNs by Raissi et al. (2019), which provide the solution for all points in space and time simultaneously. One of the stated advantages of PINNs is that it offers a meshless approach to reservoir simulation. However, analytical representation of heterogeneous reservoirs is challenging.

With regards to solution time and accuracy. It is found that the solution time is proportional to the size of the reservoir and the depth of the neural network. However, the simulation framework does not employ a Newton-Raphson solver and carries the implicit solution into an explicit implementation of simulation. At this time, an apt and extensive comparison of solution times is pending. In the proposed simulation framework errors from the current time step are carried into the updated time step. Therefore, it is conceivable that a transfer training approach could be employed, whereby the model is initially trained globally and subsequently fine-tuned with data for a specific case.

At present, our model neglects gravity and capillarity in order to reduce complexity and focus on the transport solution. The influence of gravity is of great importance for the modelling of CO_2 injection. In future research these phenomena will be considered as well.

Acknowledgements

G. Hadjisotiriou acknowledges Shell Netherlands for supporting his work as a PhD at the TU Delft under the ASSET project.

References

Appleyard, J.R. and Cheshire, I.M. [1982] The Cascade Method for Accelerated Convergence in Implicit Simulators. In: *All Days*. SPE.

Barker, J.W. and Thibeau, S. [1997] A Critical Review of the Use of Pseudorelative Permeabilities for Upscaling. *SPE Reservoir Engineering*, **12**(02), 138–143.

Chen, Z., Liu, Y. and Sun, H. [2021] Physics-informed learning of governing equations from scarce data. *Nature Communications*, **12**(1), 6136.

Christie, M.A. and Blunt, M.J. [2001a] Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. In: *All Days*. SPE.

Christie, M.A. and Blunt, M.J. [2001b] Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. In: *All Days*. SPE.

Chu, A.K., Benson, S.M. and Wen, G. [2022] Deep-Learning-Based Flow Prediction for CO2 Storage in Shale–Sandstone Formations. *Energies*, **16**(1), 246.

Fuks, O. and Tchelepi, H.A. [2020] LIMITATIONS OF PHYSICS INFORMED MACHINE LEARNING FOR NONLINEAR TWO-PHASE TRANSPORT IN POROUS MEDIA. *Journal of Machine Learning for Modeling and Computing*, **1**(1), 19–37.

Guo, Z. and Reynolds, A.C. [2018] Robust Life-Cycle Production Optimization With a Support-Vector-Regression Proxy. *SPE Journal*, **23**(06), 2409–2427.

Hamon, F.P. and Tchelepi, H.A. [2016] Ordering-based nonlinear solver for fully-implicit simulation of three-phase flow. *Computational Geosciences*, **20**(5), 909–927.

Iranshahr, A., Chen, Y. and Voskov, D.V. [2014] A coarse-scale compositional model. *Computational Geosciences*, **18**(5), 797–815.

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.

Khait, M. and Voskov, D.V. [2017] Operator-based linearization for general purpose reservoir simulation. *Journal of Petroleum Science and Engineering*, **157**, 990–998.

Kwok, F. and Tchelepi, H. [2007] Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media. *Journal of Computational Physics*, **227**(1), 706–727.

Lyu, X. and Voskov, D. [2023] Advanced modeling of enhanced CO2 dissolution trapping in saline aquifers. *International Journal of Greenhouse Gas Control*, **127**, 103907.

Lyu, X., Voskov, D., Tang, J. and Rossen, W.R. [2021] Simulation of Foam Enhanced-Oil-Recovery Processes Using Operator-Based Linearization Approach. *SPE Journal*, **26**(04), 2287–2304.

Ma, Z., Kim, Y.D., Volkov, O. and Durlofsky, L.J. [2022] Optimization of Subsurface Flow Operations Using a Dynamic Proxy Strategy. *Mathematical Geosciences*, **54**(8), 1261–1287.

Müller, F., Jenny, P. and Meyer, D.W. [2016] Parallel Multilevel Monte Carlo for Two-Phase Flow and Transport in Random Heterogeneous Porous Media With Sampling-Error and Discretization-Error Balancing. *SPE Journal*, **21**(06), 2027–2037.

Olawoyin, A.A. and Younis, R.M. [2023] Learning to Solve Parameterized Single-Cell Problems Offline to Expedite Reservoir Simulation. In: *Day 2 Wed, March 29, 2023*. SPE.

Orr, F. [2007] *Theory of Gas Injection Processes*. Tie-Line Publications, Copenhagen.

Raissi, M., Perdikaris, P. and Karniadakis, G. [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.

Rodriguez-Torrado, R., Ruiz, P., Cueto-Felgueroso, L., Green, M.C., Friesen, T., Matringe, S. and Togelius, J. [2022] Physics-informed attention-based neural network for hyperbolic partial differential equations: application to the Buckley–Leverett problem. *Scientific Reports*, **12**(1), 7557.

Voskov, D., Saifullin, I., Wapperom, M., Xiaoming, T., Palha, A., Orozco, L. and Novikov, A. [2024] open Delft Advanced Research Terra Simulator (open-DARTS).

Voskov, D.V. [2017] Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. *Journal of Computational Physics*, **337**, 275–288.

Wang, N., Chang, H. and Zhang, D. [2021a] Efficient Uncertainty Quantification and Data Assimilation via Theory-Guided Convolutional Neural Network. *SPE Journal*, **26**(06), 4128–4156.

Wang, S., Sankaran, S., Wang, H. and Perdikaris, P. [2023a] An Expert's Guide to Training Physics-informed Neural Networks.

Wang, Y., de Hoop, S., Voskov, D., Bruhn, D. and Bertotti, G. [2021b] Modeling of multiphase mass and heat transfer in fractured high-enthalpy geothermal systems with advanced discrete fracture methodology. *Advances in Water Resources*, **154**, 103985.

Wang, Y., Voskov, D., Daniilidis, A., Khait, M., Saeid, S. and Bruhn, D. [2023b] Uncertainty quantification in a heterogeneous fluvial sandstone reservoir using GPU-based Monte Carlo simulation. *Geothermics*, **114**, 102773.

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A. and Benson, S.M. [2022] U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, **163**, 104180.

Yan, B., Chen, B., Harp, D.R. and Pawar, R.J. [2021] A Robust Deep Learning Workflow to Predict Multiphase Flow Behavior during Geological CO2 Sequestration Injection and Post-Injection Periods.

Zhao, M., Wang, Y., Gerritsma, M. and Hajibeygi, H. [2023] Efficient simulation of CO2 migration dynamics in deep saline aquifers using a multi-task deep learning technique with consistency. *Advances in Water Resources*, **178**, 104494.

Zhu, Y. and Zabaras, N. [2018] Bayesian Deep Convolutional Encoder-Decoder Networks for Surrogate Modeling and Uncertainty Quantification.