Hybrid Newton method for the acceleration of well event handling in the simulation of CO2 storage using supervised learning

Antoine Lechevallier, Sylvain Desroziers, Thibault Faney, Eric Flauraud, Frédéric Nataf

Hybrid Newton method for the acceleration of well event handling in the simulation of CO_2 storage using supervised learning

Antoine Lechevallier[12] Sylvain Desroziers[3]
Thibault Faney[2] Eric Flauraud[2] Frédéric Nataf[1]

[1]Laboratoire Jacques-Louis Lions
[2]IFP Energies Nouvelles
[3]Michelin

September 21, 2023

1 Abstract

CO_2 geological storage is an essential instrument for efficient Carbon Capture and Storage policies. Numerical simulations provide the solution to the multi-phase flow equations that model the behavior of the CO_2 injection site.

However, numerical simulations of fluid flow in porous media are computationally demanding: it can take up to several hours on a HPC cluster in order to simulate one injection scenario for a large CO_2 reservoir if we want to accurately model the complex physical processes involved. This becomes a limiting issue when performing a large number of simulations, e.g. in the process of 'history matching'.

During the numerical simulation of CO_2 storage in the subsurface, well events cause important numerical difficulties due to their instant impact on the pressure and saturation unknowns. This often forces a drastic reduction of the time step size to be able to solve the non-linear system of equations resulting from the discretization of the continuous mathematical model. However, these specific well events in a simulation have a relatively similar impact across space and time.

We propose a methodology to alleviate the impact of well events during the numerical simulation of CO_2 storage in the underground. We complement the standard numerical algorithm by predicting an initialization of Newton's method directly in the domain of convergence using supervised learning.

More specifically, we replace the initialization in pressure by a linear approxima-

tion obtained through an implicit solver and we use a Fourier Neural Operator (FNO) to predict the saturation initialization.

We apply our methodology to two test cases derived from a realistic $C0_2$ storage in saline aquifer benchmark. We reduce the required number of Newton iterations to handle a well opening by 53% for the first test case, i.e required number of linear system to solve and by 38% for the second test case.

Contents

2 Introduction

Context

The reduction of CO_2 emissions into the atmosphere is mandatory to achieve ecological transition. CO_2 geological storage is an essential instrument for efficient Carbon Capture and Storage (CCS) policies. Numerical simulations provide the solution to the multi-phase flow equations that model the behavior of the CO_2 injection site. They are an important tool to decide whether or not to operate a potential carbon storage site and to monitor these operations (long term storage of injected CO_2, potential gas leakage, optimal positioning of CO_2 injection wells, etc.).

However, numerical simulations of fluid flow in porous media are computationally demanding: it can take up to several hours on a HPC cluster in order to simulate one injection scenario for a large CO_2 reservoir if we want to accurately model the complex physical processes involved. This becomes a limiting issue when performing a large number of simulations, e.g. in the process of "history matching" : in order to fit the various model parameters to match the available historical data, a large number of simulations corresponding to various parameter sets needs to be performed.

More specifically, well events (opening and closure) cause important numerical difficulties due to their large impact on the pressure and saturation unknowns. This often forces a drastic reduction of the time step size to be able to solve the non-linear system of equations resulting from the discretization of the continuous mathematical model. However, these specific well events in a simulation are relatively similar across space and time: the degree of similarity between two well events depends on a few parameters such as the injection condition, the state of the reservoir at the time of the event, the boundary conditions or the porous media parameters (permeability and porosity) around each well. Therefore, the potential of machine learning algorithms to alleviate numerical difficulties related to well events is worth investigating.

State of the art

Recent interest in machine learning applied to the prediction of physical processes has fueled the development of "Physics Informed Deep Learning", wherein machine learning models either substitute or enhance traditional numerical algorithms while preserving the inherent constraints from the physical model. These models can be trained in a supervised or unsupervised manner.

In unsupervised learning, no labeled data is available and the objective is to directly enforce physical constraints. This can be achieved, for instance, by minimizing the residual of the partial differential equations governing the evolution of the solution or by penalizing deviations to mass conservation, etc. Physics Informed Neural Networks (PINN) , as introduced by Lagaris et al. [14] in 1998 typically is an example of unsupervised learning applied to the resolution of

differential equations. More recently in 2019, Raissi et al. [22] remastered the PINN approach in a semi-supervised manner in order to solve PDE problems.

In supervised learning, the objective is to match the labeled data available from experiment or previous simulations. In particular, the supervised learning of mappings between infinite dimensional function space through neural operators [13] [19] has yielded notable outcomes in learning PDE solutions [28]. Especially Fourier Neural Operator (FNO) [16] has demonstrated great performances and a lot of recent architectures based on it have emerged [34] [17] [23].

Physics Informed Deep Learning have a been applied to geological sequestration of CO_2 using a wide range of machine learning techniques, either for Physical property prediction [2][38][41][25][40][4][12][37][36][30][18][35][29][27][5][33], CO_2 migration and leakage analysis [32][31] or reservoir monitoring [21]. In most of the cases, the resulting machine learning model is used as a surrogate model that supplant the traditional solver. This allows for faster evaluation of new input parameters in contrast to the original numerical solver. This becomes especially valuable when the numerical solver is computationally expensive and time-consuming, as surrogate models can provide quick approximations. However, this 'black-box' approach lacks a guarantee of predicting solutions accurately, in contrast to the reliability of a numerical solver. Consequently, hybrid approaches that complement the numerical solver with a machine learning model while preserving the numerical guarantees are worth considering for evaluation.

Contribution

We propose a methodology to alleviate the impact of well events during the numerical simulation of CO_2 storage in the subsurface. We complement the standard numerical algorithm by predicting an initialization of Newton's method directly in the domain of convergence using a supervised learning approach based on recently developed Fourier Neural Operators. Our results show a significant decrease in the number of Newton iterations required for convergence, while ensuring the convergence to the correct solution.

Plan

In the first section 3 We begin by formulating the continuous model alongside the numerical solver associated with underground CO_2 storage. Next, in section 4, we provide a comprehensive exposition of our methodology, while section 5 outlines the database generation process using two test cases inspired by a realistic benchmark. Finally, we apply our methodology to the test cases and present the results in section 6.

3 Problem formulation

3.1 Continuous model

Let us consider a two-phase fluid composed of an aqueous phase noted w and a gaseous phase noted g flowing in a porous medium. We are interested in determining the evolution of the pressure P_α and saturation S_α in each phase.

The standard approach of modeling a two-phase flow in a porous medium is based on the application of Darcy's law [11] for each phase $\alpha \in \{w, g\}$ that relates the phase velocity v_α with the pressure gradient:

$$v_\alpha = -\frac{\overline{\mathrm{K}} k_{r_\alpha}(S_\alpha)}{\mu_\alpha} \nabla P_\alpha, \tag{1}$$

where $\overline{\mathrm{K}}$ is the permeability tensor, k_{r_α} is the relative permeability that depends on the saturation S_α, and μ_α is the dynamic viscosity.

We consider the medium as isotropic, hence the tensor of permeability $\overline{\mathrm{K}}$ can be considered as a scalar field K.We consider reservoir geometries such that gravity can be neglected. Capillary effects are neglected, therefore $P_w = P_g$. Finally, we use a quadratic relative permeability law , i.e $kr_\alpha(S_\alpha) = S_\alpha^2$

For incompressible fluids, the conservation equations describing the evolution of the pressure P and saturations S_w and S_g are then:

$$\phi \frac{\partial}{\partial t}(S_w) + div(v_w) = 0, \quad \phi \frac{\partial}{\partial t}(S_g) + div(v_g) = q_g, \tag{2}$$

$$v_w = -\frac{K kr_w(S_w)}{\mu_w} \nabla P, \quad v_g = -\frac{K kr_g(S_g)}{\mu_g} \nabla P, \tag{3}$$

$$S_g + S_w = 1, \tag{4}$$

where ϕ is the medium porosity field and q_g is the gas flow injected in the wells.

We directly substitute S_w for $1 - S_g$ using equation (4) so that we can drop the indices: the unknowns are the pressure P and the gas saturation denoted by S.

We suppose that the permeability K and the porosity ϕ vary in space but are constant through time. The viscosities μ_α are constant.

3.2 Numerical resolution

The continuous model is discretized using a standard two-points finite volume spatial scheme on a cartesian mesh and an implicit Euler time scheme. The pressure and saturation are solved simultaneously through a fully implicit scheme [6][15]. Finally, the resulting non-linear system of equations is solved using Newton's method.

Newton initialization

In the 'standard' or 'classic' Newton's method, the initialization for step $n+1$ is derived from the solution X_{sol}^{n} obtained in the preceding step n. Therefore, we denote $X_{init}^{n+1} = X_{sol}^{n}$ the standard initialization of the step $n+1$.

Time step management

In theory, the time step of a fully implicit reservoir simulator is not restricted by stability since the Euler implicit time scheme is unconditionally stable. However, in practice, the standard Newton's method may encounter converge issues with large time steps. To address this, a common practice is to allow a maximum number of Newton iterations N_{max} for the Newton's method to converge. If convergence is not achieved, we reduce the time step by a factor two and start over. This sometimes necessitates multiple time step reductions to achieve Newton convergence, resulting in a significant number of superfluous iterations.

3.3 Impact of well events during the numerical resolution

Well events generate pressure and saturation discontinuities (in time) inside the reservoir and lead to nonlinear convergence problems as they act as singular point sources that are tightly coupled to the reservoir model [1]. These discontinuities often prevent Newton's method from converging while attempting to solve the system as the initial guess may be far from the solution. In such situations, we attempt to solve the system again with a reduced time step and continue this process until convergence is achieved. Well events can thus account for a significant portion of the actual simulation time. An illustrative scenario, inspired by the test case 2 discussed in Section 5, is depicted in Figure 1. This scenario involves a simulation of 40 time steps, each spanning 2 years. At 4 years, an injection well is opened using an well injection flow rate of $10^{-3.1}$ and closed at 24 years. We observe that immediately following the well opening, Newton's method necessitates 40 iterations to reach convergence, with a minimum of 5 iterations required at each subsequent step throughout the entire well opening process. In contrast, in the absence of well events, the maximum number of Newton iterations is 5 immediately after well closure and 3 otherwise. Consequently, the well event-induced Newton iterations constitute the majority of the total Newton iterations throughout the entire scenario, accounting for 54%. Furthermore, if the maximum allowable iteration count were less than 40, the time step would be halved, resulting in significantly longer simulation times.

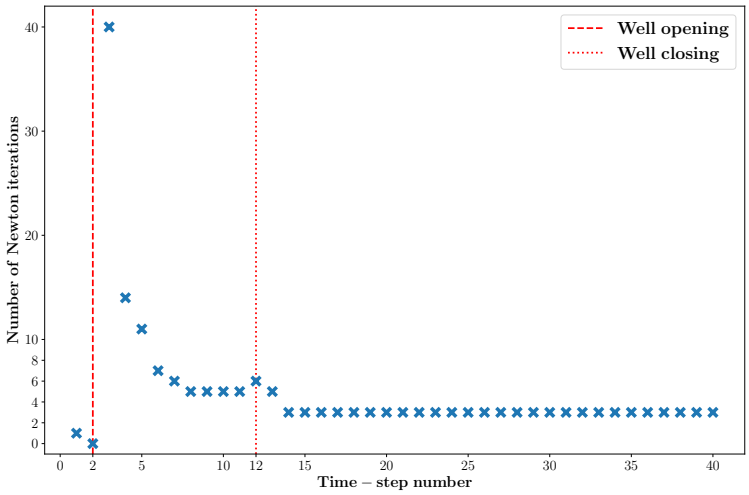

Figure 1: Example of scenario and it's impact on Newton's method. The scenario is composed of a well event opened after 2 time steps (red dashed line) and closed at 12 time steps (red dotted line).

4 Methodology

We propose a methodology based on the Hybrid Newton algorithm [20] [10] [39] [3] to alleviate the impact of well events. It consists in predicting via machine learning an initialization closer to the solution than the standard initialization, hence accelerating the Newton's method convergence while still guaranteeing the correct solver solution. We use a fixed reservoir geometry and configuration where a well event occurs at a specific location. We aim to alleviate the impact of the well opening for a wide range of scenarios.

4.1 Hybrid Newton algorithm

The hybrid Newton algorithm is a modification of the standard Newton algorithm with respect to the initialization. Indeed, the hybrid method proposes to initialize with a more accurate solution instead of using the standard initialization (i.e. the previous time step solution). By using an initialization closer to the solution, we aim to converge in fewer iterations.

In this article, our primary focus is on the hybrid guess construction X_{ML}^{n+1} and its performance compared to the standard initial guess X_{init}^{n+1}. We will compare them through their impact on the number of Newton iterations.

4.2 Initial guess construction

In the hybrid Newton algorithm, a prediction guess $X_{ML}^{n+1} := (P_{pred}, S_{pred})$ is required. We propose the following construction for the pressure and saturation:

1. P_{pred}: Implicit Pressure solver
2. S_{pred}: Prediction using supervised learning

4.2.1 Pressure

During a well event, the pressure discontinuity is global in the reservoir and instantaneous in time. We propose to use the solution of an implicit pressure solver P_{imp} (IMP) [24] at the step $n+1$ as a prediction guess P_{pred}. The implicit pressure solver solves the linear elliptic equation (5) and catches the main global variations of pressure but does not catch the small local variations of pressure due to saturation variations. Moreover, the implicit pressure solver only requires to solve a linear system. Therefore, we consider that it is not worth it to train a machine learning model as we have a cheap good approximation.

$$div(v(P_{imp}, S^n)) = 0 \tag{5}$$

with $v = v_g + v_w$, v, v_g and v_w are respectively the total velocity, the gas velocity and the water velocity inside the reservoir.

4.2.2 Saturation

During a well event, the saturation discontinuity inside the reservoir is local and near to the well. One could use an implicit saturation solver (IMS) which would require a Newton's method to solve a non linear system. Therefore, we propose to predict the saturation using a neural network considering that the inference of a neural network is rather fast compared to the multiples Newton iterations of the IMS solver which each require to solve a linear system. The predicted saturation is denoted S_{ML}.

The standard methodology uses the initial guess $X_{init}^{n+1} = X_{sol}^{n} = (P_{sol}^{n}, S^n)$ and the hybrid methodology uses the initial guess $X_{ML}^{n+1} = (P_{imp}, S_{ML})$. Given that we aim to conduct a fair assessment of the impact of a saturation predictive model, we compare the hybrid initialization with the standard initialization. Therefore we use in this article the following initial guess for the standard methodology $X_{init}^{n+1} = (P_{imp}, S^n)$.

In the end, we will compare in terms of Newton iterations the two following initialization:

1. Standard initialization: $X_{init}^{n+1} = (P_{imp}, S^n)$
2. Hybrid initialization: $X_{ML}^{n+1} = (P_{imp}, S_{ML})$

4.3 Neural Network architecture

The objective is to predict using reservoir and well information the global saturation state reached after a well event. A well event at a specific location and a specific well geometry can be described with few parameters: an injection well flow q_g and a time-step dt. A neural network architecture with good predictive capability for different physics-based processes is required.

Fourier Neural Operator

Neural networks are commonly used to learn relationships between finite-dimensional spaces, but they can struggle to adapt to changes in governing equations or conditions [7]. The Fourier Neural Operator (FNO) [16] addresses this issue by learning relationships between infinite-dimensional spaces using data-driven methods. This allows the FNO to understand the rules governing an entire family of partial differential equations. Additionally, the FNO improves computational efficiency by converting convolution operations in neural networks to multiplication through the use of discrete Fourier transforms.

Selected Architecture

We use the architecture presented on figure 2 based on an uplifting dense layer, four Fourier Layers (see figure 3) and two dense layers. We use the Gaussian Error Linear Units (GELU) [9] as activation function for each layer. We use two different versions of this general architecture for the training on the two test cases. **Nc** and **Ni** are parameters respectively representing the number of channels and the number of inputs.

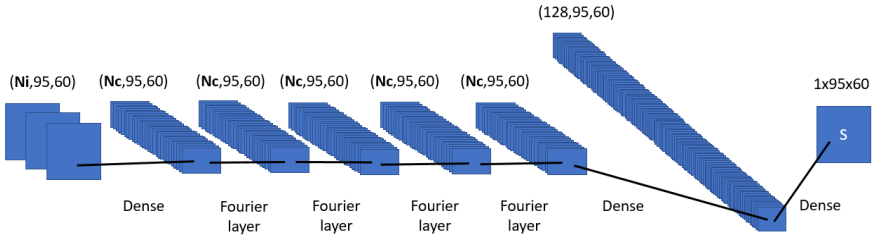

Figure 2: Selected neural network architecture composed by an uplifting dense layer, four Fourier layers, a dense layer and a projection dense layer. **Nc** and **Ni** are parameters respectively representing the number of channels and the number of inputs.

The architecture of a Fourier layer fig. 3 is composed of two parts, one that apply Fourier transform, a linear transform on the lower Fourier modes and a filter on the higher mode, then it applies the inverse Fourier transform. The other part is composed of a local linear transformation applied to the original input. Finally, the output of the two parts are added together and an activation function is applied.

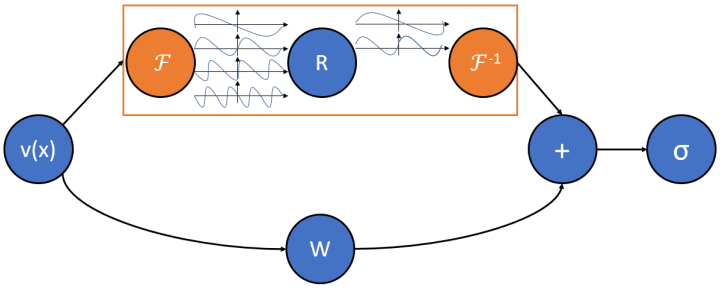

Figure 3: The Fourier layer starts with an input vector v, applies the Fourier transform $\mathscr{F}$ to it, then performs a linear transformation R on the lower Fourier modes while filtering out the higher modes. The inverse Fourier transform $\mathscr{F}^{-1}$ is then applied. A local linear transformation W is applied to the original input vector v. The output of the top and bottom operations are then added together and an activation function is applied.

Input Features

There are multiple features that can be considered as input features for the selected neural architecture. We select four possible input features, the time-step dt, the injection flow rate q_g, the initial saturation S^n and the implicit pressure P_{imp}. As S^n and P_{imp} have the same shape as the output feature, they can be used straightforward. However, as q_g and dt are scalars, they need to be reshaped. We propose to reshape dt into a constant map of value dt everywhere. For the injection flow rate q_g, we reshape it to a map of value zero everywhere except at the well location where it takes the value q_g.

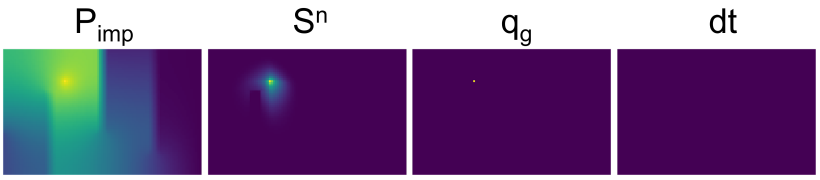

Figure 4: Qualitative view of Neural network input feature possible assembly.

5 Test case and Database Generation

5.1 SHPCO2 benchmark

Context

As a practical use case, we use the SHPCO2 [8] benchmark to test our methods. This benchmark was created for modelling reactive transport for CO_2 geological

storage. The SHPCO2 geological configuration is inspired from Sleipner, the world's first commercial CO_2 storage project. Sleipner is an area located in the North-Sea, belonging to Norway and exploited for its natural gas field since the mid-1990s.

As the natural gas produced contains up to 9% CO_2, the Sleipner CO_2 gas processing and capture unit is built to evade the expensive 1991 Norwegian CO_2 tax. The captured CO_2 is thus injected and stored in a deep saline formation one kilometer below the seabed.

Reservoir configuration

We adapt the original 2D SHPCO2 benchmark by removing the gas zone and replacing it by a well at its center.

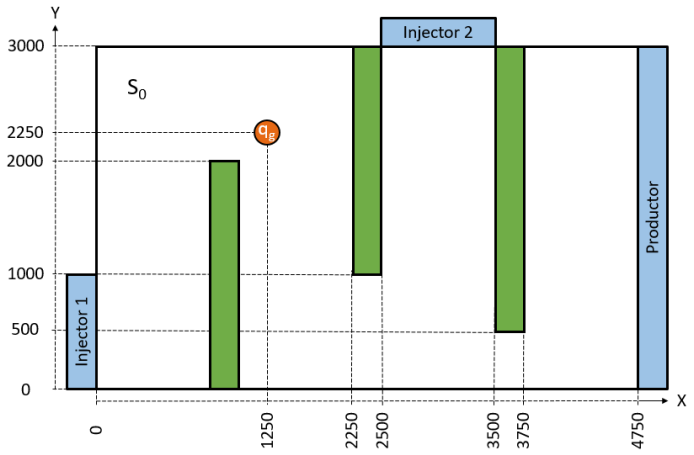

Figure 5: Adapted 2D SHPCO2 case geometry

The domain after modification is separated in two zones, the first zone called "Barrier zone" is coloured in green on the figure 5 and the second zone called "Drain zone" is formed of the rest of the domain. These two domains have different petrophysical properties.

Petrophysical properties

	Barrier zone	Drain zone
Porosity [-]	0.2	0.2
Permeability[m^2]	1.e-15	100.e-15

Table 1: Petrophysical Parameters

Phase properties

	Gas phase	Water phase
Viscosity [Pa.s]	0.0285e-03	0.571e-03

Table 2: Physical properties of fluids

Relative permeability model

We use a quadratic relative permeability model:

$$\text{Quadratic kr } kr(S) = S^2 \qquad \text{and} \qquad kr(S_w) = (1-S)^2$$

Boundary Conditions

The boundary conditions of the adapted SHPCO2 5 are presented in the following table 3:

	Pressure[Pa]	Composition
Injector 1	110.e+05	Water
Injector 2	105.e+05	Water
Productor	100.e+05	-

Table 3: Boundary condition parameters

It is to note that the Productor has the composition of what is produces. We consider the following mesh geometry which corresponds to the small (S) mesh size of the SHPCO2 benchmark.

Mesh	Dx [m]	Dy [m]	Nx	Ny	NCell
S	50	50	95	60	5700

Table 4: 2D mesh parameters

Well conditions

We detail the well conditions in the following table 5

5.2 Test cases

We propose two different test cases based on the SHPCO2 benchmark and it's reservoir configuration presented in 5.1. The test case 1 has constant initial saturation $S0$ while the test case 2 has more realistic initial saturations.

5.2.1 Test case 1

Database Generation

We launch simulations with a constant reservoir configuration except for three parameters, $S0$ the initial saturation, q_g the well injection flow rate and dt

Injection parameters	Flow rate $[m^2/s]$	Composition	well radius $[m]$	Opening period $[s]$
CO2 injector	q_g	Gas (S = 1.)	0.1	dt

Table 5: CO_2 injection well conditions

the time-step. We allow a maximum of 200 Newton iterations to converge. The convergence criterion is based on the residual norm and we iterate till $dt\|R\|_\infty \leq \epsilon$ with $\epsilon = 1e^{-6}$ and R the residual of the physical system.

We generate 5004 parameter combinations through a Latin Hypercube Sampling strategy [26] within the following ranges: $S0 \in [0, 0.6]$, $|q_g| \in [1e^{-5}, 1e^{-3}]m^2/s$ which corresponds to a well pressure $\in]10, 20]$ MPa and $dt \in [0.1, 10]$ years. Note that P_{imp} is obtained using the Implicit Pressure solver (IMP).

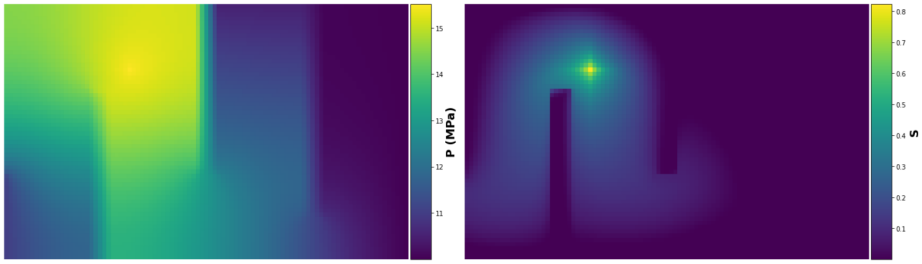

Figure 6: Test case 1 example of reservoir Pressure (left) and Saturation (right) obtained after a time-step with $S0 = 3.81e^{-4}$, $q_g = 7.61e^{-4}m^2/s$ and $dt = 2.4e^{+8}s$.

5.2.2 Test case 2

In the test case 1, we use constant reservoir saturation maps as initial saturation S0 and realised one well opening with a particular well injection flow rate q_g and time-step dt. In the test case 2, we use more realistic initial saturation maps. To do so, we realise N consecutive well opening and closure events (see figure 7. We use Latin Hypercube Sampling strategy to generate parameter combinations. The initial parameters are q_g, dt and S^n. After the first simulation, we close the well (i.e $q_g = 0m^2/s$) and launch another simulation using the previous reservoir state obtained and a new time-step dt (sampled through Latin Hypercube Sampling). Finally we open the well and launch a simulation with q_g and dt as parameters. The opening and closure step are repeated as many times as needed (see figure 7). The reservoir state is saved at every well opening or closure.

We generate data with $N = 9$ (i.e 5 well openings and 4 closures) and 3600 parameter combinations for each step. The parameters are sampled using a

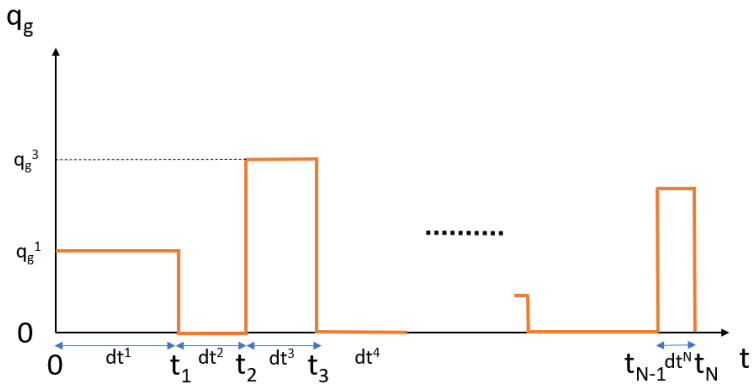

Figure 7: Test case 2 workflow with multiple well openings and closures. A step of time with a null well flow is realised between each closure and opening.

Latin Hypercube Sampling strategy within the following ranges: $S0 \in [0, 0.6]$, $q_g \in [-1e^{-5}, -1e^{-3}] m^2/s$ and $dt \in [1, 10]$ years in seconds.

We perform simulations for each parameter combination and we only consider data where the well injected flow rate q_g is not null. For $N = 9$, we have 5 wells openings (i.e $q_g > 0$) and 3600 parameter combinations for each one. Therefore, there is a total of 18000 samples. When splitting the samples in train and test sets, data coming from a same scenario are sent together in a set (i.e 5 per 5 for $N = 9$).

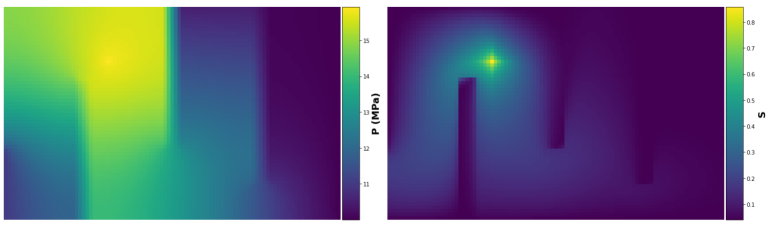

Figure 8: Example of reservoir Pressure (left) and Saturation (right) obtained after a time-step with $q_g = -9.8e^{-4} m^2/s$ and $dt = 3.1e^{+8} s$. The simulation required 12 Newton iterations to converge.

6 Results and discussion

6.1 Neural Network training

6.1.1 Test case 1

We use the Neural Network architecture presented in figure 2 with $Nc = 32$ channels in the Fourier layers and $\{q_g, dt, S^n\}$ as inputs (i.e Ni = 3). The input parameter dt is a scalar, therefore, we reshape it in a constant map of shape $(95, 60)$. Moreover, q_g is also a scalar. We reshape it in a (95,60) map which values are zeroes everywhere except at the well location where it takes q_g as value.

We split the data in a train and test sets with a 80/20 splitting ratio. We train the model on a NVIDIA V100 GPU during 27 hours using Adam optimizer, a batch size of 10, a learning rate of $5e^{-}5$, a momentum of 0.9, a weight decay of $1.e^{-4}$ and keep the model parameters corresponding to the lowest test loss value. We show the relative $L2$ loss evolution on the figure 9. The lowest test loss value is $1.9e^{-3}$ reached at epoch 17285. The corresponding train loss value is $2.0e^{-3}$.

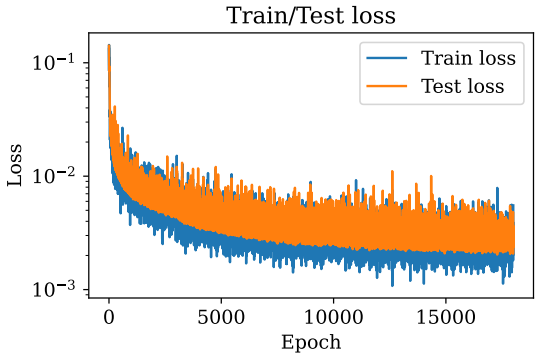

Figure 9: L2 loss evolution through epochs for test case 1. The lowest test loss value is $1.9e^{-3}$ reached at epoch 17285. The corresponding train loss value is $2.0e^{-3}$.

6.1.2 Test case 2

We use the Neural Network architecture presented in figure 2 with $Nc = 64$ channels in the Fourier layers. As the case is more complex than the test case 1, the neural network is harder. To alleviate this complexity, the implicit pressure is added to the input features . We therefore use $\{P_{imp}, q_g, dt, S^n\}$ as input features (i.e Ni = 4). The input parameter dt is a scalar, therefore, we reshape it in a constant map of shape $(95, 60)$. Moreover, q_g is also a scalar. We reshape it in a (95,60) map which values are zeroes everywhere except at the well location where it takes q_g as value. S^n and P_{imp} can be used straightforward.

We split the data in a train and test set with a 80/20 splitting ratio. We train the model on a NVIDIA V100 GPU during 132 hours using Adam optimizer, a batch size of 128, a momentum of 0.9, a weight decay of $1.e^{-4}$ and keep the model parameters corresponding to the lowest test loss value. We start with a learning rate of $1.e^{-4}$, at iteration number 1000, we decrease it to $5.e^{-5}$, then $1.e^{-5}$ at iteration number 1600 and finally we set the learning rate to $5.e^{-7}$ at iteration number 3600 and until the end.

We show the $L2$ loss evolution in the figure 10. The lowest test loss value is 8.7^{-4} reached at epoch 7295. The corresponding train loss value is $9.3e^{-4}$.

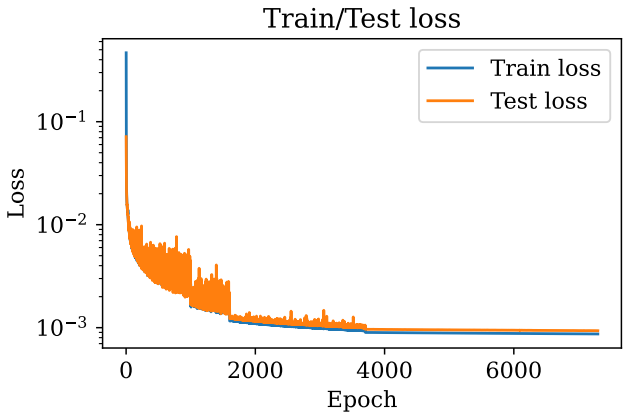

Figure 10: L2 loss evolution through epochs for the test case 2. The lowest test loss value is 8.7^{-4} reached at epoch 7295. The corresponding train loss value is $9.3e^{-4}$. We start with a learning rate of $1.e^{-4}$, at iteration number 1000, we change it to $5.e^{-5}$, then $1.e^{-5}$ at iteration number 1600 and finally we set the learning rate to $5.e^{-7}$ at iteration number 3600 and until the end.

6.2 Results

In this section, we present the results obtain using the hybrid methodology, first on the scenario presented in the figure 1 and then on the two tests cases. We compare the performances in term of Newton iterations obtained between the standard and hybrid approaches.

6.2.1 Single prediction example

We apply the hybrid methodology on the example scenario presented in figure 1 only during the well event, this implies that every step outside of the well event requires the same number of Newton iterations for the standard and hybrid approaches.

We observe a significant reduction in the number of Newton iterations during the

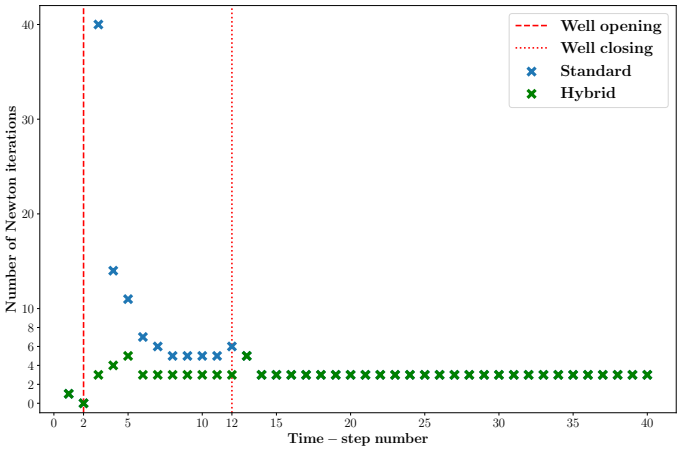

Figure 11: Hybrid Newton approach applied to the example scenario and it's impact on Newton's method.

well event. Immediately after the well opening, the standard approach (depicted by blue crosses) requires 40 iterations, whereas the hybrid approach (represented by green crosses) only necessitates 3 iterations. Furthermore, throughout the entire well event, the hybrid approach consistently requires fewer Newton iterations compared to the standard approach, resulting in a 68% reduction in Newton iterations during the well event and a 38% reduction across the entire scenario.

6.2.2 Test case 1

We launch simulations with the same parameters combinations of test case 1, $X_{init}^{n+1} = (P_{imp}, S^n)$ and $X_{ML}^{n+1} = (P_{imp}, S_{ML})$ as initial guesses. P_{imp} is calculated using the Implicit Pressure Solver and S_{pred} using the model obtained in the previous section. The results are presented in figure 12.

We observe that the hybrid methodology facilitates Newton's method initialization directly within the domain of quadratic convergence, resulting in a maximum of 5 Newton iterations for the training set and 4 for the test set. This is particularly intriguing as it appears to scale with problem complexity. In essence, the more challenging the problem using the standard methodology, the greater the potential benefit from employing the hybrid methodology. With the hybrid formulation, we achieve an average speedup of 54%, translating to 54% fewer Newton iterations compared to the standard methodology for the training set and a 53% reduction for the test set.

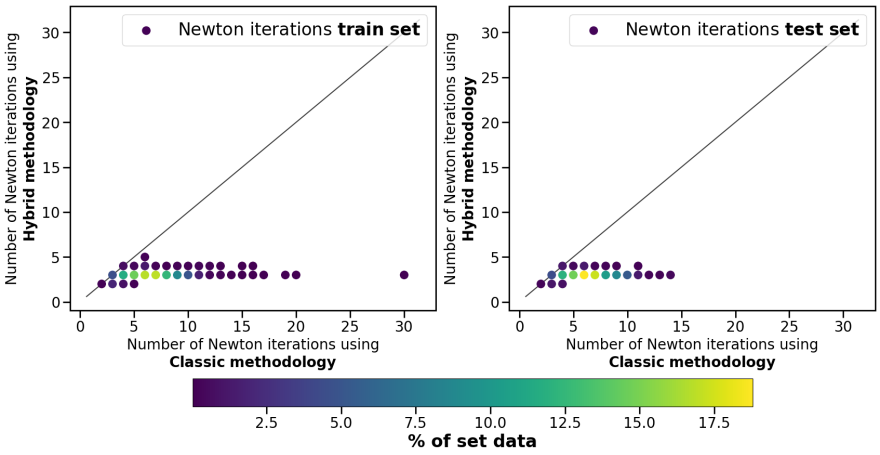

Figure 12: Test case 1 scatter plot of the number of Newton iterations needed to converge using standard methodology versus using hybrid methodology on the train set (left figure) and on the test set (right figure). The color bar shows the distribution of Newton iterations using standard and Hybrid methodologies for the train and test set respectively.

6.2.3 Test case 2

We launch simulations with the same parameters combinations of test case 2, $X_{init}^{n+1} = (P_{imp}, S^n)$ and $X_{ML}^{n+1} = (P_{imp}, S_{ML})$ as initial guesses. P_{imp} is calculated using the Implicit Pressure Solver and S_{ML} using the model obtained in the previous section.

Considering every simulations and using the hybrid methodology, we speed up by 39% , i.e 39% less Newton iterations than the standard methodology the computations for the training set and by 38.7% for the test set.

6.3 Discussion

Considering a wide range of injection scenarios, we show that it is possible to learn the impact of a well event on a reservoir. We speed up by 53% the handling of well events for the test case 1 and by 38% for the test case 2. Moreover the hybrid Newton methodology is quite general and can be applied to any problem that requires an important number of costly iterations. Finally, we observe that the more challenging the problem is using the standard methodology, the greater the potential benefit from employing the hybrid methodology. However, there are some limiting issues that needs to be considered.

Constant discretization

We use a specific discretization (SHPCO2 S mesh) for the data generation and we predict on the same discretization. This implies that the model would not

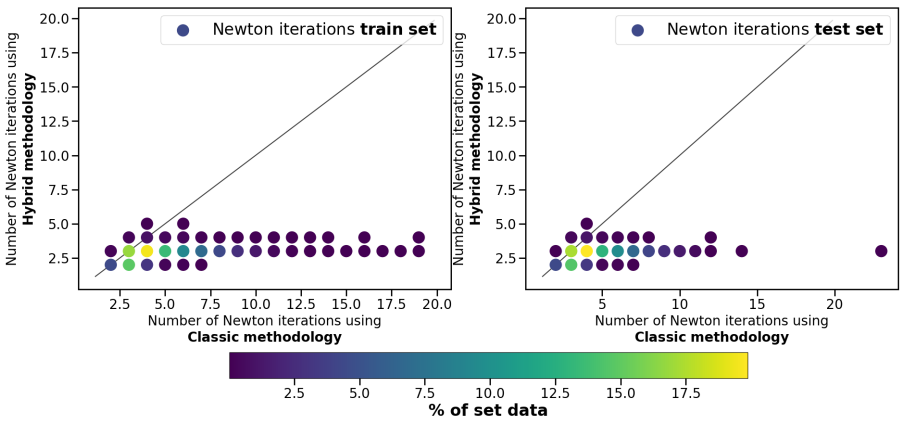

Figure 13: Test case 2 scatter plot of the number of Newton iterations needed to converge using standard methodology versus using hybrid methodology on the train set (left figure) and on the test set (right figure). The color bar shows the distribution of Newton iterations using standard and Hybrid methodologies for the train and test set respectively.

work for different meshes. A new model has to be generated. However, the idea of a model invariant to discretization is developed through Neural Operators [13]. A model could be trained using data from different discretizations and predict the solution on multiple discretizations.

Constant well position

The methodology is applied on a constant grid with a constant well position. While the pressure variations are global during a well event, the saturation variations are local. Therefore, if we change the well position, the model prediction will not be accurate. To alleviate this issue, a local approach could be used, i.e create a model that predicts the saturation only around a well.

7 Conclusion

We proposed in this article a methodology to alleviate the impact of well events during the numerical simulation of CO_2 storage in the subsurface. We complement the standard numerical algorithm by predicting an initialization of Newton's method directly in the domain of convergence using a supervised learning approach based on recently developed Fourier Neural Operators. Our results show a significant decrease in the number of Newton iterations required for convergence, while ensuring the convergence to the correct solution. Moreover the hybrid Newton methodology is quite general and can be applied to any problem that requires an important number of costly iterations. Finally, we observe that the more challenging the problem is using the standard methodology, the

greater the potential benefit from employing the hybrid methodology.

References

[1] E. Ahmed, Ø. Klemetsdal, X. Raynaud, O. Møyner, and H. M. Nilsen. Adaptive Timestepping, Linearization, and A Posteriori Error Control for Multiphase Flow of Immiscible Fluids in Porous Media with Wells. *SPE Journal*, pages 1–21, 10 2022.

[2] Shohreh Amini and Shahab Mohaghegh. Application of machine learning and artificial intelligence in proxy modeling for fluid flow in porous media. *Fluids*, 4(3), 2019.

[3] Subhrajyoti Bhattacharyya and Aditya Vyas. A novel methodology for fast reservoir simulation of single-phase gas reservoirs using machine learning. *Heliyon*, 8(12):e12067, 2022.

[4] Bailian Chen, Dylan R. Harp, Youzuo Lin, Elizabeth H. Keating, and Rajesh J. Pawar. Geologic CO2 sequestration monitoring design: A machine learning and uncertainty quantification based approach. *Applied Energy*, 225(C):332–345, 2018.

[5] Andrew K. Chu, Sally M. Benson, and Gege Wen. Deep-learning-based flow prediction for co2 storage in shalendash;sandstone formations. *Energies*, 16(1), 2023.

[6] Robert Eymard, Gallouet Thierry, and Raphaèle Herbin. Finite volume methods. 7, 12 2000.

[7] Olga Fuks and Hamdi Tchelepi. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. 07 2020.

[8] Florian Haeberlein. *Time Space Domain Decomposition Methods for Reactive Transport — Application to CO2 Geological Storage*. Theses, Université Paris-Nord - Paris XIII, October 2011.

[9] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2016.

[10] Jianguo Huang, Haoqin Wang, and Haizhao Yang. Int-deep: A deep learning initialized iterative method for nonlinear problems. *Journal of Computational Physics*, 419:109675, 2020.

[11] M. King Hubbert. Darcy's Law and the Field Equations of the Flow of Underground Fluids. *Transactions of the AIME*, 207(01):222–239, 12 1956.

[12] Zhongyi Jiang, Min Zhu, Dongzhuo Li, Qiuzi Li, Yanhua O. Yuan, and Lu Lu. Fourier-mionet: Fourier-enhanced multiple-input neural operators for multiphase modeling of geological carbon sequestration, 2023.

[13] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces, 2021.

[14] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[15] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[16] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *CoRR*, abs/2010.08895, 2020.

[17] Zongyi Li, Hongkai Zheng, Nikola B. Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *CoRR*, abs/2111.03794, 2021.

[18] Mingliang Liu, Divakar Vashisth, Dario Grana, and Tapan Mukerji. Joint inversion of geophysical data for geologic carbon sequestration monitoring: A differentiable physics-informed neural network model. *Journal of Geophysical Research: Solid Earth*, 128, 03 2023.

[19] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *CoRR*, abs/1910.03193, 2019.

[20] Alban Odot, Ryadh Haferssas, and Stéphane Cotin. Deepphysics: a physics aware deep learning framework for real-time simulation. *CoRR*, abs/2109.09491, 2021.

[21] Daniel Asante Otchere, Tarek Omar Arbi Ganat, Raoof Gholami, and Syahrir Ridha. Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ann and svm models. *Journal of Petroleum Science and Engineering*, 200:108182, 2021.

[22] 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*, 378:686–707, 2019.

[23] Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes, 2023.

[24] J. W. Sheldon and W. T. Jr. Cardwell. One-dimensional, incompressible, noncapillary, two-phase fluid flow in a porous medium. *Transactions of the AIME*, 216:290–296, 1959.

[25] Parisa Shokouhi, Vikas Kumar, Sumedha Prathipati, Seyyed A. Hosseini, Clyde Lee Giles, and Daniel Kifer. Physics-informed deep learning for prediction of co2 storage site response. *Journal of Contaminant Hydrology*, 241:103835, 2021.

[26] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[27] Alexander Y. Sun. Optimal carbon storage reservoir management through deep reinforcement learning. *Applied Energy*, 278:115660, 2020.

[28] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 1596–1611. Curran Associates, Inc., 2022.

[29] Meng Tang, Xin Ju, and Louis J. Durlofsky. Deep-learning-based coupled flow-geomechanics surrogate model for co2 sequestration. *International Journal of Greenhouse Gas Control*, 118:103692, 2022.

[30] Zeeshan Tariq, Ertugrul Umut Yildirim, Manojkumar Gudala, Bicheng Yan, Shuyu Sun, and Hussein Hoteit. Spatial–temporal prediction of minerals dissolution and precipitation using deep learning techniques: An implication to geological carbon sequestration. *Fuel*, 341:127677, 2023.

[31] Hung Vo Thanh and Kang-Kun Lee. Application of machine learning to predict co2 trapping performance in deep saline aquifers. *Energy*, 239:122457, 2022.

[32] Hung Vo Thanh, Qamar Yasin, Watheq J. Al-Mudhafar, and Kang-Kun Lee. Knowledge-based machine learning techniques for accurate prediction of co2 storage performance in underground saline aquifers. *Applied Energy*, 314:118985, 2022.

[33] Gege Wen, Catherine Hay, and Sally M. Benson. Ccsnet: A deep learning modeling suite for co2 storage. *Advances in Water Resources*, 155:104009, 2021.

[34] Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

[35] Gege Wen, Zongyi Li, Qirui Long, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. Real-time high-resolution co2 geological storage prediction using nested fourier neural operators. *Energy Environ. Sci.*, 16:1732–1741, 2023.

[36] Philipp A Witte, Russell Hewett, and Ranveer Chandra. Industry-scale co2 flow simulations with model-parallel fourier neural operators. In *NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*, 2022.

[37] Philipp A. Witte, Tugrul Konuk, Erik Skjetne, and Ranveer Chandra. Fast co2 saturation simulations on large-scale geomodels with artificial intelligence-based wavelet neural operators. *International Journal of Greenhouse Gas Control*, 126:103880, 2023.

[38] Hao Wu, Nicholas Lubbers, Hari S. Viswanathan, and Ryan M. Pollyea. A multi-dimensional parametric study of variability in multi-phase flow dynamics during geologic co2 sequestration accelerated with machine learning. *Applied Energy*, 287:116580, 2021.

[39] Ichrak Ben Yahia, Jean-Pierre Merlet, and Yves Papegay. Mixing neural networks and the Newton method for the kinematics of simple cable-driven parallel robots with sagging cables. In *ICAR 2021 - 20th International Conference on advanced robotics*, Ljulbjana, Slovenia, December 2021.

[40] Bicheng Yan, Bailian Chen, Dylan Robert Harp, Wei Jia, and Rajesh J. Pawar. A robust deep learning workflow to predict multiphase flow behavior during geological co2 sequestration injection and post-injection periods. *Journal of Hydrology*, 607:127542, 2022.

[41] Junyu You, William Ampomah, and Qian Sun. Development and application of a machine learning based multi-objective optimization workflow for co2-eor projects. *Fuel*, 264:116758, 2020.