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

June 7, 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 system. 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.

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. We apply our methodology to two test cases derived from a SHPCO_2 benchmark.

Contents

2 Introduction

Context

The reduction of CO_2 emission 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 exploit a potential carbon storage site and to monitor the 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.

Problem

More specifically, well events (opening and closure) cause important numerical difficulties due to their instant impact on the system. 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.

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", where machine learning models either replace or complement traditional numerical algorithms while preserving the inherent constraints from the physical model. These models can be trained in a supervised or unsupervised manner. In supervised learning, the objective is to match the labeled data available from experiment or previous simulations. In unsupervised learning, no labeled data is available and the objective is to directly enforce physical constraints, e.g. by minimizing the residual of the partial differential equations describing the evolution of the solution, by penalizing deviations to mass conservation, etc.

Contribution

We propose a methodology to alleviate the impact of well events during the numerical simulation of $C0_2$ 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

We first formulate the continuous model and it's numerical resolution associated to CO_2 storage in the underground. We then apply our methodology on two test cases based on the SHPCO2 benchmark. Finally we discuss the obtained results and the next steps that should be done.

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. The standard approach of modeling a two-phase flow in a porous medium is based on the application of Darcy's law for each phase $\alpha \in \{w, g\}$ with dimensionless factors kr_α in front of the tensor of permeability $\overline{\mathrm{K}}$, called relative permeabilities of the phases. We consider the medium as isotropic. Therefore, 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. A well can be modeled by a source term in the conservation equation. For incompressible fluids, the equations describing the conservation of pore volume and phase volume, the conservation of momentum, the capillary forces between the two phases are written:

$$\phi \frac{\partial}{\partial t}(S_w) + div(v_w) = q_w, \quad \phi \frac{\partial}{\partial t}(S_g) + div(v_g) = q_g, \tag{1}$$

$$v_w = -\frac{K kr_w(S_w)}{\mu_w} \nabla P_w, \quad v_g = -\frac{K kr_g(S_g)}{\mu_g} \nabla P_g, \tag{2}$$

$$S_g + S_w = 1, \tag{3}$$

$$P_g - P_w = P_{c_{w,g}}(S_w) \tag{4}$$

Where S_α is the phase saturation, v_α the Darcy flow velocity of the phase in the porous medium, μ_α the phase viscosity, P_α the phase pressure, $P_{c_{w,g}}$ the capillary pressure and ϕ the medium porosity. The relative permeabilities kr_α are increasing functions of the saturation only. q_w, q_g are respectively the water flow and gas flow injected or produced in the wells.

We suppose that the permeability K and the porosity ϕ vary in space but are constant through time. The viscosities μ_α are constant. Moreover, we neglect the capillary forces between the two phases leading to rewrite equation 4 as 5.

$$P_g = P_w = P \tag{5}$$

The unknowns are thus the common pressure P and the saturation in gas denoted by S.

3.2 Numerical resolution

The continuous model is discretized using a two-points finite volume scheme and a Euler type time integration. The pressure and saturation are solved simultaneously through a fully implicit scheme [2][9]. Finally, the resulting non-linear system of equations is solved using Newton's method. This last is initialized using the solution obtained at the previous step X^n as an initial guess and returns the solution at the next step X^{n+1}.

Time-step management

In theory, the time-step size of a fully implicit reservoir simulator is not limited by stability (i.e unconditionally stable). However, in practice, when using the standard Newton's method, convergence may fail for larger time-step sizes. This necessitates multiple time-step reductions to achieve convergence, resulting in a significant number of superfluous iterations.

At the scale of a single time-step (in opposition with the global simulation scale), the time-step is driven by Newton's method. One can allow a maximum number of Newton iterations N_{max} under which the method must converge. Above this number, we start over with a smaller time-step, usually by dividing by a factor two. There are other time-step management mechanisms at the global simulation scale but they do not concern us in this study.

3.3 Impact of well events during the numerical resolution

Well events generate pressure and saturation discontinuities 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 can prevent Newton's method from converging while attempting to solve the system as the initial guess may be far from the solution. In this case, we try to solve the system again with a lower time step and repeat till convergence. Well events can therefore contribute to a large percentage of the actual simulation time.

4 Methodology

We propose a methodology based on the Hybrid Newton algorithm [11] [6] [15] to alleviate the impact of well events. It consists in predicting via machine learning an initialization closer to the solution than the standard initialization. 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 comparable to Newton algorithm except for the initialisation. Indeed, the hybrid method proposes to initialize with a more accurate solution instead of using the previous step solution. By using an initialisation closer to the solution, we aim to converge in fewer iterations. We compare the residuals of the standard initialization X_0^{n+1} and the hybrid initialization X_{pred}^{n+1} and select the one which is closer to the solution.

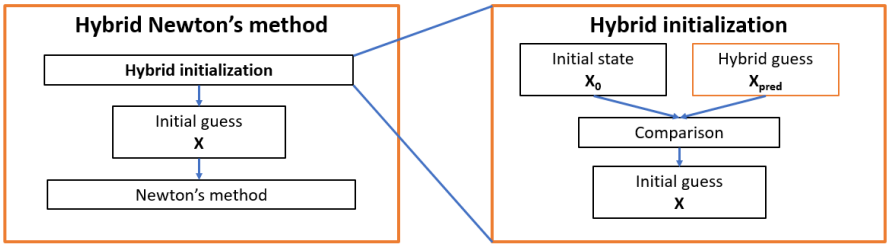

Figure 1: Hybrid Newton's method schematic. The method is composed of two parts (left figure), a hybrid initialization and the well-known Newton's method. The hybrid initialization (right figure) compares the initial guess, usually the solution at the previous time step and a hybrid guess.

In this article, we focus on the hybrid guess construction and performances compared to the standard initial guess performances. We will compare them through their impact on the number of Newton iteration and not by their residuals.

4.2 Initial guess construction

In the hybrid Newton algorithm, a prediction guess $X_{pred}^{n+1} := (P_{pred}, S_{pred})$ is required to be compared in term of residuals with the initial guess $X_0^{n+1} := (P^n, S^n)$. 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) [13] as a prediction guess P_{pred}. The implicit pressure solver solves the linear elliptic equation (6) 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.

$$div(v(P_{imp}, S^n)) = 0 \tag{6}$$

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 standard methodology uses the initial guess $X_0^{n+1} = X^n = (P^n, S^n)$ and the hybrid methodology uses the initial guess $X_{pred}^{n+1} = (P_{imp}, S_{pred})$. 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 $X_0^{n+1} = (P_{imp}, S^n)$

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. We also know the previous reservoir state at $X^n = (P_{imp}, S^n)$.

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 [3], [8], [12]. The Fourier Neural Operator (FNO) [10] 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) [5] 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.

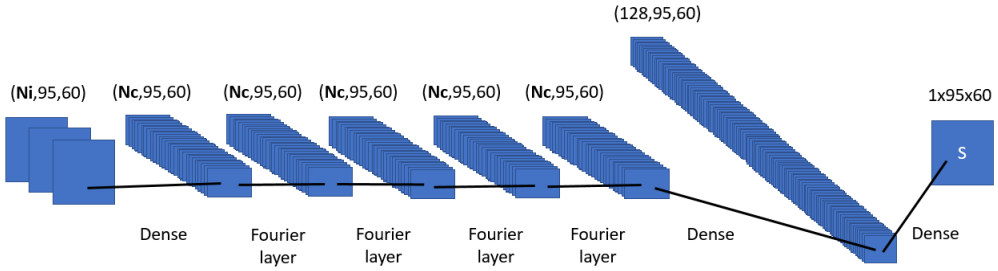

Figure 2: Selected neural network architecture composed by an uplifting dense layer, four Fourier layers, a dense layer and a projection dense layer. **Nc** and **Ni** are parameters respectively representing the number of channels and the number of inputs.

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.

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.

5 Test case

5.1 SHPC02 benchmark

Context

As a practical use case, we use the SHPCO2 [4] benchmark to test our methods.

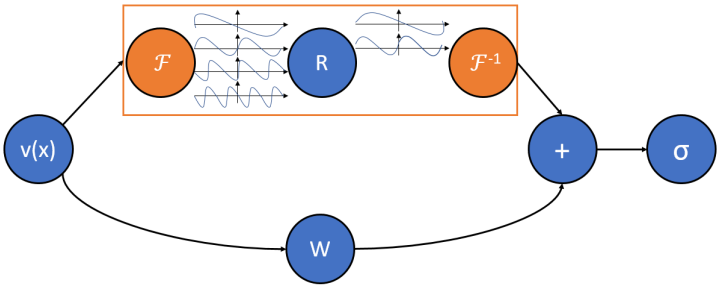

Figure 3: The Fourier layer starts with an input vector v, applies the Fourier transform $\mathscr{F}$ to it, then performs a linear transformation R on the lower Fourier modes while filtering out the higher modes. The inverse Fourier transform $\mathscr{F}^{-1}$ is then applied. A local linear transformation 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.

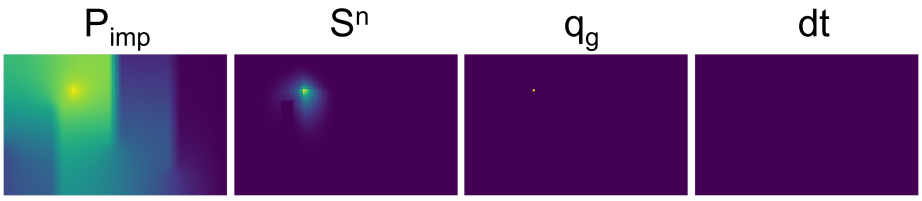

Figure 4: Qualitative view of Neural network input feature possible assembly.

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.

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

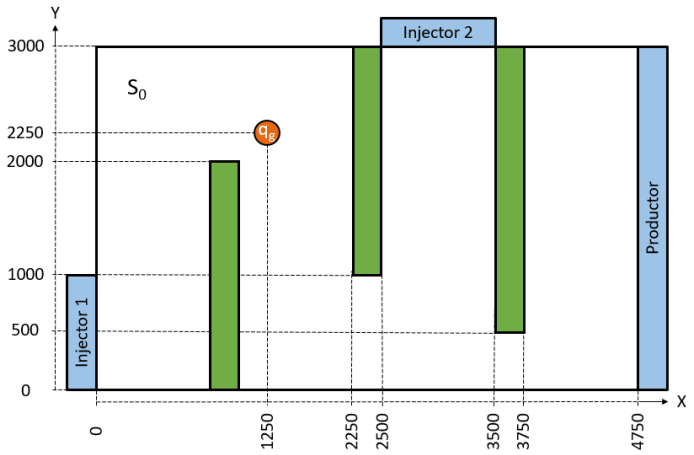

Figure 5: Adapted 2D SHPCO2 case geometry

"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 SHP CO_2 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

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

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

Case construction

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 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 [14] 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).

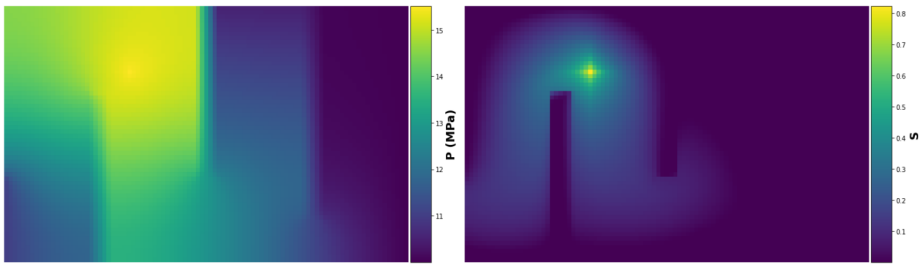

Figure 6: Test case 1 example of reservoir Pressure (left) and Saturation (right) obtained after a time-step with $S0 = 3.81e^{-4}$, $q_g = 7.61e^{-4}m^2/s$ and $dt = 2.4e^{+8}s$.

Numerical performances using standard method

For each combination of parameters, we plot on the figure 7 the number of Newton iterations needed to converge for each parameter combinations (q_g, dt). Even though the initial saturation $S0$ has impact on the number of newton iterations, we do not plot it here for clarity purpose. We observe the the number of Newton iterations needed to converge increases with the well injection flow rate and the time-step and that there is a maximum of 30 Newton iterations required to converge.

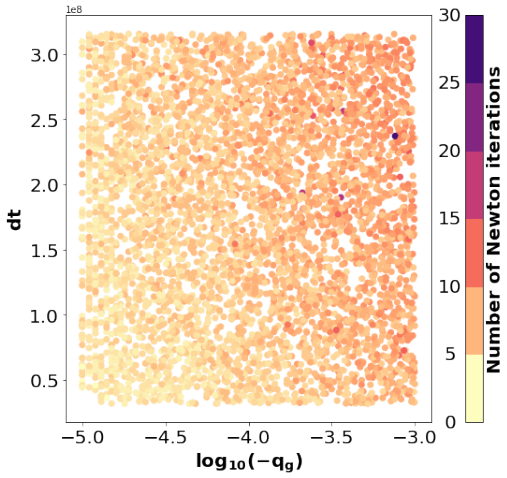

Figure 7: Distribution of test case 1 Newton iterations considering well event parameters q_g the well injection flow and dt the time-step in seconds. The maximum number of Newton iterations needed to converge is 30.

Neural Network training

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 $L2$ loss evolution on the figure 8. The lowest test loss value is $2.02e^{-2}$ reached at epoch 17285. The corresponding train loss value is $1.91e^{-2}$.

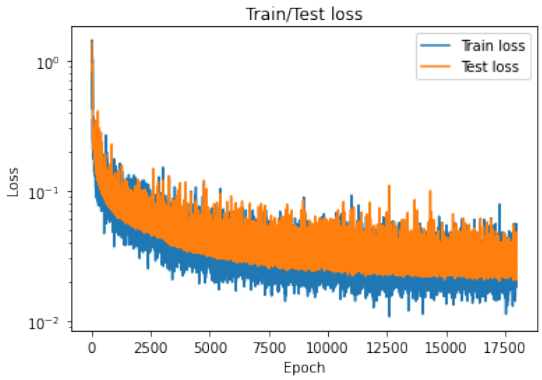

Figure 8: L2 loss evolution through epochs for test case 1. The lowest test loss value is $2.02e^{-2}$ reached at epoch 17285. The corresponding train loss value is $1.91e^{-2}$.

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 9. 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 9). 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

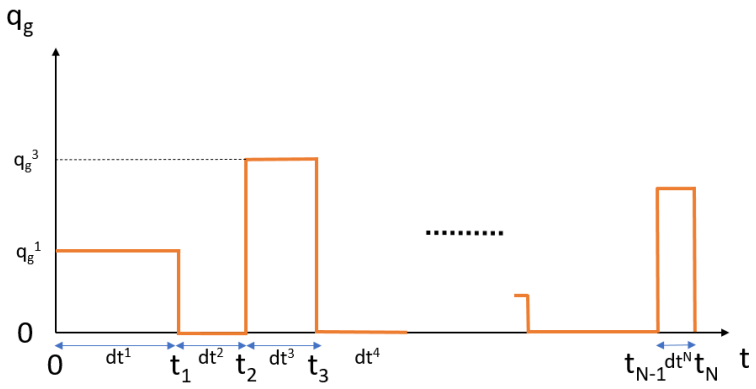

Figure 9: Test case 2 workflow with multiple well openings and closures. A step of time with a null well flow is realised between each closure and opening.

parameter combinations for each step. The parameters are sampled using a 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 where the well injected flow q_g is not null as input data for the neural network training. 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 $N = 9$).

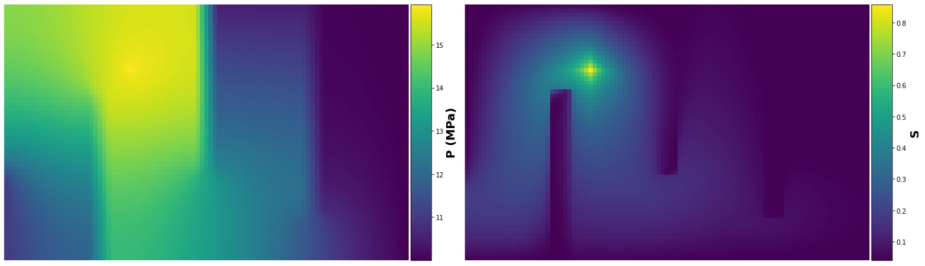

Figure 10: Example of reservoir Pressure (left) and Saturation (right) obtained after a time-step with $q_g = -9.8e^{-4}m^2/s$ and $dt = 3.1e^{+8}s$. The simulation required 12 Newton iterations to converge.

Numerical performances using standard method

For each combination of parameters, we plot on the figure 11 the number of Newton iteration needed to converge

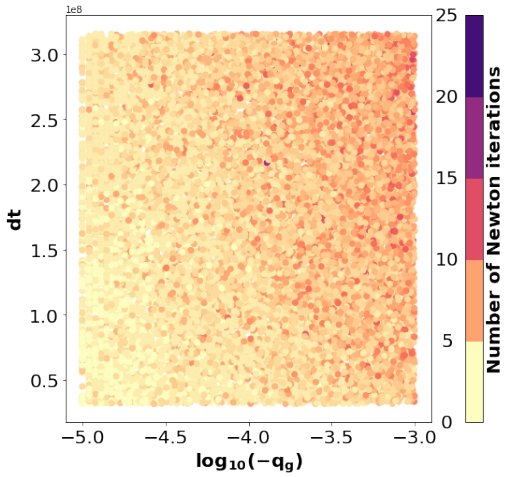

Figure 11: Test case 2 distribution of Newton iterations considering well event parameters q_g the well injection flow and dt the time-step.

Neural Network training

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 12. The lowest test loss value is $1.16e^{-1}$ reached at epoch 7295. The corresponding train loss value is 1.11^{-1}.

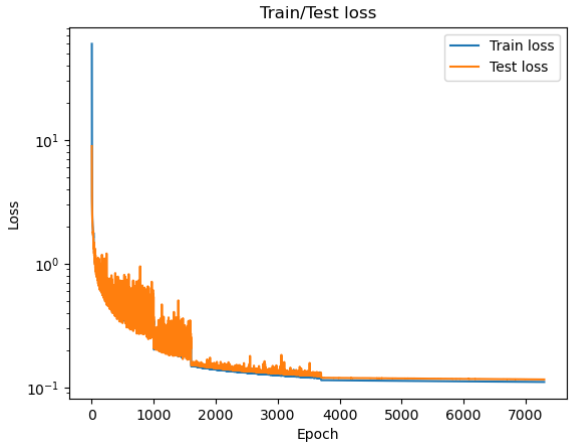

Figure 12: L2 loss evolution through epochs for the test case 2. The lowest test loss value is 1.16^{-1} reached at epoch 7295. The corresponding train loss value is $1.11e^{-1}$. We start with a learning rate of $1.e^{-4}$, at iteration number 1000, we change it to $5.e^{-5}$, then $1.e^{-5}$ at iteration number 1600 and finally we set the learning rate to $5.e^{-7}$ at iteration number 3600 and until the end.

6 Results and discussion

6.1 Results

We launch simulations following the methodology and compare the number of Newton iterations with the reference case.

6.1.1 Test case 1

We launch simulations with the same parameters combinations of test case 1, $X_0^{n+1} = (P_{imp}, S^n)$ and $X_{pred}^{n+1} = (P_{imp}, S_{pred})$ 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 13.

We observe that the hybrid methodology enables an initialization of Newton's method directly in the domain of quadratic convergence, leading to a maximum number of Newton iterations of 5 for the training set and 4 for the test set. Using the hybrid formulation, the average speed up is 54% , i.e 54% less Newton iterations than the standard methodology for the training set and by 53% for the test set.

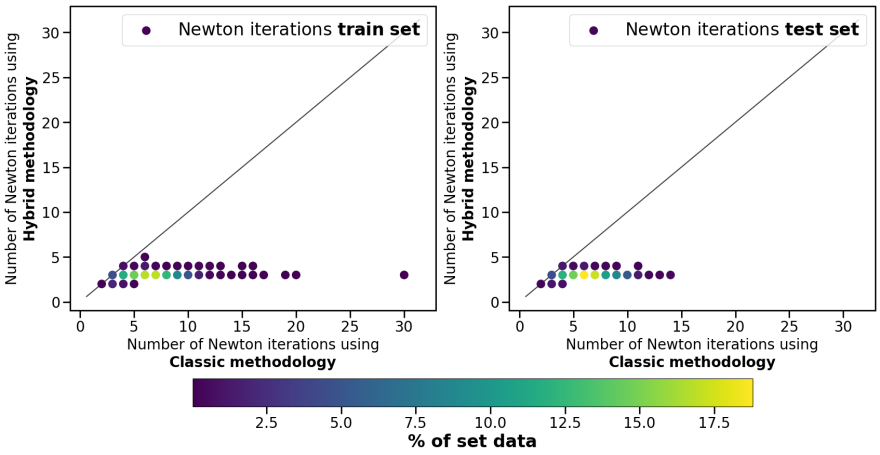

Figure 13: Test case 1 scatter plot of the number of Newton iterations needed to converge using standard methodology versus using hybrid methodology on the train set (left figure) and on the test set (right figure). The color bar shows the distribution of Newton iterations using standard and Hybrid methodologies for the train and test set respectively.

6.1.2 Test case 2

We launch simulations with the same parameters combinations of test case 2, $X_0^{n+1} = (P_{imp}, S^n)$ and $X_{pred}^{n+1} = (P_{imp}, S_{pred})$ as initial guesses. P_{imp} is calculated using the Implicit Pressure Solver and S_{pred} 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.

6.2 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. 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 work for different meshes. A new model has to be generated. However, the idea of a model invariant to discretization is being more and more developed through

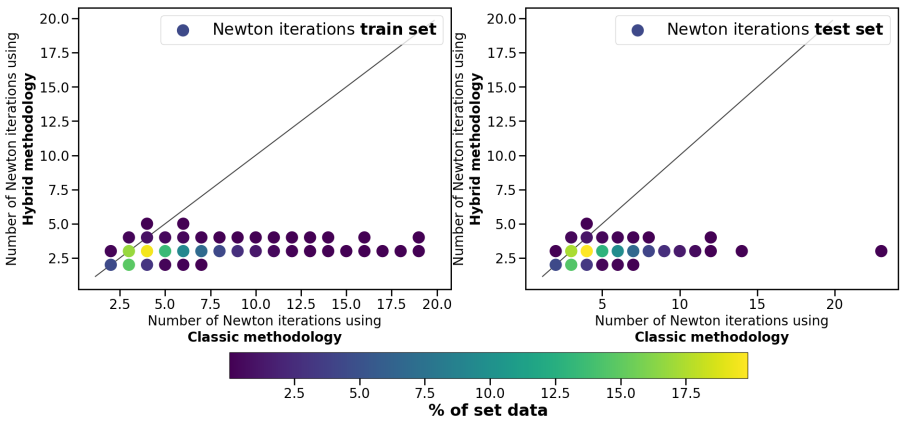

Figure 14: Test case 2 scatter plot of the number of Newton iterations needed to converge using standard methodology versus using hybrid methodology on the train set (left figure) and on the test set (right figure). The color bar shows the distribution of Newton iterations using standard and Hybrid methodologies for the train and test set respectively.

Neural Operators [7]. 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.

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

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

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

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

[6] 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.

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

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.