Approximation of Subsurface Flow via Physics Informed Neural Netwokrs

20 February, 2025

Key idea of the research

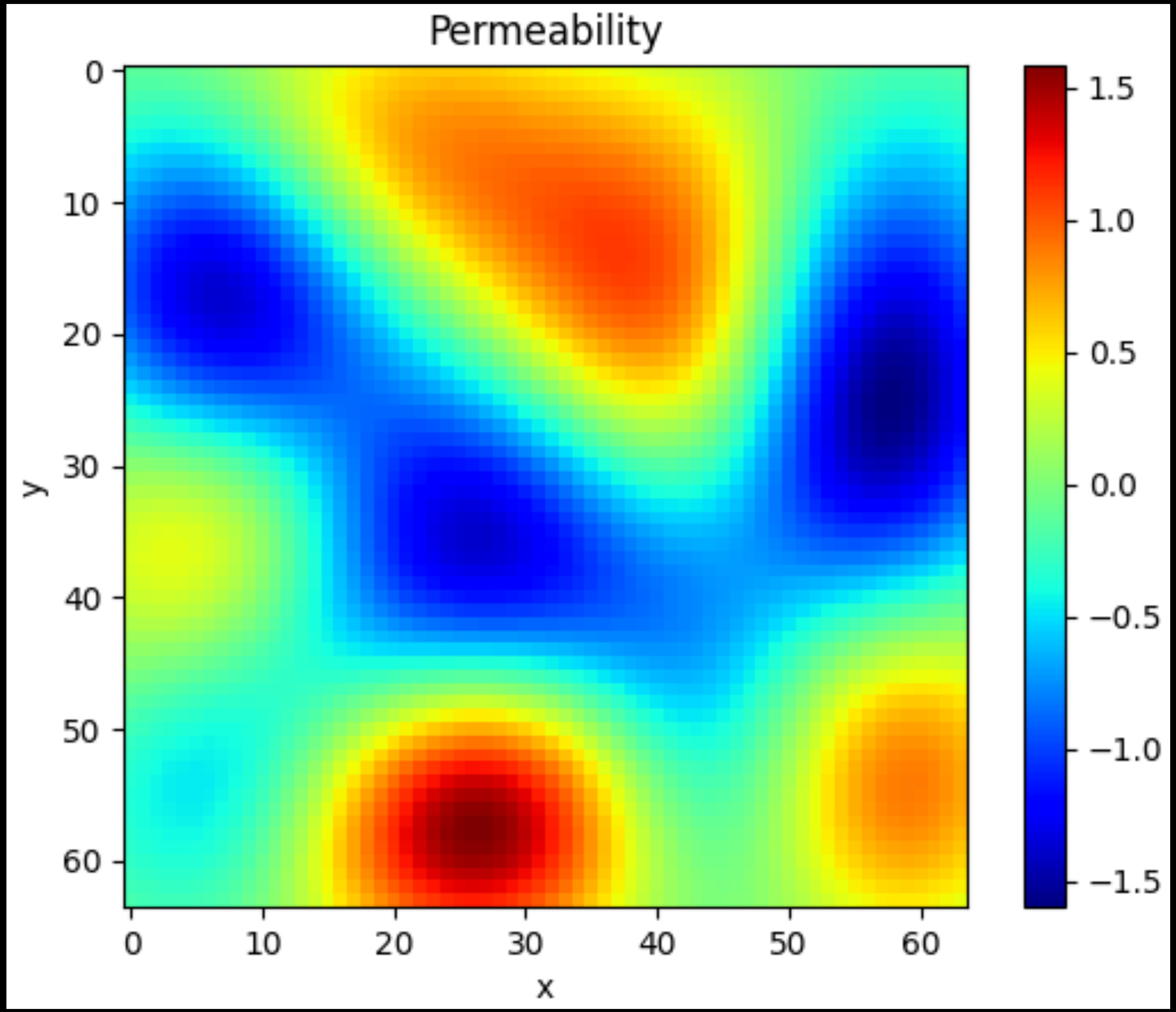
Surrogate models of complicated physical systems are routinely utilized in engineering problems. For instance, Uncertainty Quantification in hydrocarbons production or subsurface CO2 storage has an extremely high computational cost that can be reduced dramatically by utilization of cheap proxy models. Huge variety of Machine-Learning models have been iplemented to develop a proxy model starting from Polynomial functions and Decision Trees and ending by Artificial Neural Networks.

We show experimentally that accurate Neural Network can be trained on relatively small dataset. Moreover, we demonstrate that such Physics Informed Neural Networks last longer in comparision with such methods as Polynomial Chaos Expansion and Decision Trees when the complexity of the systems increases and if trained on the same dataset. In other words Physics Informed Neural Networks can approximate systems with high degree of nonlinearity even when limited number of samples is available.

Dataset

- We have simulator that gives approximation of the reservoir. To build system of oil-water displacement we need define porosity and permeability. And as a result we get meshes of velocities, saturations and pressures depending on time. Here, some examples of simulations.
- For the model training time was sampling on segment [0, 3], the coordinates if the surface belongs to [0, 1] x [0, 1].
- We generate point in two dimensional surface, but we will add nonlinearity to the system via input parameters of reservoir. This parameters will define permeability field, and therefore model should train its parameters to define which system is considered.

Permeability field



- We define a Gaussian covariance kernel $\exp\left(-\frac{1}{2}\frac{\|x_i - x_j\|^2}{l_{\mathrm{cor}}^2}\right)$ on a set of points x. This yields a covariance C, which we diagonalize as $C = Q\Lambda Q^T$.
- Then we multiply each eigenvector q_i by $\sqrt{\lambda_i}$ and summing these with random Gaussian coefficients.
- Truncating to the top few eigenvalues approximates the random field with fewer components while maintaining most of the variance.
- Finally, this linear combination produces a realization of a Gaussian field consistent with the given covariance structure.

Physics-Infromed Neural Networks

- Regression problem: learn output y from features matrix X. We consider that X is not coordinates of the point, but X can include parameters of reservoir: permeability, porosity.
- The model aimed to learn physical characteristic of the porous media, where two dimensional oil-water displacement problem is considered. Therefore we use Darcy law to define Loss function for the training.
- As an output PINN should return vectors of saturation, fluid flow rate for oil and water, pressure and pressure gradient

$$u_a(t,x) = -\frac{k(x)k_a\left(s(t,x)\right)}{\mu_a}\nabla P(t,x)$$

$$\begin{cases} P(t,(0,x_2)) = (1,1), \\ P(t,(1,x_2)) = (0,0), \\ s_{water}(0,x) = (0,0), \\ s_{oil}(0,x) = (1,1), \\ u^{x_2}_{water}(t,x)\Big|_{x_2=0} = u^{x_2}_{water}(t,x)\Big|_{x_2=1} = 0, \\ u^{x_2}_{oil}(t,x)\Big|_{x_2=0} = u^{x_2}_{oil}(t,x)\Big|_{x_2=1} = 0. \end{cases}$$

$$\begin{cases} r_1(t,x) = \phi(x)\frac{\partial s_{water}(t,x)}{\partial t} + \sum_i \frac{\partial u_{water,i}(t,x)}{\partial x_i} = 0, \\ r_2(t,x) = \phi(x)\frac{\partial s_{oil}(t,x)}{\partial t} + \sum_i \frac{\partial u_{oil,i}(t,x)}{\partial x_i} = 0, \\ r_3(t,x) = u_{water,i} + \frac{kk_{water}(s(t,x))}{\mu_{water}}\frac{\partial P(t,x)}{\partial x_i} = 0, \\ r_4(t,x) = u_{oil,i} + \frac{kk_{oil}(s(t,x))}{\mu_{oil}}\frac{\partial P(t,x)}{\partial x_i} = 0, \\ r_5(t,x) = s_{water}(t,x) + s_{oil}(t,x) - 1 = 0. \end{cases}$$

Physics-Infromed Neural Networks

- As we know all the boundary and initial condition and , we use loss functions as follows:

$$\mathcal{R}_0(t,x) = r_1(t,x)^2 + r_2(t,x)^2 + r_3(t,x)^2 + r_4(t,x)^2 + r_5(t,x)^2$$

$$\mathcal{R}_1(t,x) = (p(t,0) - 1.0)^2 + (p(t,1) - 0.0)^2 + (s_{oil}(0,x) - 1.0)^2 + (s_{water}(0,x) - 0.0)^2$$

$$\begin{aligned}\mathcal{R}_2(t,x) = &(p_{neural\ network}(t,x) - p_{simulation}(t,x))^2 \\ &+ (s_{water,neural\ network}(t,x) - s_{water,simulation}(t,x))^2 \\ &+ (s_{oil,neural\ network}(t,x) - s_{oil,simulation}(t,x))^2 \\ &+ |u_{water,neural\ network}(t,x) - u_{water,simulation}(t,x)|^2 \\ &+ |u_{oil,neural\ network}(t,x) - u_{oil,simulation}(t,x)|^2\end{aligned}$$

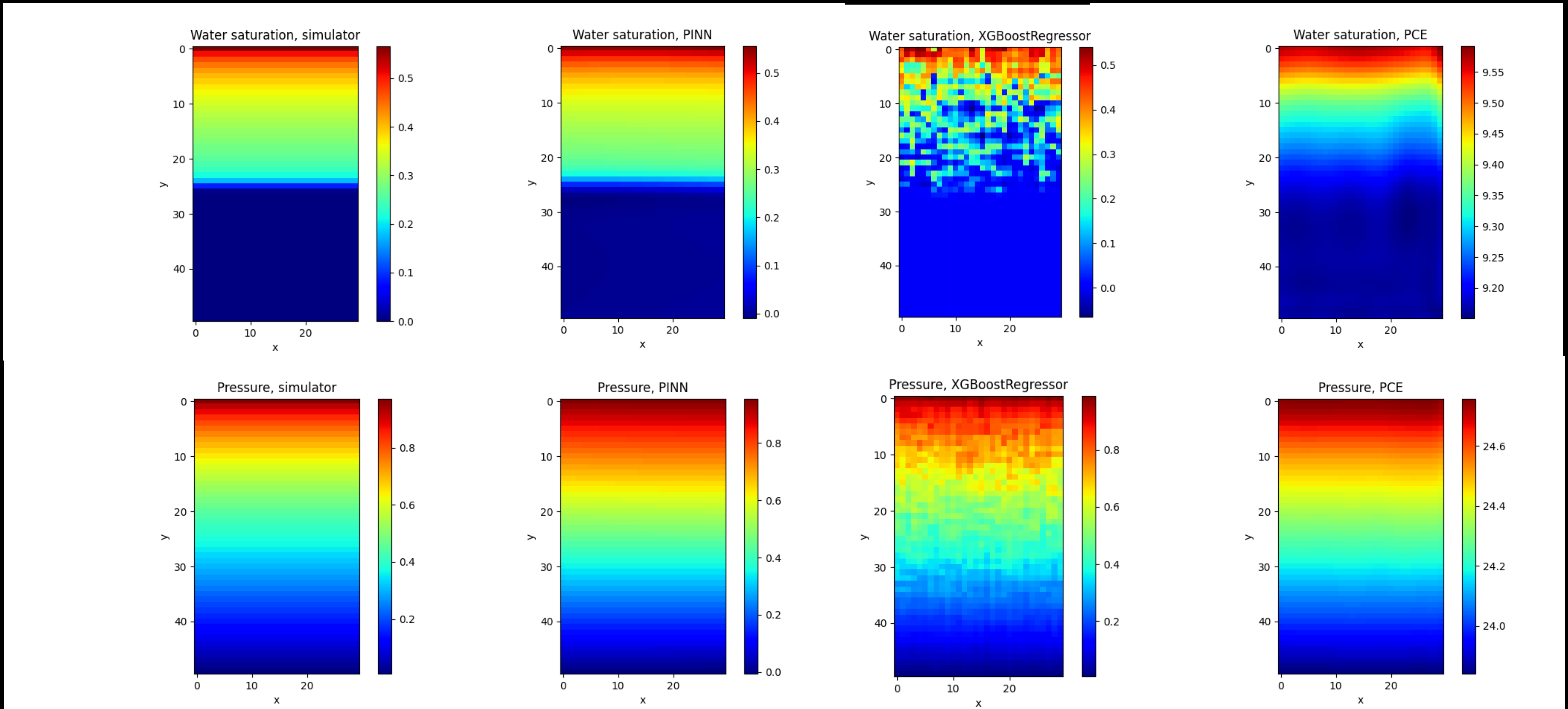
And as the result of all residuals we get weighted loss function

$$\mathcal{L} = w_0 \frac{1}{N}\sum_i \mathcal{R}_0(t_i,x_i) + w_1 \frac{1}{N}\sum_i \mathcal{R}_1(t_i,x_i) + w_2 \frac{1}{N}\sum_i \mathcal{R}_2(t_i,x_i)$$

First test case

- Here we want to demonstrate how work XGBoosting regressor, Polynomial chaos expansion and our PINN model on data with constant porosity (0.1) and constant permeability (1.0). We build prediction of the water saturation and pressure with t = 2.6. Each model was trained on same 2000 points (t, x).

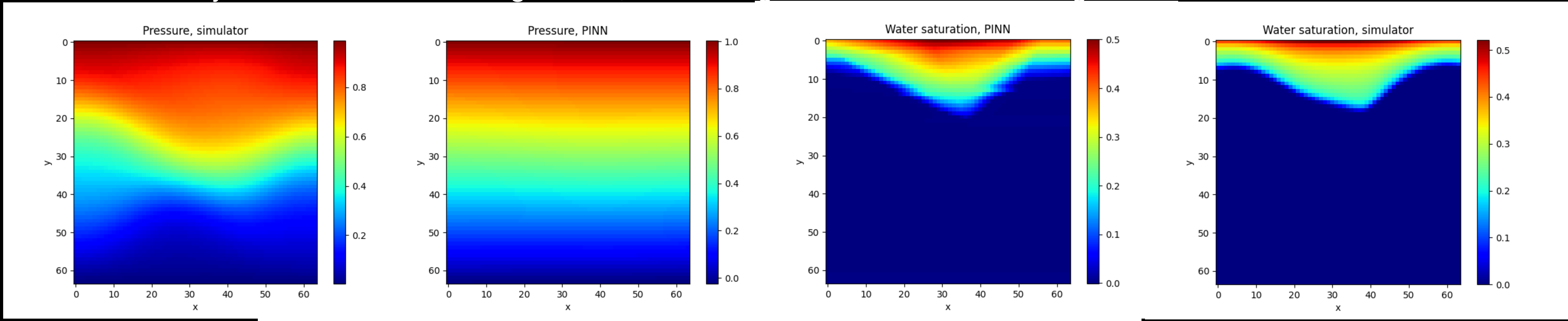
- As we can see PINN show the best result on the inference.

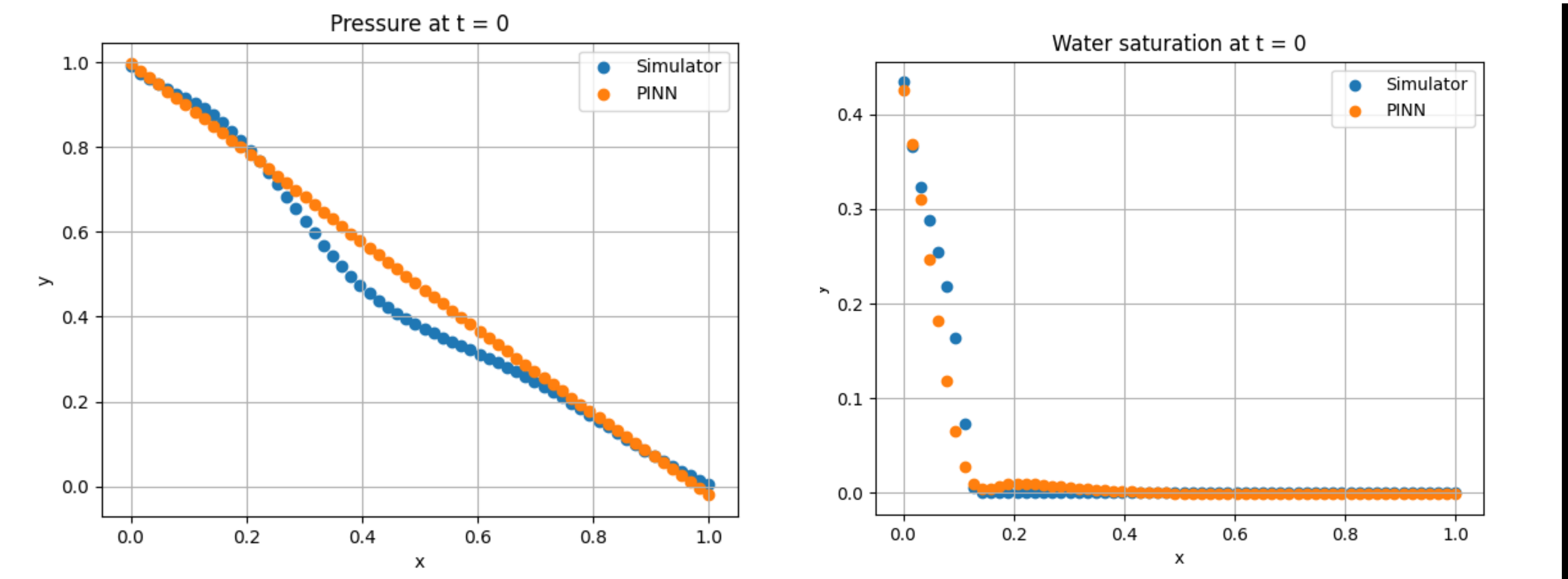


Second test case. PINN

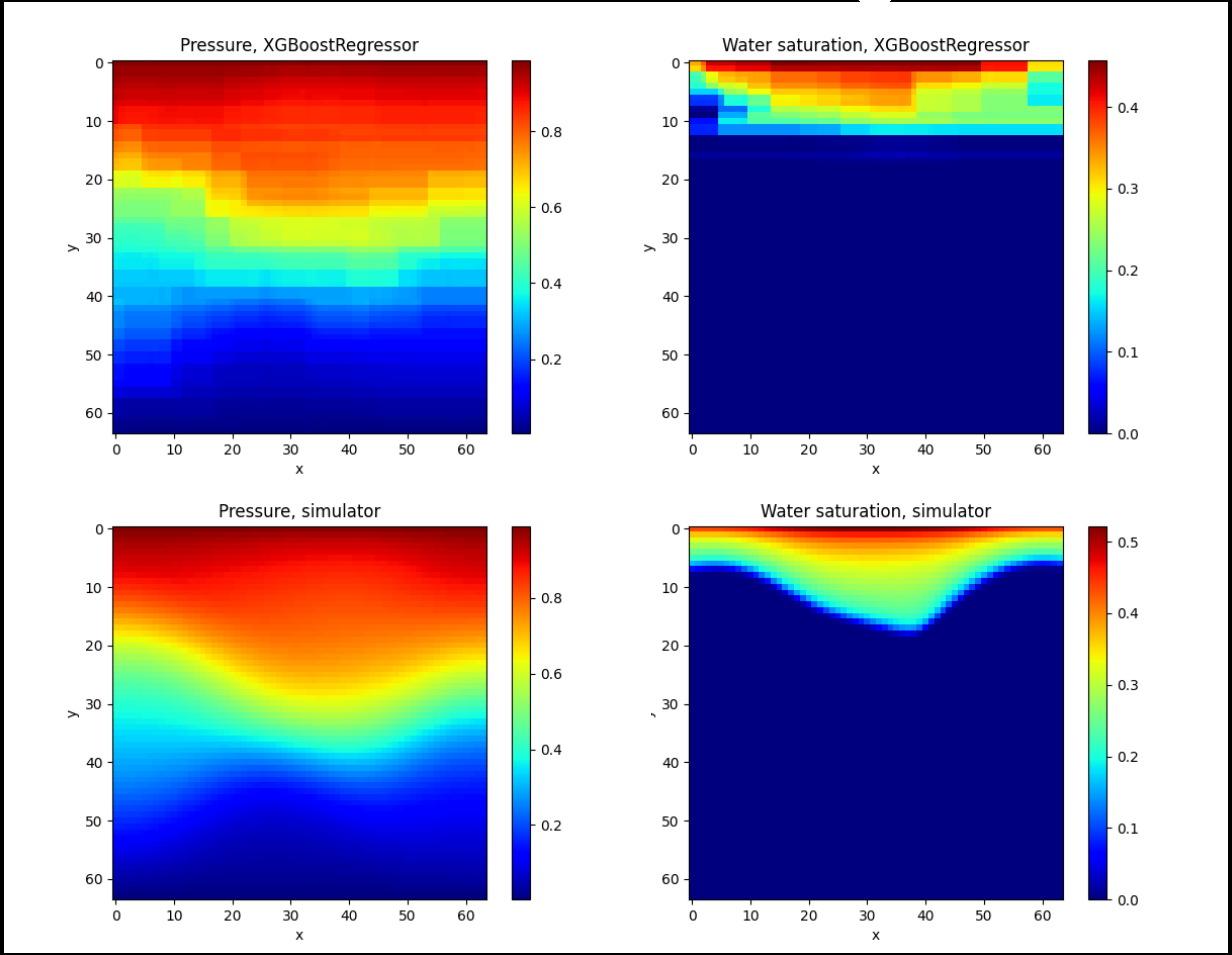
- This test aimed to challenge algorithms with nonlinear permeability. We generate new permeability field to train model for this reservoir structure.

- All the models have trained on 500 points. The goal is to show that PINNs can approximate nonlinearity with high accuracy rather that other ML algorithms, while using small dataset for training.

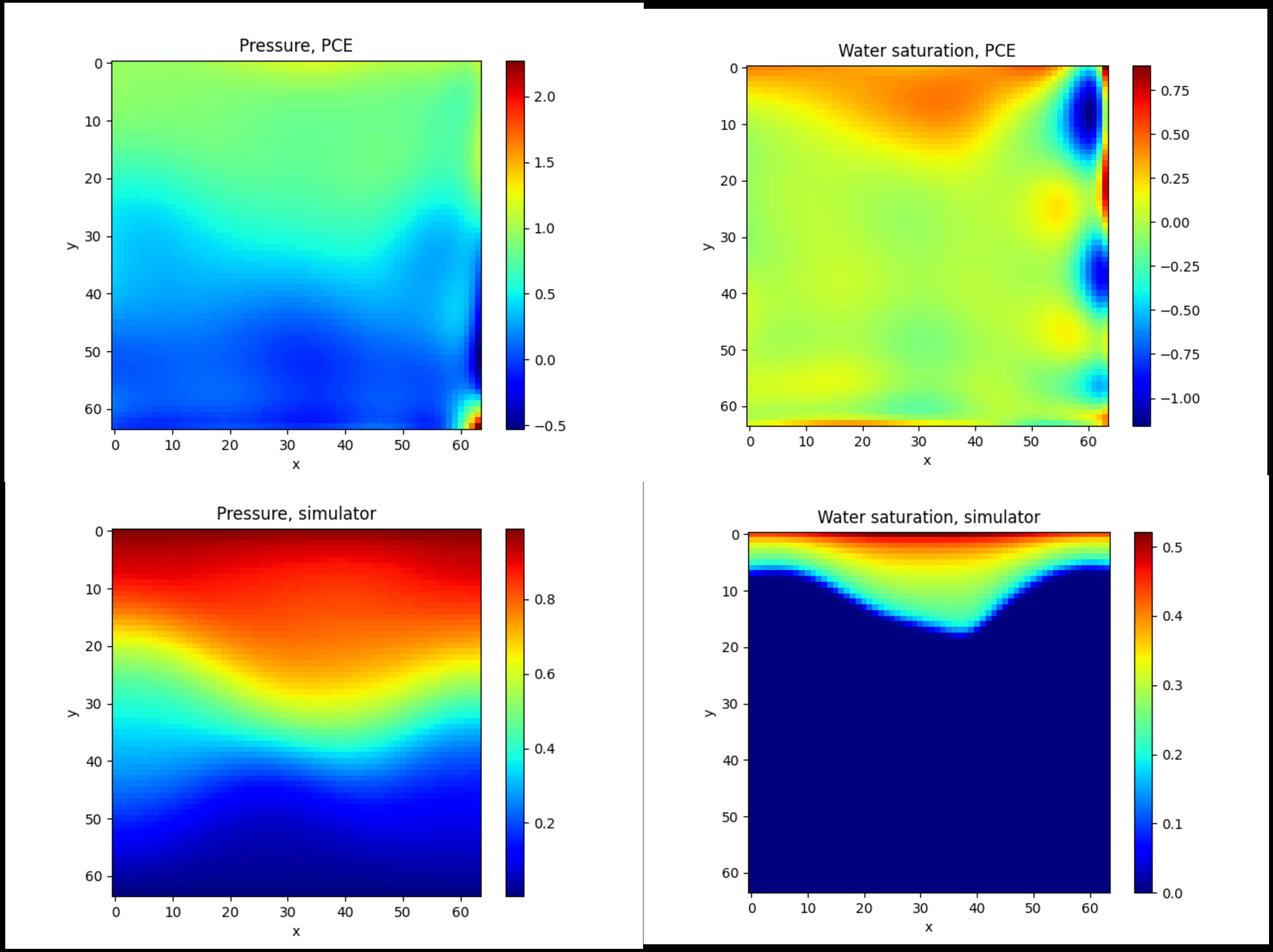




Second test case. Boosting



Second test case. PCE



Results

- Boosting and PCE can approximate systems with a low degree of nonlinearity quite well (1). However, if we increase the system's nonlinearity (as shown in the second test case by adding a nonlinear permeability field), these algorithms can only achieve good accuracy with a large amount of data. For example, with 500 training points, neither Boosting nor PCE were able to approximate the fluid propagation velocity in the reservoir, whereas PINNs managed to do so with much higher accuracy.

Train

PINNs

Metric	Saturation	Pressure
MSE	0.000016	0.004314
MAE	0.001584	0.054450
R^2	0.998415	0.960926
RMSE	0.003946	0.065684

Boosting

Metric	Saturation	Pressure
MSE	0.000281	0.000072
MAE	0.002961	0.003753
R^2	0.964437	0.999343
RMSE	0.016769	0.008501

PCE

Metric	Saturation	Pressure
MSE	0.002245	0.002057
MAE	0.022651	0.028261
R^2	0.741827	0.982704
RMSE	0.047379	0.045353

Inference

PINNs

Metric	Saturation	Pressure
MSE	0.000163	0.004363
MAE	0.003605	0.054572
R^2	0.981737	0.961467
RMSE	0.012757	0.066054

Boosting

Metric	Saturation	Pressure
MSE	0.005828	0.000781
MAE	0.031642	0.020360
R^2	0.345993	0.993107
RMSE	0.076341	0.027938

PCE

Metric	Saturation	Pressure
MSE	0.004454	0.000734
MAE	0.026650	0.019317
R^2	0.586226	0.993278
RMSE	0.066740	0.027092