energies

Article

Two-Step Predict and Correct Non-Intrusive Parametric Model Order Reduction for Changing Well Locations Using a Machine Learning Framework

Hardikkumar Zalavadia and Eduardo Gildin *

Harold Vance Department of Petroleum Engineering, Texas A&M University, College Station, TX 77843, USA; hardikzalavadia150@gmail.com

* Correspondence: egildin@tamu.edu

Abstract: The objective of this paper is to develop a two-step predict and correct non-intrusive Parametric Model Order Reduction (PMOR) methodology for the problem of changing well locations in an oil field that can eventually be used for well placement optimization to gain significant computational savings. In this work, we propose a two-step PMOR procedure, where, in the first step, a Proper Orthogonal Decomposition (POD)-based strategy that is non-intrusive to the simulator source code is introduced, as opposed to the convention of using POD as a simulator intrusive procedure. The non-intrusiveness of the proposed technique stems from formulating a novel Machine Learning (ML)-based framework used with POD. The features of the ML model (Random Forest was used here) are designed such that they take into consideration the temporal evolution of the state solutions and thereby avoid simulator access for the time dependency of the solutions. The proposed PMOR method is global, since a single reduced-order model can be used for all the well locations of interest in the reservoir. We address the major challenge of the explicit representation of the well location change as a parameter by introducing geometry-based features and flow diagnostics-inspired physics-based features. In the second step, an error correction model based on reduced model solutions is formulated to correct for discrepancies in the state solutions at well grid blocks expected from POD basis for new well locations. The error correction model proposed uses Artificial Neural Networks (ANNs) that consider the physics-based reduced model solutions as features, and is proved to reduce the error in QoI (Quantities of Interest), such as oil production rates and water cut, significantly. This workflow is applied to a simple homogeneous reservoir and a heterogeneous channelized reservoir using a section of SPE10 model that showed promising results in terms of model accuracy. Speed-ups of about 50×–100× were observed for different cases considered when running the test scenarios. The proposed workflow for Reduced-Order Modeling is "non-intrusive" and hence can increase its applicability to any simulator used. Additionally, the method is formulated such that all the simulation time steps are independent and hence can make use of parallel resources very efficiently and also avoid stability issues that can result from error accumulation over time steps.

Keywords: non-intrusive parametric model order reduction; Machine Learning; well location; flow diagnostics; Artificial Neural Network; Random Forests; Proper Orthogonal Decomposition

Citation: Zalavadia, H.; Gildin, E. Two-Step Predict and Correct Non-Intrusive Parametric Model Order Reduction for Changing Well Locations Using a Machine Learning Framework. *Energies* **2021**, *14*, 1765. https://doi.org/10.3390/en14061765

Academic Editor: Reza Rezaee

Received: 14 January 2021
Accepted: 22 February 2021
Published: 22 March 2021

1. Introduction

Numerical simulations involve solving a large-scale system of equations derived from the Partial Differential Equations (PDEs) of complex engineering models. This is the case with reservoir simulations that are used extensively in Closed-Loop Reservoir Management (CLRM) [1] and Closed-Loop Field Development (CLFD) [2] workflows. CLRM, which is based on the optimal continuous operation of existing wells, has been an area of significant research. The workflow of CLRM involves well control optimization based on current geological knowledge, using these settings to collect data for some time periods and finally performing history matching to integrate these data to update the geological models for consistency with the observed data. CLFD, which is an extension of CLRM, involves the optimization of well number, type, location, and controls based on current geological knowledge. This is followed by sequentially drilling new wells and collecting data and a history matching step from the available data to update geological information; this is repeated until the optimal number of wells are drilled. These workflows, repeated during the development of reservoirs, can help us make better reservoir management decisions than heuristic approaches can.

The optimization and history matching steps during the development cycle require rapid and repeated fine-scale reservoir simulation runs, which correspond to solving a discretized parametric PDE which is time-dependent and highly nonlinear in nature. Accurate high-fidelity computational models can therefore incur substantial computational costs. To avoid this bottleneck of costly evaluations, researchers have developed various surrogate modeling techniques that are approximations of high-fidelity models which aim to provide large computational savings while preserving the accuracy. One such technique is Reduced-Order Modeling (ROM), which includes intrusive or non-intrusive projection-based methods. They do not approximate the physics and rely on reducing the dimensionality of the state-space of fine-scale models [3,4], and thus are usually computationally cheaper than coarsened models [5,6] and more robust than data-driven models [7,8].

ROM approaches have been applied to reservoir simulation mainly within the context of well-control optimization problems with fixed well configurations and have reached a good level of maturity to achieve significant computational savings. Initial work on ROM for the well-control optimization problem started with the application of a method called Proper Orthogonal Decomposition (POD) [9,10]. However, there is not much speed-up gained by using this POD strategy, since it only targets the linear solver. Full Jacobian and residual matrices are still computed during each iteration in a simulator. In order to overcome this limitation of POD, other techniques were developed for the fast computation of non-linear terms in the reservoir simulator. One such technique developed was Trajectory Piecewise Linear (TPWL), which is applied in conjunction with POD for production optimization [11]. This is accomplished by linearizing the states around previously saved states using Taylor series expansions. Computational speed-ups of the $\mathbb{O}(10^2$–$10^3)$ in the online testing stage were shown in References [11,12]. Later, TPWL was extended to include the quadratic terms for the more accurate representation of non-linear functions and thus called the Trajectory Piecewise Quadratic (TPWQ) method [13]. This method has been proven to be more accurate compared to TPWL, but the overhead time increases significantly because it involves third-order matrix-tensor products. Another such method called the Discrete Empirical Interpolation Method (DEIM) [14] was also introduced and applied in combination with POD for a changing well-control optimization problem [15–17]. This approach can provide higher accuracies, as it aims at reducing the dimensions of discretized parametric PDEs by computing the non-linear terms at discrete locations in the spatial domain and then interpolating them to the rest of the locations using projection-based interpolation. However, less speed-ups have been reported for POD-DEIM as compared to POD-TPWL as a result of the linearization performed in the latter, and it is also more invasive with respect to the simulator as compared to POD-TPWL. Other methods were later implemented for this application of changing well controls, such as POD-Trajectory-based DEIM (TDEIM) [18], which is a combination of TPWL and DEIM techniques to utilize their benefits and avoid their shortcomings; bilinear approximation and quadratic bilinear formulation [19,20]; and the Gauss–Newton with Approximated Tensors (GNAT) method [21]. Recently, non-intrusive methods have been developed for a well-control optimization problem—namely, Dynamic Mode Decomposition with control (DMDc) and input-output DMD (ioDMD) in Reference [22].

Well location optimization in the context of surrogate or low-complexity models was developed in References [23,24]. However, until now, for projection-based reduced-order models the focus mainly has been on developing them for well-control optimization problems. However, these ROM workflows were not meant for integrated schemes such as Closed-Loop Field Development, which involve changing reservoir or well configurations. This challenge is the main objective of this work. For complex non-linear problems with non-affine parameter dependence, as mentioned before, the direct (simulator invasive) use of POD does not provide major computational advantages. This is because of the cost to compute the projection coefficients of the non-linear functions that depend on the dimension of the high-fidelity model. As the dimension of a system increases, the computational advantage of using a pure POD method decreases. Other alternative methods were also developed that still use POD implementation, but the coefficients are not obtained by a projection process, but rather by interpolation over the parameter domain of a reduced-order basis [25]. Since reduced-order basis (ROB), which is a basis of lower dimensions, belongs to non-linear matrix manifolds, standard interpolation techniques cannot be applied, as they would not preserve the characteristics of the ROB, unless a large number of samples were used [26,27]. To overcome this issue, researchers have developed an ROB method using POD and then employed Machine Learning techniques to interpolate the basis coefficients. Reference [28] developed POD-NN, where he used Neural Networks (NN) for high-dimensional coefficient prediction. Reference [29] compared the performance of different ML techniques to predict the basis coefficients for an aerodynamic and a structural problem. They considered the use of particular solutions in the POD expansion as a way to embed physical constraints that must be preserved during coefficient prediction. These methods, however, did not consider time-dependent problems. Such a non-intrusive ROM scheme has been applied here for changing well locations, which is highly non-linear in nature and also involves time dependency. In Reference [30], a local Parametric Model Order-Reduction (PMOR) technique based on error map prediction was developed to choose an appropriate basis from a dictionary of pre-existing ROBs for a new well location. However, this method is still limited to use for the cases when there is no water cut observed at the producers and for reservoirs with less permeability contrast. In the proposed work here, we seek to develop a method that is not limited by these restrictions using a non-intrusive global PMOR technique.

Thus, the novelty of this work is to propose a Reduced-Order Modeling procedure for the case of changing well locations, which, to the best of our knowledge, has not been addressed before. The proposed two-step procedure is non-intrusive to the simulator code and hence can be applied with any commercial simulator, since it takes advantage of Machine Learning models to predict the POD basis coefficients and build the error correction models. Similar error correction-based techniques have been applied before for the well-control optimization problem, as addressed above. The proposed technique also addresses the major challenge of representing the changing well locations, as described by various geometric and physics-based features. The main point to be noted here though is that this method needs to be applied in a two-step procedure as described here, since just using POD is not expected to produce reasonable results due to the nature of the state solutions for changing well locations. A POD step was observed to obtain a general idea of the solution trend; then, in the error correction step, this trend is used to close the gap to the actual fine-scale solution.

We begin with a section introducing governing reservoir simulation equations followed by concepts leading to projection-based parametric ROM. We then show the motivation behind using a non-intrusive global PMOR technique. Here, we discuss parameterizing the states of the reservoir using POD-based model order reduction followed by a way of addressing the physical constraints in the POD method. In the next section, we formulate the non-intrusive PMOR problem using Machine Learning methods with a brief discussion of the ML methods and feature selection techniques used. This formulation is then validated using some case studies for the prediction of pressure and saturation states at new well locations. In the next section, errors in the well block states are modeled using ML algorithms to correct for the errors in the output quantities of interest (QoI), such as oil, water production rates, and water cut, followed by some case studies explaining the validity of the proposed method.

2. Reservoir Simulation Concept

The governing equations for a two-phase isothermal immiscible incompressible oil–water system is obtained by combining the mass conservation law and Darcy's law for two phases. These governing equations considering the gravity effects and constant porosity φ are PDEs of the form as shown in Equations (1) and (2):

$$\varphi \frac{\partial S_l}{\partial t} = \nabla \cdot (\mathbf{v}_l) - q_l \tag{1}$$

$$\mathbf{v}_l = \lambda_l K(\nabla p_l - \rho_l g \nabla z) \tag{2}$$

where l indicates liquid, oil (o), or water (w), φ is the porosity of the reservoir, ρ_l indicates the density of the fluid per unit volume, and $\widetilde{m_l}$ (mass flow rate) $= \rho_l q_l$ denotes the external sources and sinks, with q_l being the volumetric flow rate. $\mathbf{v}_l$ here represents the Darcy velocity of the liquid (l). $\lambda_l = k_{rl}(S_l)/\mu_l$, where μ_l and k_{rl} are the viscosity of fluids and the relative permeability of fluids, respectively, and K is the absolute permeability. p_l and S_l are the pressure and saturation of each phase, respectively. g is the gravity acceleration constant and z accounts for vertical coordinates. These equations for both oil and water, along with the constraints, saturation equation ($S_o + S_w = 1$), and capillary equation ($p_{cow} = p_o - p_w$) complete the oil–water model. In these equations, p_o and S_w are considered to be the primary unknowns, and once they are solved the rest of the unknowns p_w and S_o are computed easily from the constraint equations.

In order to solve the PDE, we perform spatial discretization of this equation that results in the parametric system of ODE in time, as shown in Equation (3), to solve the state variables p_o and S_w represented by $\mathbf{x}$:

$$-\mathbf{D}(\mathbf{x},\zeta)\dot{\mathbf{x}}(t,\zeta) + \mathbf{T}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) + \mathbf{G}(\mathbf{x},\zeta) + \mathbf{Q}(\mathbf{x},t,\zeta) = \mathbf{R}(\mathbf{x},\zeta) \tag{3}$$

$\mathbf{x}(t,\zeta) = [P_o, S_w] \in R^{N_d}$ is a state vector with N_d degrees of freedom and $\zeta \in \mathbb{P} \subset R^{N_\zeta}$ is a vector of parameters (which correspond to the well location parameter in this paper). $\dot{\mathbf{x}}(t,\zeta)$ is the derivative of the states with respect to time, $\mathbf{D}$ is the accumulation matrix, $\mathbf{T}$ is the transmissibility matrix, $\mathbf{G}$ is the gravity vector, $\mathbf{Q}$ contains the (volumetric) sources and sink terms (q_l), and $\mathbf{R}$ is the residual vector. We satisfy $\mathbf{R}$ = 0 upon convergence to the solution for each time step. We consider the fully implicit procedure to solve Equation (3). The source/sink volumetric term corresponds to production/injection from a well and is written as the Peaceman equation in the form as shown in Equation (4):

$$q_l = WI_l\, \lambda_l(S_l)\, (p_o - p_{wf}) \tag{4}$$

where WI_l is the well index associated with each phase, p_o is the well-block pressure of the oil phase, and p_{wf} is the bottom hole flowing pressure. Thus, $\mathbf{Q}(\mathbf{x},\zeta)$ is a vector with q_l value at the indices corresponding to the well grid blocks and 0 elsewhere.

The parameter ζ of the system depends on the problem under consideration. For well-control optimization, $\zeta = p_{wf} \in R^{N_t}$ for each well, where N_t is the number of simulation time steps. For well placement optimization, ζ represents the indices in sparse vector $\mathbf{Q}$ corresponding to the spatial location of the well to be optimized in the source/sink vector. Since, in this paper, we are interested in the changing well location problem, we consider the latter.

$$\zeta = \begin{cases} p_{wf}\ \ or\ \ q_{inj}, & \text{for well control optimization} \\ indices\ \ in\ \ vector\ \ \mathbf{Q}, & \text{for well location optimization} \end{cases} \tag{5}$$

Using the fully implicit method, for a given system parameter at each time step this non-linear system of Equation (3) is solved using Newton's method, as shown in Equation (6) and then the update is shown in Equation (7):

$$\mathbf{J}(\zeta)^{n+1}\,\delta(\zeta)^{n+1} = -\mathbf{R}(\zeta)^{n+1} \tag{6}$$

$$\mathbf{x}(\zeta)^{n+1} = \mathbf{x}(\zeta)^{n} + \delta(\zeta)^{n+1} \tag{7}$$

where n and $n+1$ represent the previous and current time levels, respectively, and $\mathbf{J}^{n+1} = \frac{\partial \mathbf{R}^{n+1}}{\partial \mathbf{x}^{n+1}}$ is the Jacobian matrix. We satisfy $\mathbf{R}^{n+1} = 0$ upon convergence to the solution for each time step, which thus requires multiple Newton iterations (see ([31]) for the detailed derivation). The size of the Jacobian matrix is $N_d \times N_d$ and that of the residual is $N_d \times 1$. The degrees of freedom in a reservoir simulation usually ranges from hundreds to millions. Thus, the size of the system matrices (Jacobian and residual) significantly increases the time to solve the linear system in the Newton method. For a detailed explanation of oil–water flow simulation, one can refer to some classical literature ([31–34]).

3. Projection-Based Model Order Reduction

In this section, we introduce a projection-based MOR technique to be used for changing well locations that projects a high-dimensional reservoir simulation equation onto a subspace of a much lower dimension, thus reducing both the number of equations and variables involved.

3.1. Galerkin Projection

The main idea of projection-based MOR is to generate a dynamical system of much lower-dimension r as compared to the full-order dimension N_d, such as in Equation (6), while still retaining its dominant dynamical properties. One way to achieve this is Galerkin projection, which is employed here. We begin with a state space representation of the reservoir simulation equation for two-phase flow to explain the Galerkin projection. Referring to Equation (3), at convergence, neglecting gravity, it can be re-written as:

$$\dot{\mathbf{x}}(t,\zeta) = \mathbf{D}(\mathbf{x},\zeta)^{-1}\mathbf{T}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) - \mathbf{D}(\mathbf{x},\zeta)^{-1}\mathbf{Q}(\mathbf{x},t,\zeta) \tag{8}$$

This can be written as a state space form of Equation (9):

$$\dot{\mathbf{x}}(t,\zeta) = \mathbf{A}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) + \mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{9}$$

This has the initial condition $\mathbf{x}(0,\zeta) = \mathbf{x}^0(\zeta)$. The states here represent pressure and saturation for two-phase flow—i.e., $\mathbf{x}(t,\zeta) = [P_o, S_w]$. Here, $\mathbf{A} = \mathbf{D}(\mathbf{x},\zeta)^{-1}\mathbf{T}(\mathbf{x},\zeta)$ and $\mathbf{B} = -\mathbf{D}(\mathbf{x},\zeta)^{-1}$.

The first step in Galerkin projection is to define a trial basis $\Phi(\zeta) \in R^{N_d \times r}$, referred to as ROB (Reduced-Order Basis), which is full rank and describes the subspace $\mathcal{S}_\Phi(\zeta)$. The state vector $\mathbf{x}$ can then be decomposed as the sum of two orthogonal components, one in $\Phi(\zeta)$ and the other in $\Phi^\perp(\zeta)$, which can be written as:

$$\mathbf{x}(t,\zeta) = \Phi(\zeta)\,\mathbf{x}_r(t,\zeta) + \Phi^\perp(\zeta)\,\tilde{\mathbf{x}}_r(t,\zeta) \tag{10}$$

where $\mathbf{x}_r(t,\zeta) \in R^r$ and $\tilde{\mathbf{x}}_r(t,\zeta) \in R^{N_d - r}$. For a reduced representation of the state space, we neglect components of the state in $\Phi^\perp(\zeta)$. Thus,

$$\mathbf{x}(t,\zeta) \approx \Phi(\zeta)\,\mathbf{x}_r(t,\zeta) \tag{11}$$

Thus, $\mathbf{x}_r \in R^r$ represent the components of the state vector $\mathbf{x} \in R^{N_d}$ in the subspace $\Phi(\zeta)$. We can now write the Equation (9) as the expression shown in Equation (12):

$$\Phi(\zeta)\dot{\mathbf{x}}_r(t,\zeta) = \mathbf{A}(\mathbf{x},\zeta)\Phi(\zeta)\mathbf{x}_r(t,\zeta) + \mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{12}$$

The next step is to define the test basis, which for Galerkin projection is the same as the trial basis—i.e., $\Phi(\zeta)$. This test basis is left multiplied in Equation (12) to obtain a lower-order system equation.

$$\Phi(\zeta)^T\Phi(\zeta)\dot{\mathbf{x}}_r(t,\zeta) = \Phi(\zeta)^T\mathbf{A}(\mathbf{x},\zeta)\Phi(\zeta)\mathbf{x}_r(t,\zeta) + \Phi(\zeta)^T\mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{13}$$

The columns of $\Phi(\zeta)$ are orthonormal—i.e., $\Phi(\zeta)^T\Phi(\zeta) = I_r$ and the projector is $\Pi_{\Phi(\zeta),\Phi(\zeta)} = \Phi(\zeta)\Phi(\zeta)^T$. Let $\hat{\mathbf{x}}$ be the approximated full state after projection. The model reduction error is then:

$$\begin{aligned}\epsilon_{ROM}(t,\zeta) &= \mathbf{x}(t,\zeta) - \hat{\mathbf{x}}(t,\zeta)\\ &= \mathbf{x}(t,\zeta) - \Pi_{\Phi(\zeta),\Phi(\zeta)}\mathbf{x}(t,\zeta) + \Pi_{\Phi(\zeta),\Phi(\zeta)}\mathbf{x}(t,\zeta) - \hat{\mathbf{x}}(t,\zeta)\\ &= (I_n - \Pi_{\Phi(\zeta),\Phi(\zeta)})\mathbf{x}(t,\zeta) + \Phi(\zeta)(\Phi(\zeta)^T\mathbf{x}(t,\zeta) - \mathbf{x}_r(t,\zeta))\\ &= \epsilon_{\Phi(\zeta)^\perp}(t,\zeta) + \epsilon_{\Phi(\zeta)}(t,\zeta)\end{aligned} \tag{14}$$

The first term $\epsilon_{\Phi(\zeta)^\perp}(t,\zeta)$ corresponds to the projection error that results from neglecting the state projection on the orthogonal subspace, while the second term $\epsilon_{\Phi(\zeta)}(t,\zeta)$ is the result of error from solving a dynamical system that is different from the original one. Since the error components are orthogonal, we can write the following equality:

$$\parallel \epsilon_{ROM}(t,\zeta) \parallel_2^2 = \parallel \epsilon_{\Phi(\zeta)^\perp}(t,\zeta) \parallel_2^2 + \parallel \epsilon_{\Phi(\zeta)}(t,\zeta) \parallel_2^2 \tag{15}$$

This is used for an a priori indication of the ROM error that is indicative of the quality of the basis $\Phi(\zeta)$. The orthogonal component of the projection error can be computed by just knowing the fine-scale solution. Thus, if $\Phi(\zeta)$ is not a suitable basis for parameter ζ, $\epsilon_{\Phi(\zeta)^\perp}$ will be large and the full ROM error $\epsilon_{ROM}(t,\zeta)$ will be even larger.

3.2. Proper Orthogonal Decomposition

Proper Orthogonal Decomposition (POD) constructs a basis of dimension r by the orthogonal transformation of the data observations, such that it represents the data in a certain least-square optimal sense. The data can thus be represented by a linear combination of the basis vectors, which are called the basis functions.

POD is typically employed in the state solutions of the system. As a first step to generate the reduced basis, the full-order system is solved for a given parameter ζ, which is called the training step, to generate an ensemble of snapshots which are basically the state solutions gathered at all simulation time steps by solving Equations (6) and (7). More details on this can be found in [15]. The snapshot matrix $\mathbf{X}(\zeta)$ is defined as:

$$\mathbf{X}(\zeta) = \{\mathbf{x}^1 \quad \mathbf{x}^2 \quad \mathbf{x}^3 \quad \ldots \quad \mathbf{x}^{N_t}\}, \quad \mathbf{X}(\zeta) \in R^{N_d \times N_t} \tag{16}$$

where N_d is the full-order dimension of states P_o and S_w, and N_t is the total number of snapshots collected over a time period. From now, we denote P_o and S_w as $\mathbf{p}$ and $\mathbf{S}$, respectively. In order to project a fine-scale system to a low-dimensional space, the projection basis $\{\phi_i\}_{i=1}^r$ is obtained by solving the minimization problem:

$$\min_{\phi_i} \sum_{j=1}^{N_t} \parallel \mathbf{x}^j - \sum_{i=1}^{r} (\mathbf{x}^{j^T}\phi_i)\phi_i \parallel \tag{17}$$

The solution to this minimization problem is given by the SVD of snapshot matrix $\mathbf{X}(\zeta)$ [4] and selecting the first r columns of the left projection matrix.

$$\mathbf{X}(\zeta) = \mathbf{U}(\zeta)\mathbf{\Lambda}(\zeta)\mathbf{V}(\zeta)^T \tag{18}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular matrices, respectively, and $\mathbf{\Lambda}$ is a diagonal matrix with singular values in decreasing order. The r columns of $\mathbf{U}$ are usually selected by the fraction of energy to be captured (generally more than 90%), as shown in Equation (19):

$$E = \frac{\sum\limits_{i=1}^{r} \sigma_i}{\sum\limits_{i=1}^{N_d} \sigma_i} \tag{19}$$

where σ_i is the i^{th} diagonal element of $\mathbf{\Lambda}$. Thus, for a fixed parameter, the states span the space $\Phi(\zeta) = [\mathbf{U}_1(\zeta) \;\; \mathbf{U}_2(\zeta) \;\; ... \;\; \mathbf{U}_r(\zeta)] \in R^{N_d \times r}$ and for each state:

$$\mathbf{p}(\zeta) \approx \Phi_p(\zeta)\mathbf{p}_r(\zeta), \qquad\qquad \mathbf{S}(\zeta) \approx \Phi_s(\zeta)\mathbf{S}_r(\zeta) \tag{20}$$

Thus, for a given parameter, we may write:

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \Phi_p & 0 \\ 0 & \Phi_s \end{bmatrix} \begin{bmatrix} \mathbf{p}_r \\ \mathbf{S}_r \end{bmatrix} \tag{21}$$

This takes a short notation:

$$\mathbf{X} = \Phi \mathbf{X}_r \,, \quad r \ll N_d \tag{22}$$

where Φ is a diagonal matrix with pressure basis and saturation basis as its diagonal elements. Using Galerkin projection, the linear system of equations in Equation (6) can be written as follows:

$$\Phi(\zeta)^T \mathbf{J}(\zeta)^{n+1} \Phi(\zeta)\, \delta_r(\zeta)^{n+1} = -\Phi(\zeta)^T \mathbf{R}(\zeta)^{n+1} \tag{23}$$

This leads to a system of reduced-order equations as shown in Equation (24):

$$\mathbf{J}_r(\zeta)^{n+1} \delta_r(\zeta)^{n+1} = -\mathbf{R}_r(\zeta)^{n+1} \tag{24}$$

After solving Equation (24), the reduced state space variables are updated as:

$$\mathbf{x}_r(\zeta)^n + \delta_r(\zeta)^{n+1} = \mathbf{x}_r(\zeta)^{n+1} \tag{25}$$

Thus, with r being of much smaller magnitude compared to the fine-scale degrees of freedom N_d, POD has been shown to achieve significant speed-ups at the linear solver level in reservoir simulation applications by greatly reducing the dimensions of the huge Jacobian and residual matrices. However, since this method still requires us running fine-scale simulations to evaluate the full Jacobian and residual matrices, it poses a challenge in terms of overall computational benefit. This problem will be addressed in the next sections as we build on the MOR strategy for changing well locations.

3.3. Parametric Model Order Reduction Concept

Thus, as we saw above for POD, the main aim is to replace the computationally expensive solver in Equation (6) with a reduced form of the Jacobian and Residual, as shown in Equation (24), where $r \ll N_d$. To this end, we aim to construct reduced-order models or reduced-order basis that are an accurate approximation of the fine-scale solution in the *entire* parameter domain—i.e., for any ζ in Equation (3). For clarity, ROB is defined as the basis of the lower dimension obtained by POD and the ROBs are used to construct ROMs, which are lower-dimensional system matrices, the accuracy of which is reflected in the predicted output of the system. Thus, ROM error refers to the error in output predicted by the reduced system.

Thus, for any parameter $\zeta \in \mathbb{P}$ representing well location, we require the condition that the ROM error is less than a specified threshold, which is defined mathematically as:

$$\parallel \mathbf{X}(t,\zeta) - \Phi(\zeta)\mathbf{X}_r(t,\zeta) \parallel < \eta \tag{26}$$

where η is some maximum allowable error tolerance between a fine-scale and ROM solution, with $\Phi(\zeta)$ obtained by POD in this work. We note again that the parameter ζ depends on the optimization problem considered.

In the following sections, we develop MOR techniques that are developed in the context of POD concept for the fast computation of simulations for changing well locations during well location optimization.

4. Motivation for Non-Intrusive Global PMOR Using Machine Learning

In this section, we discuss the motivation behind developing the non-intrusive global PMOR method using Machine Learning techniques. It was shown in the work [30] that the ROB corresponding to one well location is only a valid basis for fewer locations when compared to the ROB obtained by concatenating solutions from different well locations. This is the reason behind considering the global ROB strategy for well location problems. The global ROB can be obtained by random sampling in the parameter space or using greedy algorithms. For the former case, fine-scale simulations for randomly distributed parameters are computed and, using POD, a reduced basis is obtained. It is highly likely that the distribution contains many unnecessary samples or neglects some important samples. Meanwhile, when using greedy algorithms, the samples are carefully chosen using optimal criteriaand this usually leads to much better samples. However, the computational expense involved in a greedy sampling procedure for certain kinds of problems, especially highly non-linear dynamical systems such as those used in reservoir simulation, makes it infeasible to use. Thus, for this work we just focus on the random sampling of the candidate well locations.

Some of the non-intrusive methods that have been used for changing well controls are DMD [35], DMDc, and ioDMD [22]. For changing well locations, we propose a new non-intrusive PMOR technique that has been applied in [28,29] for problems that are time-independent or for steady-state systems. We extend this technique to time-dependent problems, which is the case for reservoir simulation where states evolve over time. One of the other main advantages of proposing this strategy is the fact that it is extremely fast compared to using global ROB within a simulator to project non-linear functions. As discussed before, the reason for this is that, as more parameters are used for concatenating solutions for a global ROB, we expect the basis dimension to increase monotonically and hence the projection of non-linear functions to become slower. The proposed PMOR technique does not involve computing non-linear functions and thus projecting them, and can make efficient use of parallel computational facilities, as will be discussed later in Section 5.

The use of Machine Learning has revolutionized the field of decision-making in numerous areas of applications, where it learns complex models entirely from huge data sets. However, the availability of an extensive amount of data for engineering applications can be a bottleneck, as this usually involves running a large number of fine-scale simulations. Another challenge is to ensure that the ML models understand the physics of the system, such as conservation laws. Thus, blending the positives of both MOR and ML (in this, MOR provides a low-dimensional representation of the dynamics of fluid flow and hence represents the physics of the system, and ML provides a complex mapping of the input parameters (well locations in our case) to the MOR parameters, as will be discussed in the next section) is the main motivation for using this strategy.

4.1. Parameterizing States of Reservoir Using POD

We begin by introducing the parameterization of the reservoir states using Proper Orthogonal Decomposition. Let us consider the states $\mathbf{x}(t,\zeta) \in R^{N_d}$ at time $t \in \mathbb{T}$ and

$\zeta \in \mathbb{P} \subset R^{N_\zeta}$, an input parameter. Each column in $\mathbf{x}$ collected over time is also referred to as snapshots. $\mathbf{x}$ represents the pressure and saturation states for our case—i.e., $x = [p, s]$—and ζ corresponds to the well location. In order to construct the global ROB, we collect the states snapshots from different well locations and these hence can be written as Equation (27):

$$S = \{\mathbf{x}(t_i, \zeta_j) | i = 1, ..., N_t, j = 1, ..., N_\zeta\} \tag{27}$$

Thus, we have a total of $N_t N_\zeta$ snapshots, where each ζ parameter has N_t snapshots, all concatenated in a single snapshot matrix S. To avoid confusion, ζ corresponds to a single well location and N_ζ is the total number of well configurations selected in training. The N_ζ parameters can be selected using experimental design sampling methods such as Lattice Hypercube Sampling (LHS), full factorial design, etc., or randomly. Thus, POD entails obtaining a reduced-order basis (ROB) by taking a singular value decomposition of the snapshot matrix S and then selecting first few vector r (reduced dimension) of the left singular matrix by certain energy criteria on the singular values. This ROB is represented by $\mathbf{\Phi}$ = $\{\phi_1(x), \phi_2(x), ..., \phi_r(x)\}$, where $\phi_i(x), i = 1, ..., r$, are called the reduced-basis functions, and we seek solutions at any parameter that is a linear combination of these basis functions. We assume that these basis functions span a space Φ_{rb} called the reduced basis space:

$$\Phi_{rb} = span\{\phi_1(x), \phi_2(x), ..., \phi_r(x)\} \tag{28}$$

where $r \ll N_d$. Thus, for any $\zeta \in \mathbb{P}$, we seek a solution $x(\zeta)$, as shown in Equation (29):

$$\tilde{\mathbf{x}}(t, \zeta) = \sum_{i=1}^{r} c_{\mathbf{x}i}(t, \zeta)\phi_{\mathbf{x}i}(x) \tag{29}$$

where $\mathbf{x}$ corresponds to p (pressure) or s (saturation). Note that $\mathbf{x}$ here denote the states and x denotes the spatial variable. Thus, we have:

$$\tilde{\mathbf{p}}(t, \zeta) = \sum_{i=1}^{r_p} c_{pi}(t, \zeta)\phi_{pi}(x) \tag{30}$$

$$\tilde{\mathbf{s}}(t, \zeta) = \sum_{i=1}^{r_s} c_{si}(t, \zeta)\phi_{si}(x) \tag{31}$$

$c_{pi}(t, \zeta)$ and $c_{si}(t, \zeta)$ represent the basis coefficients or POD expansion coefficients for pressure and saturation, respectively, and $\tilde{\mathbf{p}}(t, \zeta)$ and $\tilde{\mathbf{s}}(t, \zeta)$ represent the approximated pressure and saturation solutions at a parameter ζ and time t. Note that the magnitudes of pressure and saturation scale very differently, with pressures usually in the range of thousands of psi and saturation as a fraction between 0 and 1; we compute the global ROBs of pressure and saturation separately, as denoted by $\Phi_{pi}(\mathbf{x})$ and $\Phi_{si}(\mathbf{x})$. Thus, the dimension of both these basis can be different, as depicted by r_p and r_s for pressure and saturation, respectively. From now on, we just use r for both the states for ease of notation, but it should be remembered that it is different for different states.

These POD parameterize the states in terms of the basis coefficients. These basis coefficients can be calculated as:

$$\mathbf{c}_\mathbf{x}(\zeta) = \Phi_\mathbf{x}^T \mathbf{x}(\zeta) \in R^{r \times t}, \tag{32}$$

for both states. The matrix $\mathbf{c}_\mathbf{x}(\zeta)$ contains columns of basis coefficients at each time step. Thus, $\mathbf{c}_\mathbf{x}(t, \zeta) = [c_{\mathbf{x}1}(t, \zeta)\ \ , c_{\mathbf{x}2}(t, \zeta)\ \ , ..., c_{\mathbf{x}r}(t, \zeta)]^T$.

4.2. Addressing Physical Constraints in POD

In this method, we also enforce physical constraints in the form of POD representation [29]. This is achieved by writing the Equation (29) in a different way as:

$$\tilde{\mathbf{x}}(t,\zeta) = \bar{\mathbf{x}} + \sum_{i=1}^{r} c_{\mathbf{x}i}(t,\zeta)\bar{\phi}_{\mathbf{x}i}(x) \tag{33}$$

Here, $\bar{\mathbf{x}}$ is called the particular solution which enforces certain characteristics of the spatial behavior of states during POD-based prediction. For our case, this particular solution is defined as the mean of all the snapshots corresponding to all the parameters in snapshot matrix. By doing so, as expected, the accuracy of state prediction increased using the formulation proposed in the next Section 5. The procedure to calculate the global basis functions remain the same as above, except now, the POD basis $\bar{\Phi} = \{\bar{\phi}_1(x), \bar{\phi}_2(x), ..., \bar{\phi}_r(x)\}$ is computed on the mean subtracted snapshot matrix:

$$\tilde{S} = \{\mathbf{x}(t_i,\zeta_j) - \bar{\mathbf{x}} | i = 1,...,N_t, j = 1,...,N_\zeta\} \tag{34}$$

After the basis coefficients are obtained, the states at new parameters and a given time are computed as Equations (35) and (36).

$$\tilde{\mathbf{p}}(t,\zeta) = \bar{\mathbf{p}} + \sum_{i=1}^{r_p} c_{pi}(t,\zeta)\phi_{pi}(x) \tag{35}$$

$$\tilde{\mathbf{s}}(t,\zeta) = \bar{\mathbf{s}} + \sum_{i=1}^{r_s} c_{si}(t,\zeta)\phi_{si}(x) \tag{36}$$

5. Global PMOR Problem Formulation

5.1. Formulation

Once the global ROB is obtained using the method shown above, a traditional way of solving such system is solving the non-linear reservoir simulation equation online and getting the coefficients of the basis for a new parameter by projecting the non-linear Jacobian and residual functions. However, it has been shown several times in the literature that this technique shows modest improvements in the computational speed-ups as it requires computing the fine-scale non-linear functions during each Newton iteration. Additionally, for well location changes, implementing the simulator intrusive POD-based online procedure caused lot of stability issues and it is very challenging to decide suitable dimensions of global basis. Changing well location as the parameter for PMOR is observed as much more challenging as compared to well control changes as even a global PMOR technique is found difficult to represent the controllability properties of a new well location. We present a way to overcome this issue by the multidimensional mapping of input parameters to the basis coefficients given a global ROB. We use ML techniques that have capabilities of mapping complex non-linear relationships and specially suitable for interpolation of basis coefficients where the parameters have a non-affine dependence. Similar approach was applied to steady state cases in [28,29]. We extend this technique to time dependent problems. Thus, the ML model $\mathscr{F}$ is trained to learn the relation, as shown in the Equation (37):

$$\mathscr{F} : [\zeta_j, t_k] \rightarrow [c_{\mathbf{x}1}(\zeta_j, t_k), c_{\mathbf{x}2}(\zeta_j, t_k), ..., c_{\mathbf{x}r}(\zeta_j, t_k)], \quad j = 1,...,N_\zeta \text{ and } k = 1,...,N_t \tag{37}$$

$$[c_{\mathbf{x}i}(\zeta_j, t_k)]_{i=1}^{r} = \Phi_{\mathbf{x}}^T \mathbf{x}(\zeta_j, t_k), \;\; \mathbf{x} = [p,s]^T \tag{38}$$

Note that the basis $\Phi_{\mathbf{x}}$ is obtained from the mean subtracted snapshot matrix and the basis dimensions of pressure and saturation are different shown by r here. ζ_j are the parameters representing the well location used for training which are the features of the ML model. In order to consider the temporal evolution of the coefficients, we also add time as one of the features. We use multi-dimensional output in the ML models corresponding to all the coefficients of basis functions at a given time.

Once this ML model is trained, we predict the POD basis coefficients for each state, pressure, and saturation for a new well location. These coefficients correspond to the states in low dimensional subspace that are projected later back to the full dimensional space. For a new well location represented by parameter $\zeta^* \in \mathbb{P}$, the online procedure can be written as shown in Equation (39) and the projection of predicted coefficients back to fine scale solution is shown in Equation (40):

$$\mathscr{F}(\zeta^*, t_k) = [c_{\mathbf{x}i}(\zeta^*, t_k)]_{i=1}^{r} = \mathbf{c}_{\mathbf{x}}(\zeta^*, t_k) \quad (39)$$

$$\tilde{\mathbf{x}}(\zeta^*, t_k) = \sum_{i=1}^{r} c_{\mathbf{x}i}(t_k, \zeta^*)\phi_{\mathbf{x}i} + \bar{\mathbf{x}} = \Phi_{\mathbf{x}} c_{\mathbf{x}}(\zeta^*, t_k) + \bar{\mathbf{x}} \quad (40)$$

Note that here, $\bar{\mathbf{x}}$, which is the training snapshot mean, act as a physical constraint enforced on the predicted solution.

5.2. *Machine Learning and Feature Selection*

We are interested in understanding the relationship between the input parameters representing well locations and the basis coefficients of pressure and saturation. The inputs representing the well locations are defined in high-dimensional spaces as will be shown later, and the outputs, which are the POD coefficients, also are expected to be very high dimensional based on the number of basis functions chosen that increase with increasing number of training well locations and size of the reservoir. It can be shown that moving well locations just a by few grid blocks in heterogeneous reservoir can change the state solution siginificantly and hence the rates and thus, it cannot be obtained by a simple averaging of nearby well location solutions. Such high dimensional input-output relation and a highly non-linear relation that is very difficult to define explicitly calls for Machine Learning strategies. However, for engineering applications like the ones considered, it becomes very challenging to generate a large data set for the Machine Learning algorithms to capture the underlying dynamics. The re is trade-off that needs to be considered between getting a good training data set and the least computational expense at building a good predictor. Thus sometimes, specially for such applications as developing MOR for changing well locations in the reservoir, it is just beneficial to use simpler ML algorithms that are faster to train and can still capture the underlying non-linearity. Thus, we consider using Random Forest (RF) regressors as our algorithm of choice, since they are much faster to train and capable of mapping complex input-output relationships while using concepts like bootstrap aggregation to avoid overfitting. Other techniques like Artificial Neural Networks (ANN) have been used for non-intrusive ROM in Reference [28]. According to many applications, NN is found to be a feasible method where we have access to huge data sets and spending a lot of time in training is a viable option. We however, analyzed its performance as compared to that obtained by a fast algorithm like Random Forests (RF) and it was found very slow to train as well as did not lead to better accuracies as compared to RF. Other faster and simpler methods like kNN and multi polynomial regression were also considered but their performance was very poor for such complex problem and hence not shown here. We thus only show the results obtained with the RF models. A brief description about the Random Forest model is shown below.

5.3. *Random Forests (RF)*

Random Forest (RF) is a supervised machine-learning technique that constructs an ensemble of decorrelated decision trees. It is based on the concept of bootstrap aggregating (bagging) to reduce the high variance from single decision trees and further using decorrelated trees to induce more randomness and variance reduction that eventually improves prediction performance for test cases. Like ANN, RF can be used for both regression and classification.

We briefly describe the working of a RF here. The way a single decision tree is constructed is by using a recursive splitting strategy. At each node of a tree, best split of a predictor variable is made that results in two or more branches and then successively splits the predictor space at each step. For the regression problem, the best split is the one that leads to the maximum possible reduction in error Residual Sum of Squares (RSS), and for the classification problem, Gini index or the cross-entropy are generally used to evaluate the best split. Bagging generates new training sets by sampling the data with replacement and then we generate an ensemble of decision trees for each of these training sets. A Random Forest further decorrelates the trees by selecting only a random subset of features for building each tree. During prediction, for regression, the outcome is simply averaged over all the trees in the forest and for classification, the outcome is based on majority vote. For more details on Random Forests refer to [36]. Figure 1 shows the working of Random Forests pictorially. For this work, the hyperparameters for the Random Forest regressor model are chosen to be the minimum number of splits at the leaf node (N_l) and the number of features to search for split (N_{fmax}). For all the cases shown, the best model was found through cross-validation.

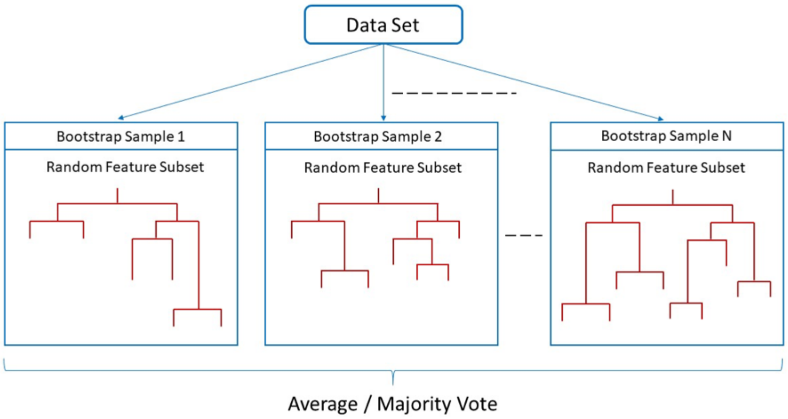


Figure 1. Description of Random Forests model where, each of the branches from sample dataset correspond to a bootstrap sample and a random feature subset is chosen for each branch to construct a decision tree.

5.4. Well Location Feature Determination

Before we employ the ML techniques to predict the basis coefficients, we need to quantify the features representing the parameter ζ. As we stated before, the system parameter ζ for well location optimization corresponds to the index in the source/sink term vector. The re is no explicit quantification of this parameter that represents changing well location in the PDE (Equation (3)). Thus, defining the input (well location)-output (basis coefficient) relationship is difficult. In this section, we first define geometric features for changing well locations. The n, we introduce some physics-based features using the concepts from flow diagnostics.

Geometric and Physics-Based Features

For the input features of prediction problems (Equation (37)), one of the obvious choices for changing well locations is the coordinates of the new well locations. We also consider the distance and angle between the wells (injectors and producers) in the reservoir. The se features are chosen based on the understanding that the state solution behavior to some extent is influenced by distance and orientation between the wells.

As we discussed before, the geometric features that correspond to changing well locations are deemed important for relation to the output state solutions (as represented by basis coefficients); we now define the physics-based features that are of major importance, as they describe the physics of flow and their relation to the well locations.

When we consider a heterogeneous reservoir case, one obvious choice of physics-based feature that need to be included is permeability of the grid block at the well location considered. What is also of major importance for us to understand the change in physics of flow with changing well location is some representation of well connectivity and flow heterogeneity between well pairs. In order to understand such properties, we now introduce the concept of flow diagnostics and various properties that are calculated during flow diagnostics that serve as important features for our problem.

5.5. Flow Diagnostics

Flow diagnostics are computational tools based on simple and controlled numerical flow experiments to help us quickly get quantitative and qualitative information regarding the flow patterns in a reservoir model. Traditional reservoir simulations can perform this task but computationally very demanding, and in contrast, flow diagnostic measures can be obtained within seconds. Thus, they are an inexpensive and reliable alternative to rank and/or compare realizations or strategies, and ideal for interactive visualization output due to computational advantage.

The reason behind this computational advantage is because the flow diagnostic tools provide quantitative information based on the steady-state flow. Flow diagnostic tools have been developed using streamline simulation techniques [37,38] that has shown to be very effective for various applications, but have some limitations in terms of computational complexity and extensibility. Most of the existing commercial simulators are based on finite volume methods capable of simulating fluid flow with different spatial grid geometry because of their mass conservative nature. Keeping this in mind, the flow diagnostic properties were obtained using standard finite-volume discretization in [39,40]. We use the flow diagnostics module of MRST (Matlab Reservoir Simulation Toolbox) [40] to evaluate these properties with the aim to aid in accurate construction of reduced-order models for changing well locations. We now very briefly introduce these parameters that lead to quantitative measures of flow dynamics as explained in the works [39,40].

5.6. Time of Flight

Single-phase pressure solutions are considered in the flow diagnostics study to get the time of flight (TOF) coordinate $\tau(x)$ which indicates the time it takes for a tracer to travel from the nearest injector to a given point x in the reservoir. We can derive the TOF equation in Eulerian coordinates using this identity:

$$\mathbf{v} \cdot \nabla\tau = \phi \tag{41}$$

Here, ϕ is the porosity of the reservoir and $\mathbf{v}$ is the Darcy velocity. The forward TOF τ_f is the time required for a tracer to reach at a given point in the reservoir after injected at an inflow boundary (injector in our case). The backward TOF τ_b is obtained by reversing the flux field which indicates the time required for a tracer released at a given location within the reservoir to reach a producer/outflow boundary. The sum of both these quantities gives the total time of flight also defined as the residence time of the particle.

For cases with more than one producer and one injector one can also consider the tracer distribution to identify quantities like well allocation factors, well-pair connections etc. as important features.

5.7. Lorenz Coefficient

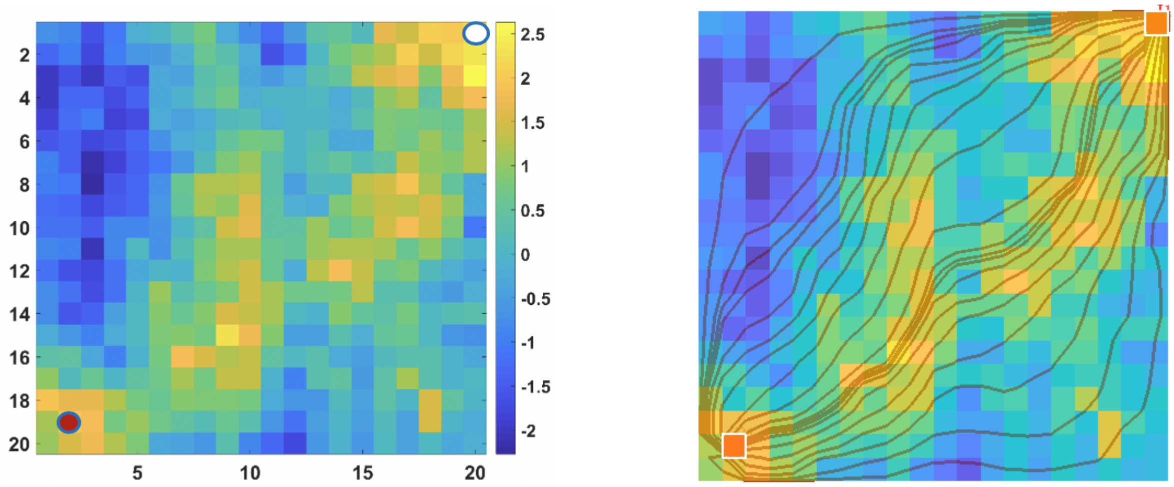
Time of flight and tracer concentration can also be used to assess the displacement heterogeneity of the reservoir using a quantity called the Lorenz coefficient. The heterogeneity is represented in terms of flow capacity-storage capacity ($F - \Phi$) diagrams. (Note that for the sake of consistency in notations with the literature, for this section we refer to Φ as the storage capacity and not to be confused with POD basis). This is equivalent to plotting fractional flow versus saturation for a 1D flow displacement. As defined in Reference [41], the storage capacity Φ is defined as the cumulative pore volume as fraction of total travel time τ, i.e., $\Phi(\tau) = \int_0^\tau \phi(x(\tau))d\tau$. The flow capacity is defined as the cumulative flux for increasing travel time. For an incompressible flow, $F(\tau) = \int_0^\tau \frac{\phi(x(\tau))}{\tau} d\tau$, since the pore volume equals the product of the flux and the total travel volume. Lorenz coefficient which is the measure of displacement heterogeneity is defined by measuring how much the flow capacity deviates from the ideal piston like displacement and hence can be written as:

$$L_c = 2 \int_0^1 (F(\Phi) - \Phi) d\Phi \tag{42}$$

This is twice the area under the $F - \Phi$ curve and above the line $F = \Phi$. Thus, it is 0 which corresponds to homogeneous displacement and unity for infinitely heterogeneous displacement. For a detailed explanation on calculating F and Φ, the reader is referred to [41] for streamline-based calculations and [39] for finite volume-based diagnostics.

5.8. Flow Diagnostic Features—Example

We now show an example of the flow diagnostic features for one of the well configurations. Consider the heterogeneous permeability field as shown in Figure 2a with wells at location $(2, 19)$ and streamlines imposed for this well configuration in Figure 2b. We use the MRST flow diagnostics module that computes TOF and Lorenz coefficient based on single phase pressure equations.

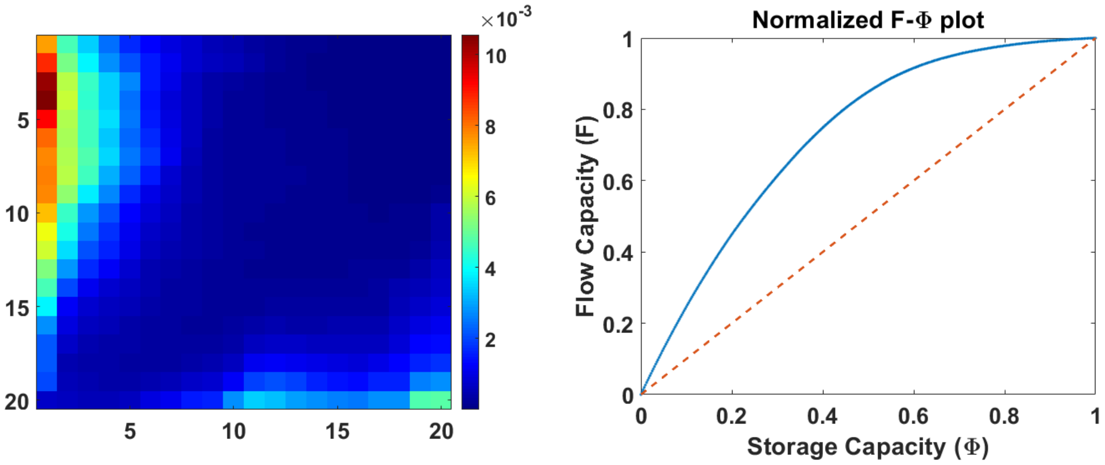


(**a**) Normalized Permeability Field

(**b**) Streamlines

Figure 2. (**a**) Heterogeneous channelized permeability field with one producer in red and one injector in white (**b**) Streamlines imposed on the permeability field.

It can be seen in Figure 3a that the low permeability area in the reservoir that is not well connected to the injector has a higher travel time and the producer location as shown above has a very low travel time as it is located on a high permeability grid block and connected to the injector via a channel. Thus, while considering the feature to be added for ML procedure, we use the TOF at the producer well location. Note that, since we are interested in finding time of flight at producer well location, it is sufficient to compute forward time of flight since backward time of flight will be negligible. Figure 3b shows the $F - \Phi$ plot to compute Lorenz coefficient as a measure of displacement heterogeneity indicated by deviation from the $F = \Phi$ line.



(**a**) Total travel time

(**b**) $F - \Phi$ plot to compute Lorenz coefficient

Figure 3. (**a**) The total travel time of the particle injected at injector to any point in the reservoir (**b**) $F - \Phi$ plot to compute Lorenz coefficient as the area under the solid $F - \Phi$ curve and the dashed line $F = \Phi$.

The set of features used is listed in Table 1. Note that the features correspond to information that we think best describes the well location. We also include time information to account for understanding the temporal behavior of the POD basis coefficients.

Table 1. Geometric and physics-based features for ML model construction corresponding to the well configuration.

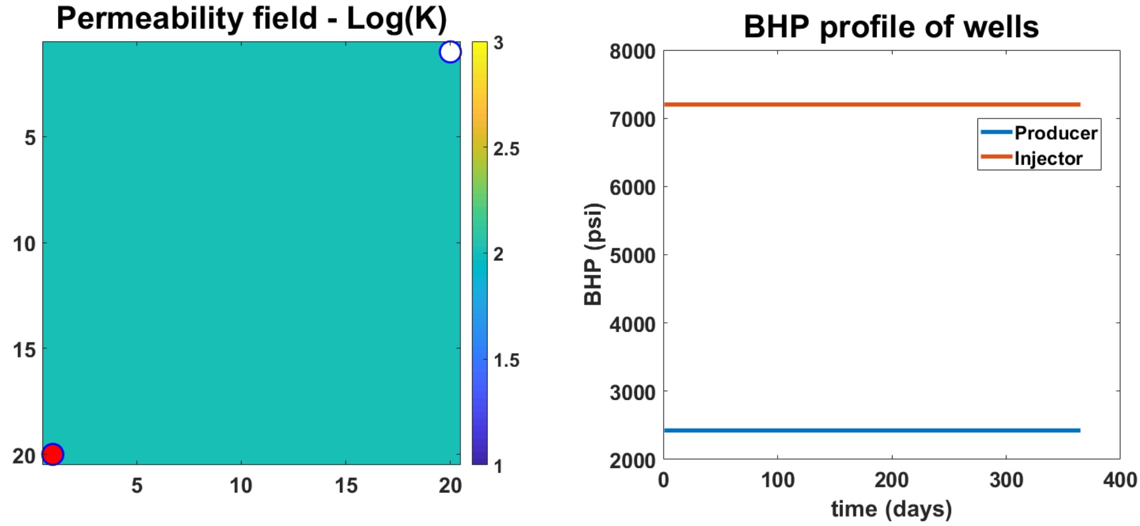
Feature Set for a Well Configuration
x—X coordinate of the well
y—Y coordinate of the well
r—Distance between well and injector
θ—Angle between well and injector
K—Permeability at the well location
TOF—Total time of flight at the producer well location
LC—Lorenz coefficient for the well configuration
$index$—Well grid block number in the reservoir
t—Time at which the POD coefficients are computed

5.9. Remarks

As we have formulated the method, we now demonstrate its performance with some case studies. However, before that, we point out some specifics of these cases. The injector for all the cases shown here is at a fixed location and the producer location is considered as the parameter i.e., only producer can change location in the reservoir. All the new producer well locations have the same BHPs to consider only the well location as a system parameter. The workflow can be extended in the future to also include varying well BHPs as the feature but not in the current scope of work. The simulations are run long enough to observe significant water cut at most of the well locations in order to introduce complexity to the problem which was a limitation in the local ROB-based workflow [30]. For the cases shown below, we use random sampling in the parameter space to train the ML models. Experimental design sampling techniques or greedy sampling procedures as listed before can be used for sampling the parameter space for better representative parameters, but, for simplicity and fast sampling we choose the parameters randomly for training purpose. However, the number of random samples chosen is an open challenge that needs to be addressed in the future. With random sampling we usually expect a lot more samples than something like greedy sampling procedures, in efforts to capture representative samples for global ROB. The main aim here is to show the validity of proposed methodology given the high complexity of the problem compared to that of changing well control problem and hence we use relatively large training sample sizes here. With the feature set that we propose for well locations, we observe that, the feature selection strategies like the wrapper approach [42] sometimes selects all the features depending on the data set and the black-box model used. This suggests than no features in this case are correlated. So while training the models, we use all the features and then for feature importance, we try to tune better the hyperparameters and regularization parameters.

6. Case Study and Results

In this section, we present the numerical results for the proposed non-intrusive global PMOR technique. We start with a simple example of a homogeneous reservoir model. The model has 20 × 20 grid blocks and has one injector and one producer well, as shown in Figure 4a. The porosity is set constant to 0.2 and the relative permeability model is defined by Corey function of degree 2. The permeability of the field is considered to be 100 mD. The 2-phase flow is considered incompressible neglecting the capillary and gravitational effects. All the producers are produced with a constant BHP of 2425 psi and injector injects water a constant BHP of 7200 psi (Figure 4b). We look to change the producer locations only and the injector location is fixed at $(20, 1)$ grid block. For all the cases in this paper, we assume fixed injector location while only changing producer location. The simulations are run for a period of 1 year with which we observe watercut at all the producer locations. This is the example where Productivity Index (PI) for all well locations will be the same and has a less magnitude. The case of well locations with high PI will be shown next with a channelized reservoir.



(a) (b)

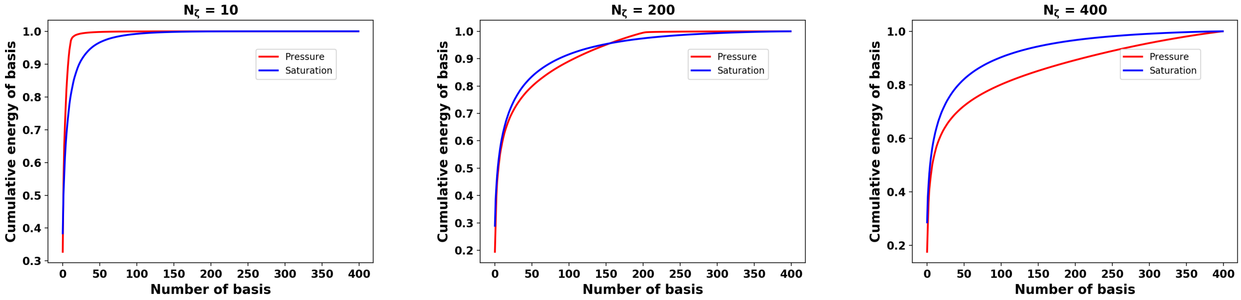
Figure 4. (a) Homogeneous permeability reservoir with 1 injector (white) and 1 producer (red). Producer is the parameter and can change location anywhere in the reservoir (b) BHP profiles of injector and producer that is set constant for all well configurations considered.

The problem is set up as follows. First, in order to construct a global ROB, we randomly sample the well locations. Fine-scale simulations are run for each well location and the snapshots are collected in a single snapshot matrix as shown in Equation (27). We then follow the procedure as described before to train the ML model. Since, we have 400 grid blocks, there are 399 locations we consider where the producer can move (1 injector block). So we randomly sample 100 cases to train the ML models and the rest 299 cases are used as test cases.

6.1. Energy of Eigenvalues—Global PMOR

Now, we discuss about an observation on energy of eigenvalues that is important in understanding the complexity of developing PMOR strategy for changing well location using this case of homogeneous reservoir. For problem of changing well controls and fixed well configuration, we usually observe higher number of saturation basis as compared to pressure basis due to the fast moving pressure front and slow saturation front. Here, for changing well location problem, as we construct the global PMOR by concatenating solutions from different well configurations, as expected, we will see an increase in the number of basis for both pressure and saturation. However, we look how these eigenvalues decay or rather how the energy of the basis change with increasing number of parameters in the snapshot matrix.

PMOR for well control usually needs a lot more saturation basis as compared to pressure basis for a fixed well configuration, as the pressure behavior has a fast-moving front and is not expected to change significantly for a fixed well location. However, the saturation behavior has a very slow moving front with major updates occurring at the saturation front. For the problem of changing well locations, at each well location we have the lowest or highest pressure point in the reservoir corresponding to the well BHP and hence, reducing the pressure solution for changing well locations is extremly challenging. This is evident in the observation shown as follows. A typical energy criteria used for choosing the basis dimensions is above 90% of the energy of eigenvalues from the Equation (19). As can be seen in Figure 5, we begin with $N_\zeta = 10$ where, as usual, we have a higher number of saturation basis than pressure basis satisfying more than 90% energy criteria. However, after adding more parameters, these trend changes, especially after $N_\zeta = 200$. It requires significantly more number of pressure basis to capture to understand the dynamics. Note, that the pressure basis energy plateaus at a point equaling number of training parameter N_ζ. As we add more training parameters, we see that the saturation basis energy profile does not change significantly. Thus, for many cases at hand, depending on the training parameters selected, we may observe a higher number of pressure basis as compared to saturation basis. This observation also gives a notion on the complexity of the problem. The re is a trade-off between choosing the number of training samples and the basis dimension required for a good quality basis over the domain of parameters. For example, we may choose 10 training samples and capture 99.99% energy basis but still may not produce accurate results as each parameter has very different dynamical behavior, especially at the well location. So it may sometimes be good to have more training samples, to capture the pressure behavior as close as possible at the new well location (although we know it will still not be the lowest pressure point as we see in the Figure 5). This implies a lower energy criteria at the expense of running more fine-scale simulations in the training phase and obtaining significant speed-ups in the testing phase.

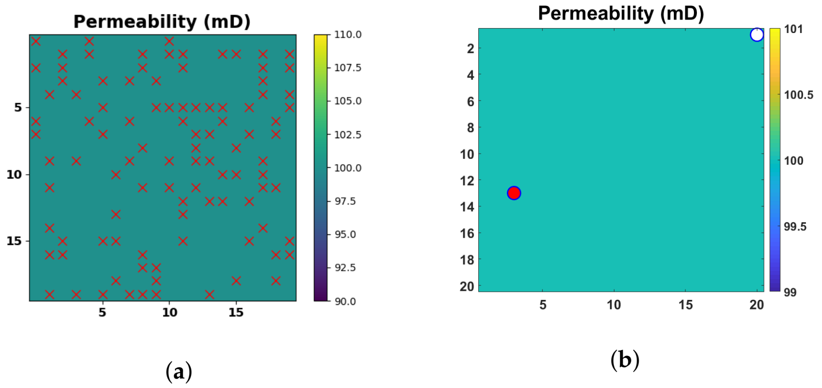


(**a**) (**b**) (**c**)

Figure 5. Change in cumulative energy of the basis, as calculated by magnitude of eigenvalues for increasing number of parameters (well configurations) (**a**) $N_\zeta = 10$, (**b**) $N_\zeta = 200$, (**c**) $N_\zeta = 400$, in the state snapshot matrix

6.2. Results

We now show the prediction results for new well locations not included in the training set. Figure 6 shows the training samples of producer well locations, each simulated one at time, and the new well location in the first test set for which we predict the POD coefficients.



(**a**) (**b**)

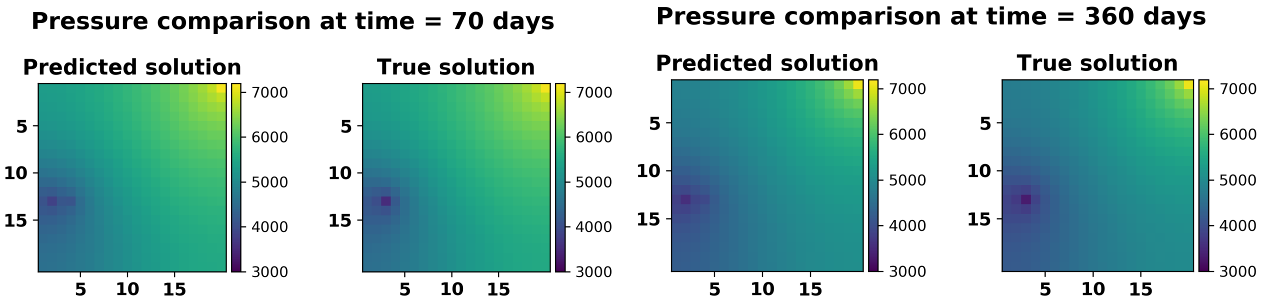
Figure 6. (**a**) Training samples of the producer well locations shown by red crosses. (**b**) Test case 1 well location at (3,13) in a homogeneous reservoir.

First, we use Random Forest regressor to compute the POD basis coefficients. Note that the RF model here is a multi-output model, with leaf nodes corresponding to the multi-dimensional basis coefficients. By doing so, we try to preserve the structure of the solution rather than having a single model for each basis coefficient acting independently. For pressure, the basis dimension is 99 and that for saturation is 113, which corresponds to the output dimension for ML models of pressure and saturation respectively. The outputs are predicted using the relation in Equation (39), and for this example the optimal tuning parameters for RF regressor found are shown in Table 2.

Table 2. Hyperparameters chosen by 5-fold cross-validation for Random Forest Regressor using 100 training samples.

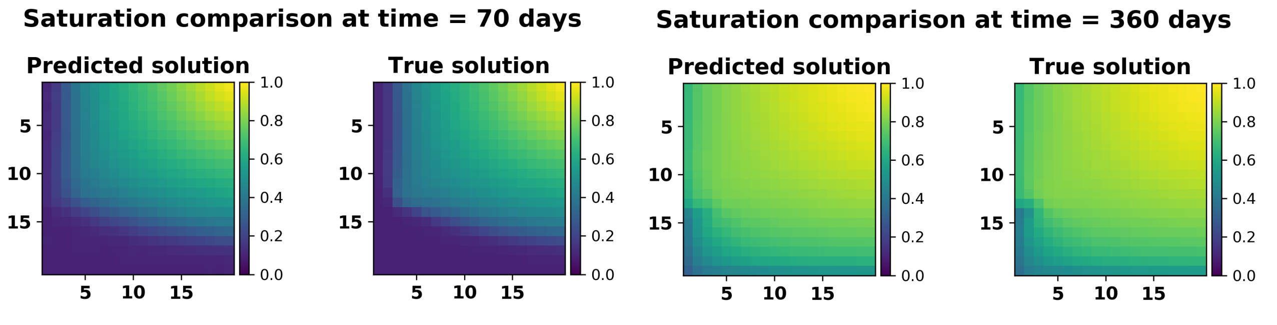
	RF Regression	Train Accuracy	Test Accuracy
Pressure	N_{fmax}=3, N_l=2	99.83	98.18
Saturation	N_{fmax}=3, N_l=3	98.89	93.21

Figures 7 and 8 show the comparison between the predicted and true states (pressure and saturation) for the new well configuration at two different times. We also show the error in states along time. As can be seen in Figures 9 and 10, prediction of the overall state behavior is fairly accurate but there is discrepancy in the solution close to the producer well location for pressure, and at the fluid front for saturation.



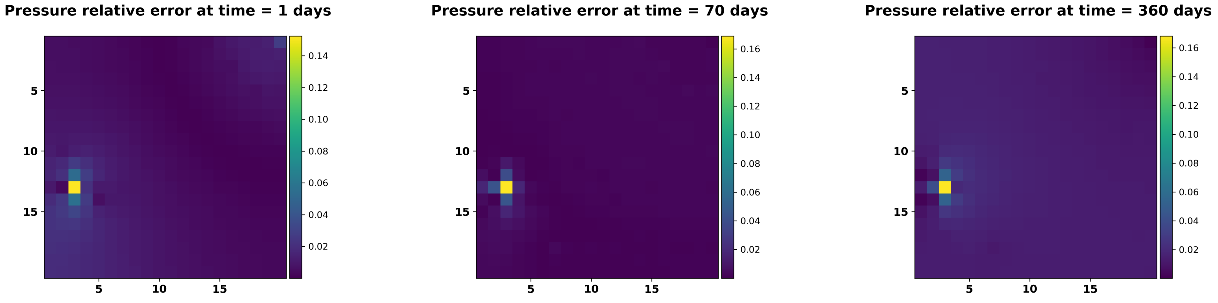
(**a**) (**b**)

Figure 7. Pressure solution comparison at (**a**) Time = 70 days and (**b**) Time = 360 days.



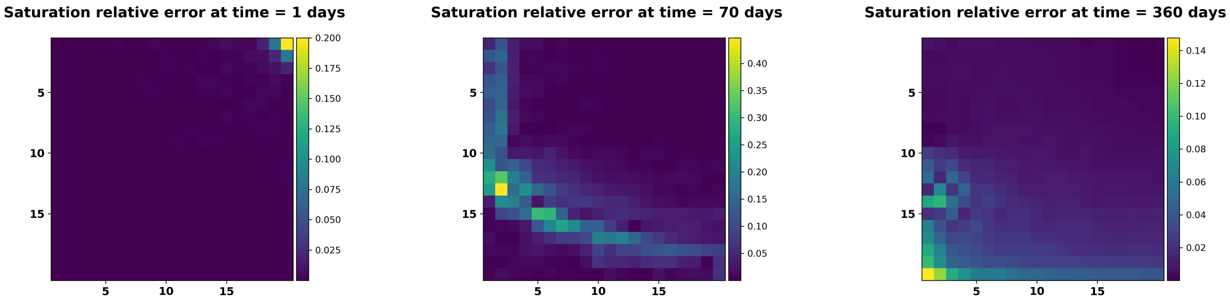
(**a**) (**b**)

Figure 8. Saturation solution comparison at (**a**) Time = 70 days and (**b**) Time = 360 days.



(**a**) (**b**) (**c**)

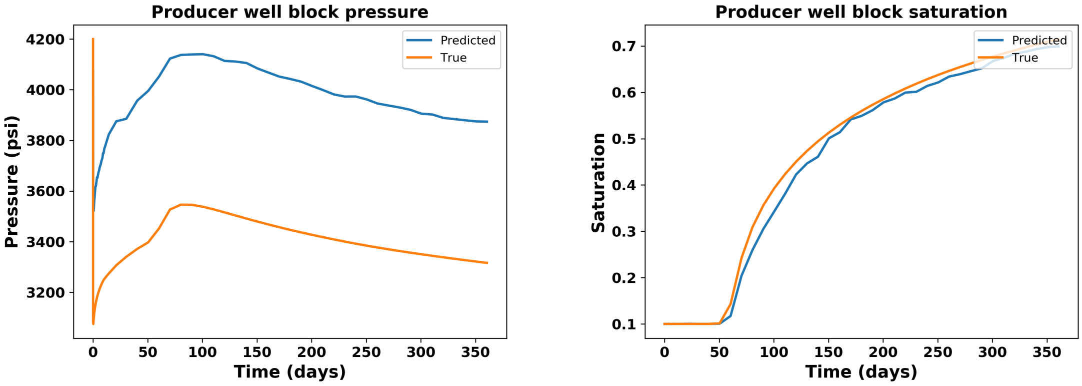
Figure 9. Relative error in pressure at time = (**a**) 1 day, (**b**) 70 days and (**c**) 360 days.



(**a**) (**b**) (**c**)

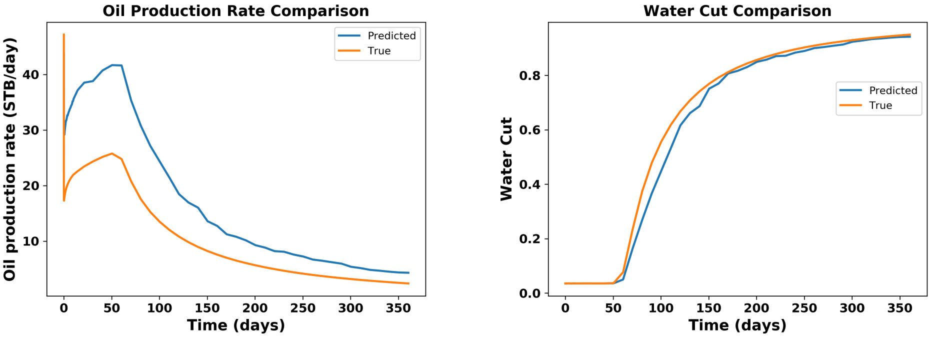
Figure 10. Relative error in saturation at time = (**a**) 1 day, (**b**) 70 days and (**c**) 360 days.

The quantities of interest that are the outputs of the simulation, such as fluid production rates, water cut, etc., are solely dependent on the state quantities at the well blocks through the Peaceman Equation (4). Thus, for this prediction results we expect some error in these quantities due to higher prediction error close to the wells. To quantify these discrepancies, we show the pressure and saturation solution comparison at the producer well block in Figure 11. As we can see, the predicted solution captures the overall trend of dynamical behavior of the states but has kind of a bias in the solution. Figure 12a,b show the comparison of oil production rate and water cut between the true and predicted solution. Since this is the case of well locations with low PI, there is not a big discrepancy seen in the production rates, but this discrepancy is expected to increase significantly for the cases with high PI or high permeability reservoirs as will be seen in the heterogeneous case. This is the motivation to adopt a second step in this methodology, where we use this discrepancy in the reduced-order solution as an input to build a correction model outlined in the next section.



(**a**) (**b**)

Figure 11. Well block state solution comparison (**a**) Pressure and (**b**) Saturation.



(**a**) (**b**)

Figure 12. Quantities of Interest comparison (**a**) Oil production rate and (**b**) Water cut.

We also consider a second test case with a new producer well location, as shown in Figure 13, to validate the method. Figures 14 and 15 shows the pressure and saturation comparison between ML predicted and true solutions. Both the solutions show a very good agreement visually. For a more detailed analysis, we also plot the well block pressure and saturation in Figure 16a,b. This case shows that ML predicts the solution with a very good accuracy which eventually reflects in the Figure 17, where we plot the oil production rate and water cut.

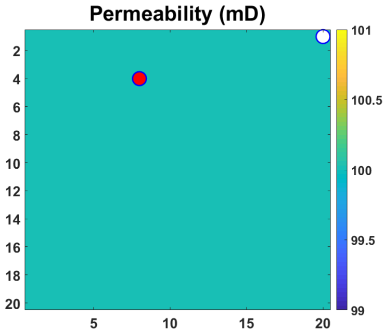
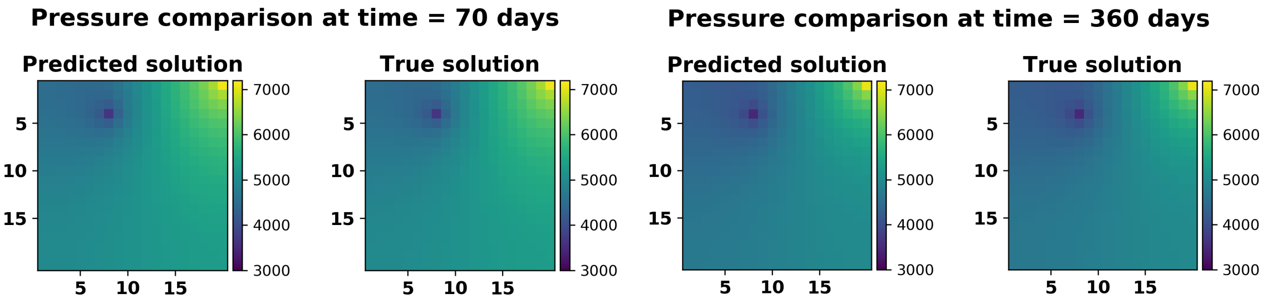
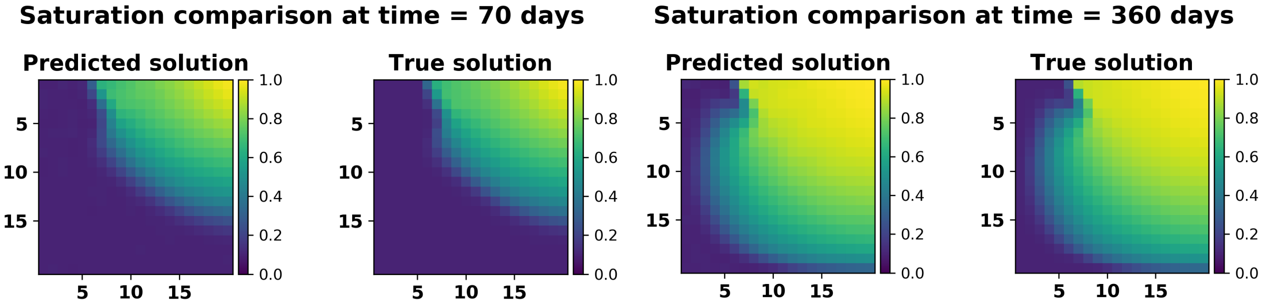


Figure 13. Test case 2 with producer at (8,4) grid block and injector at (20,1) grid block in homogeneous permeability reservoir.



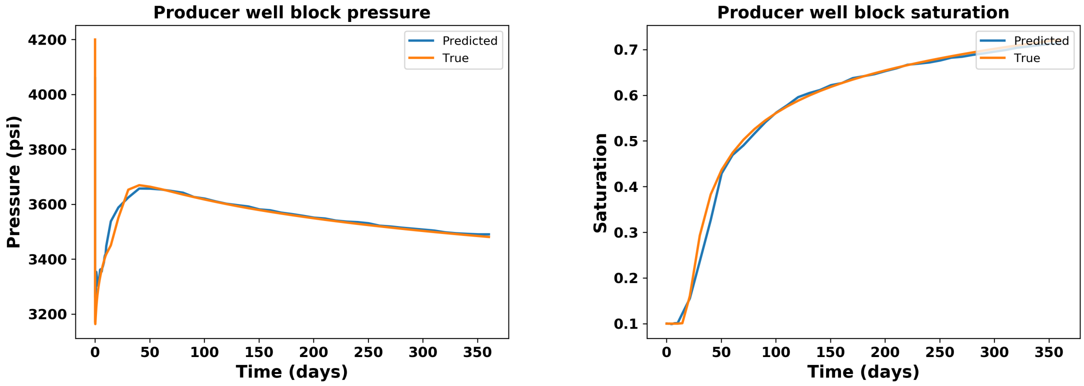
(**a**) (**b**)

Figure 14. Pressure solution comparison at (**a**) Time = 70 days and (**b**) Time = 360 days.



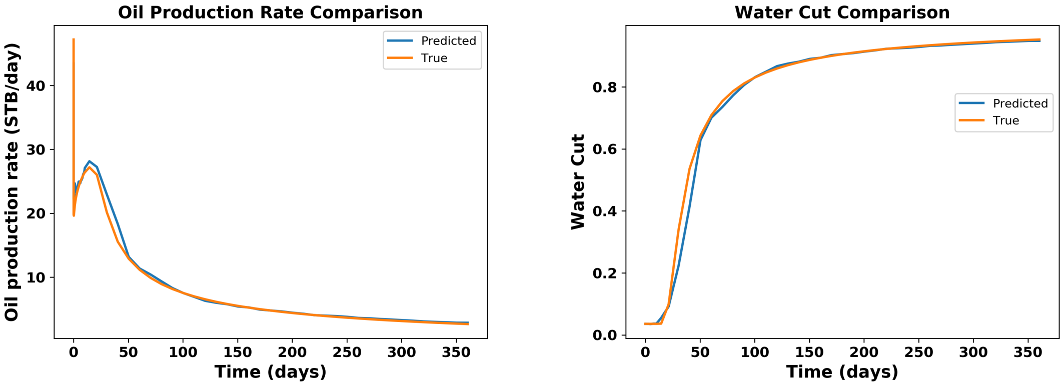
(**a**) (**b**)

Figure 15. Saturation solution comparison at (**a**) Time = 70 days and (**b**) Time = 360 days.



(**a**) (**b**)

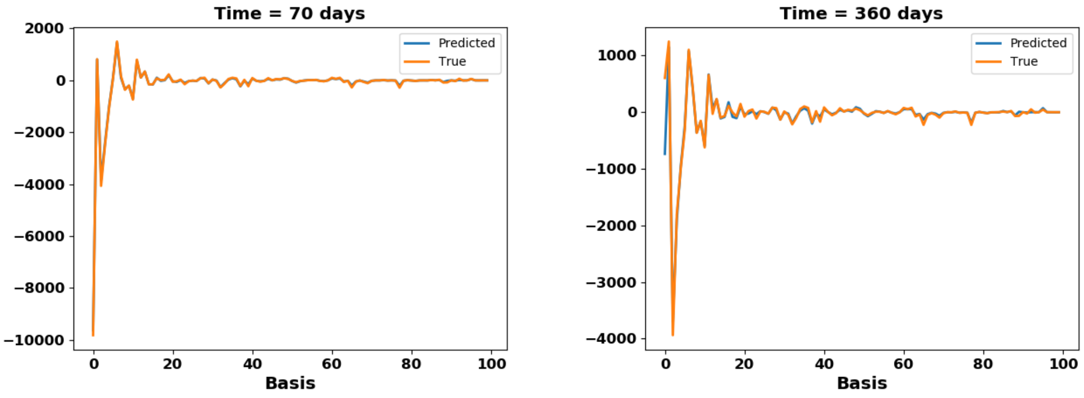
Figure 16. Well block state solution comparison (**a**) Pressure and (**b**) Saturation.



(**a**) (**b**)

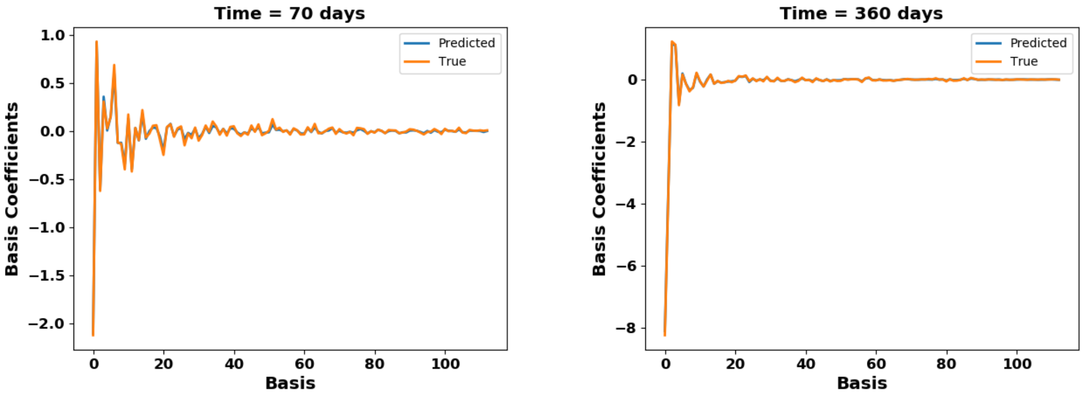
Figure 17. Quantities of Interest comparison (**a**) Oil production rate and (**b**) Water cut.

After plotting the results from other test cases, we found that, for some cases, the prediction is reasonably accurate, similar to that in test case 2. However, there are many well configurations that show discrepancies, such as test case 1, which overall captures the solution trend, but there is some bias associated with it. Thus, we need to analyze the reason behind the solution discrepancy which can either be due to the error in Machine Learning model or due to the quality of global basis. In order to get an intuition about these two factors, we first analyze the Machine Learning model performance. In Figures 18 and 19, we compare the predicted and true basis coefficients for pressure and saturation respectively for different time steps for test case 1. The se show that the ML model predicts the coefficients with a good accuracy except for some instances like the first pressure basis at time 360 days. The true coefficients for a new parameter ζ^* are obtained by projecting the fine-scale simulation on the subspace spanned by global basis Φ by computing $\Phi_{\mathbf{x}}^T \mathbf{x}(\zeta^*)$.



(**a**) (**b**)

Figure 18. True and ML predicted pressure basis coefficient comparison at time = (**a**) 70 days and (**b**) 360 days.



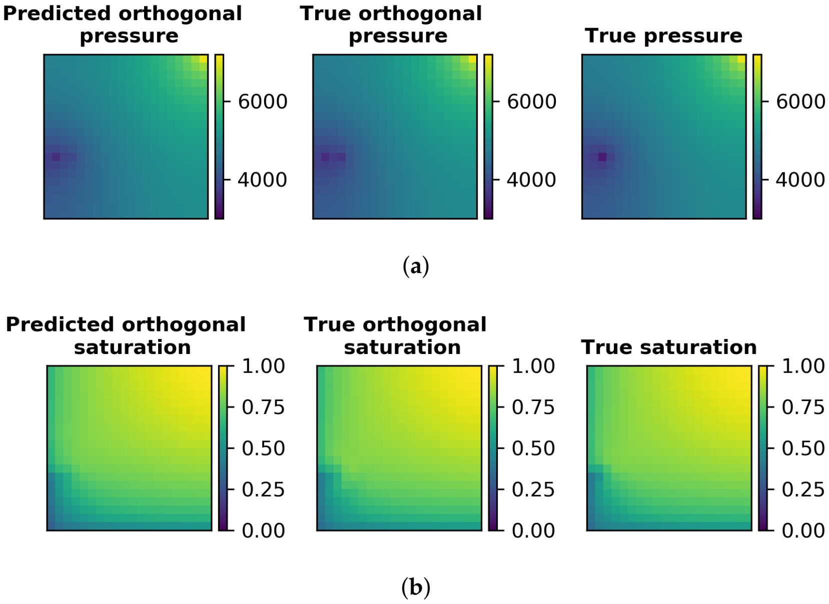
(**a**) (**b**)

Figure 19. True and ML predicted saturation basis coefficient comparison at time = (**a**) 70 days and (**b**) 360 days.

Thus, the performance of the ML model suggests that the discrepancies near the well locations are less likely due to the errors in ML model, specially the fact in understanding that the new producer well location should be the lowest pressure point in the reservoir. Thus, now we analyze the quality of the global basis Φ used in test case 1. Here, we look back at Section 3, where we introduced the concept of MOR error that has a orthogonal component which basically results from neglecting the state projection on the orthogonal subspace. At a given time instant for a new parameter ζ^*, it can be written as shown in Equation (43):

$$\begin{aligned} \epsilon_{\Phi^\perp}(t,\zeta^*) &= (I_n - \Pi_{\Phi,\Phi})\mathbf{x}(t,\zeta^*) \\ \Pi_{\Phi,\Phi} &= \Phi\Phi^T \end{aligned} \tag{43}$$

Thus, the deviation of the product $\Phi\Phi^T$ from the identity matrix gives an a priori estimation of the quality of basis for the new parameter. To visualize this for our case, we project the true coefficients obtained by $\Phi_{\mathbf{x}}^T\mathbf{x}(\zeta^*)$ back to the fine-scale domain. This is basically performing the operation $\Pi_{\Phi,\Phi}\mathbf{x}(t,\zeta^*)$, which we refer to here as the true orthogonal solution. The Figure 20a,b show the predicted orthogonal solution, true orthogonal solution, and true fine-scale solution for pressure and saturation, respectively, at the end of simulation for test case 1. As can be seen, the true orthogonal pressure and saturation solutions differ from the true solutions around the producer well. This is clearly visible for the case of pressure solution. However, the predicted orthogonal pressure and saturation solutions are in a very good agreement to their true orthogonal solutions. This is an indication of the quality of basis that is the main reason behind the higher errors around producer well locations. This indicates that even with a global basis of parameters, it is very difficult to preserve the controllability properties for all the new well locations. The global basis here proves to be a good quality basis for some cases like the test case 2. Thus, the best possible way to alleviate this problem is to try getting the most representative set of parameters in the training sample set, which can be a future direction of research.



(**a**)

(**b**)

Figure 20. Comparison of predicted orthogonal and true orthogonal solutions with true solution at time = 360 days for (**a**) Pressure and (**b**) Saturation.

The first step to predict the POD basis coefficients; thus, we do not expect to predict the state solution accurately, especially in cases of high PI or high well block permeability. However, with a good quality POD basis we can obtain a model to understand the trend of the expected solution.

7. Modeling Error Correction Concept

As we saw, finding a good global POD basis for the entire domain of well locations is very difficult due to the nature of pressure solutions with changing well locations. However, from above analysis we observed that by getting a good enough training set to compute the POD basis, the overall trend of the state solution is captured. Thus, with the given information about the reduced-order model, we propose the second step, which is building an error correction method to adjust the solution discrepancies. The error correction model takes as input the reduced-order model solution at a new parameter and then predicts the correction required in the quantity of interest. This model is constructed using Machine Learning techniques to account for high dimensional feature space and mapping the complex non-linear relationship between inputs and outputs. By introducing the reduced model solution in the feature set, we include physics-based information for the model to understand the expected error. The quantity of interest can be the oil/water production rate, water cut or the well block states. For the current work, we just consider the well block states (pressure and saturation) as the QoI to be corrected as these states directly transfer to the correction in production rates and water cut. At a given instant in time, the error in the well block states is:

$$\begin{aligned} &\Delta \mathbf{x}_{wb}(\zeta, t_k) = (\mathbf{x}_{wb}(\zeta, t_k))_{fine} - (\mathbf{x}_{wb}(\zeta, t_k))_{PMOR} \\ &\text{where,} \quad k = 0, 1, ..., N_t \ \text{ and } \ \zeta \in \mathbb{P} \end{aligned} \tag{44}$$

Thus, we are interested in determining $\Delta \mathbf{x}_{wb}$ for a new well location over time, given the reduced model solution obtained in Equation (40). We also note that the saturation map comparison at different times during the prediction (Figure 8) and comparison of the predicted injection rates with true injection rates showed good agreement. This information helps us consider the pore volumes injected predicted by the reduced model as a good reduced model information. The formulation of the error correction model is thus given by:

$$\mathscr{E}(\zeta, t_k, \tilde{\mathbf{x}}, PVI_r) \rightarrow \Delta \mathbf{x} \tag{45}$$

This model is constructed using ML techniques, where the ζ corresponds to the well location parameter and hence is represented by the same features as used in the global PMOR formulation. $\tilde{\mathbf{x}}$ correspond to the PMOR predicted solution at all the grid blocks in the reservoir and PVI_r is the pore volumes injected, predicted by the reduced-order model. However, since we are interested in the correction of states at the well grid block only, it is worthwhile to consider the relation between the reduced solution at grid block location $\tilde{\mathbf{x}}_{wb}$ to the correction $\Delta \mathbf{x}_{wb}$. By this assumption, the formulation changes to:

$$\mathscr{E}(\zeta, t_k, \tilde{\mathbf{x}}_{wb}, PVI_r) \rightarrow \Delta \mathbf{x}_{wb} \tag{46}$$

This formulation is thus a global error model that is used for the entire parameter space. A similar idea was implemented to construct local error models for POD-TPWL method [43] that showed promising results for well control changes but can be computationally expensive to construct. However, the global error model is used here to avoid further computational complexity to the already expensive PMOR training procedure. In cluding the reduced-order model solution to the input features achieves good accuracy as we take into account the physics of ROMs rather than constructing completely data driven error models as in [7,8]. Similar to the cases of constructing error maps using ML, we have the single output here corresponding to state error at well grid block. For such cases, NN proved to be a better model at mapping the complex input-output relation and is faster to train for a single output system rather than using it for the prediction of high dimensional POD coefficients in global PMOR formulation. A brief description of Neural Networks is provided below.

Neural Networks

Artificial Neural Networks are nonlinear statistical models that detect patterns in data by discovering the input-output relationships, used for both regression and classification. The choice of ANN model is motivated by its capability to capture highly nonlinear complex relationship between the input features and the output. A feed-forward neural network used here consists of L layers with each layer consisting of predefined nodes and an input layer consisting of the features or independent variables used for prediction. Each of these layers have an associated transfer function and the nodes are connected to the nodes from previous layers by weights. A simple mathematical description is as follows:

$$Z^{l+1} = \Theta^{l+1} X^{l} \tag{47}$$

where Θ^{l+1} is the weight matrix connecting the nodes between layer l and $l+1$. X^l is the output of the layer l and Z^{l+1} a simple linear regression of these outputs. Then, the output of the layer $l+1$ is given by:

$$X^{l+1} = G(Z^{l+1} + B^{l+1}) \tag{48}$$

B^{l+1} is the bias term added to each node. The non-linearity of the model arises from the activation function G. The most widely used activation functions include sigmoid, hyperbolic tangent, and rectified-linear unit (ReLU). For the regression problem here, we use hyperbolic-tangent activation in the hidden layers (as they work better in most cases than a sigmoid function) and simple linear regression in the output layer.

The model is trained using backpropagation and optimization algorithms to adjust the weights Θ of the network to minimize the cost function. A detailed description on ANN and backpropagation algorithm can be found in [44]. Figure 21 shows a schematic of ANN. The neural network usually tend to overfit the data and thus, we use a regularized cost function for regression given by:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right] \tag{49}$$

Here, m represents the total number of data examples, $h_\theta(x)$ represent the predicted output of the network and y is the expected output. θ represents each of the elements of the weight matrix Θ in each layer and λ is the regularization parameter to control feature importance and prevent overfitting. We use λ and number of units in each layer as the tuning parameters while training the ANN. The values of these hyperparameters are chosen here using K-fold cross-validation technique. For the case studies considered here, the hyperparameter search space for the error NN model was chosen to be between 5–20 nodes per layer and regularization parameter ranging from 0.01–100 spaced by order magnitude differences. We search within finer partitions around the best found regularization parameter after this initial analysis.

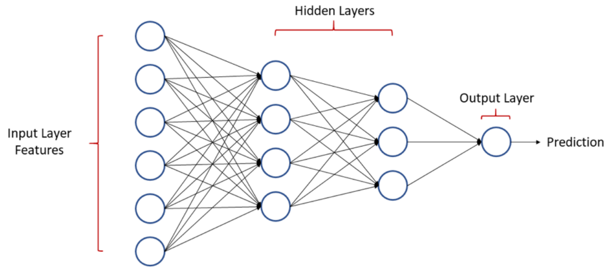
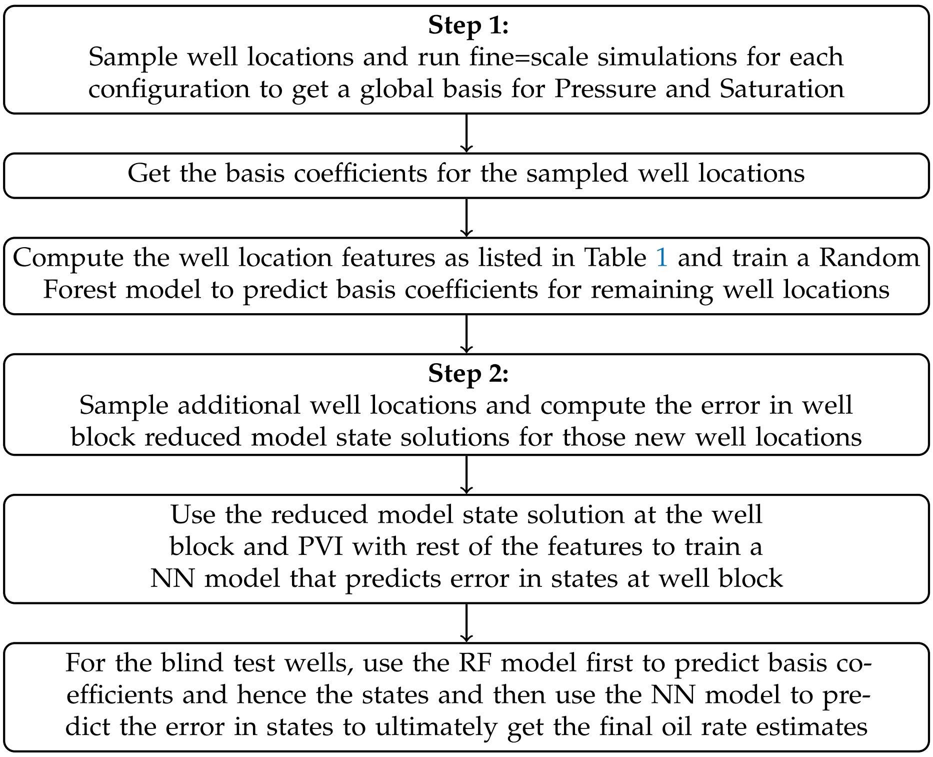


Figure 21. Schematic of a feedforward Artificial Neural Network.

8. Case Study and Results

8.1. Homogeneous Reservoir Model

We use the error correction model now in an effort to minimize the solution discrepancies at the well locations. We use the same case study as before for the homogeneous reservoir model and changing producer well locations. In order to construct the error model, we sample randomly other 100 well locations to generate the data set that account for running 100 fine-scale simulations. In order for the ML models to capture the underlying behavior of the system, it requires many data points especially to understand such complex relationships. We realize the computational expense associated with fine-scale simulation runs, but better sampling strategies should be the research focus in the future, that can reduce the number of sampling points. Additionally, the current case is a small model and hence the number of sampling points relative to the size of the problem is high here. However, as we move to the bigger reservoir models, we expect much lower sample points relative to the reservoir size. Figure 22 shows the workflow summary of proposed PMOR procedure which we follow for the cases considered.



Step 1:
Sample well locations and run fine=scale simulations for each configuration to get a global basis for Pressure and Saturation

Get the basis coefficients for the sampled well locations

Compute the well location features as listed in Table 1 and train a Random Forest model to predict basis coefficients for remaining well locations

Step 2:
Sample additional well locations and compute the error in well block reduced model state solutions for those new well locations

Use the reduced model state solution at the well block and PVI with rest of the features to train a NN model that predicts error in states at well block

For the blind test wells, use the RF model first to predict basis coefficients and hence the states and then use the NN model to predict the error in states to ultimately get the final oil rate estimates

Figure 22. Workflow of the ML-based predict and correct PMOR procedure to estimate well production rates for new well locations.

Thus, for the reservoir description shown in Figure 4, we have 100 well configurations used to construct the global PMOR model and 100 other well configurations for training the error correction ML model. Thus, we have 199 test cases in total to test the performance of this methodology. Figures 23–25 show the comparison of the results obtained by just implementing the non-intrusive global PMOR method, its implementation with the error correction models and the true solution obtained by fine-scale simulation.

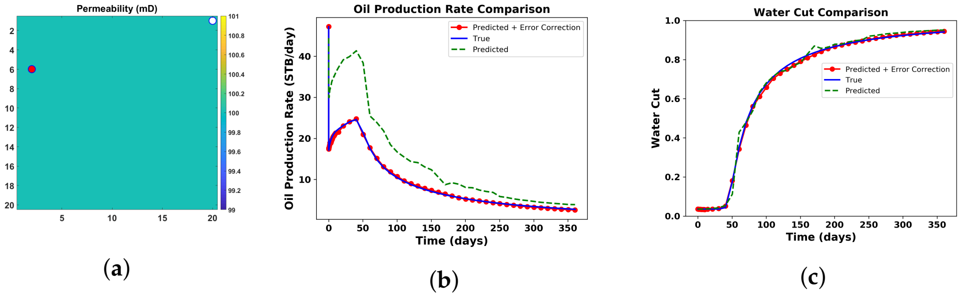


Figure 23. (**a**) Test case with a producer well location at (2,6). (**b**) Comparison of oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone using 100 samples (dotted green line) and after the implementation of the error correction model (red circled line) with the true solution (blue line).

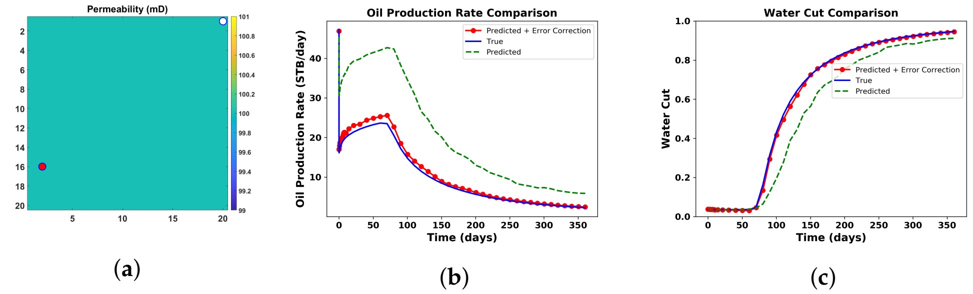


Figure 24. (**a**) Test case with a producer well location at (3,16). (**b**) Comparison of the oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of the error correction model (red circled line) with the true solution (blue line).

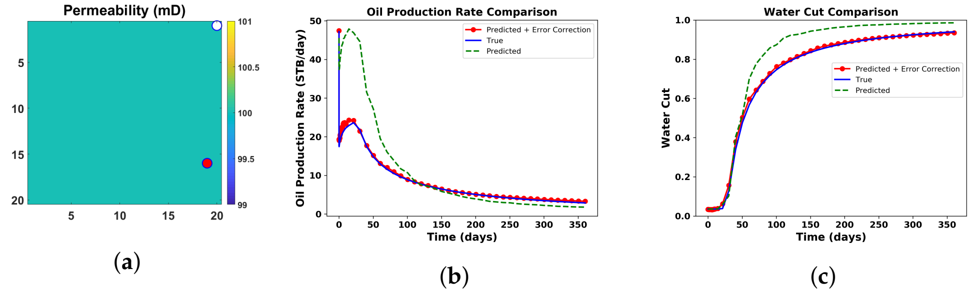
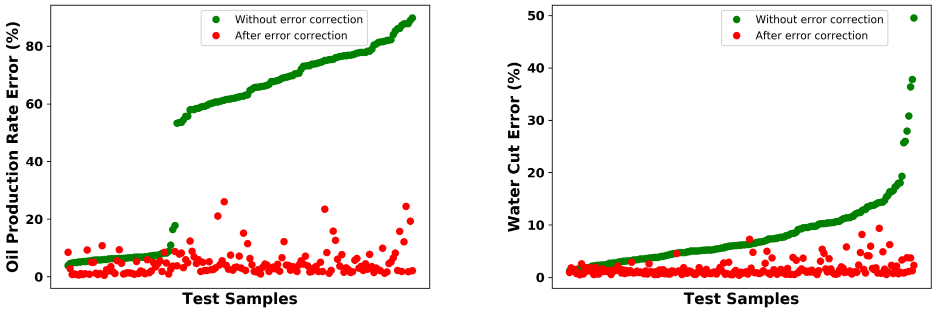


Figure 25. (**a**) Test case with a producer well location at (16,19). (**b**) Comparison of the oil production rate and (**c**) comparison of water cut, predicted using global the PMOR method alone using 100 samples (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).

It can be observed from these test cases that the proposed method shows a very good accuracy with the solution trend captured by global PMOR and then adding error correction to the solution to get a much improved accuracy. This was examined for the other test cases as well. The predicted results accurately captures the water breakthrough time which is different for different test cases, as can be seen in the plots of water cut comparison.

We predicted the solutions for all the test cases and show the errors in prediction for all these cases in Figure 26 for oil production rate and water cut. In this figure, the errors shown in red are those obtained by POD coefficient prediction without error correction. These are arranged in increasing order for all the test cases. The blue dots show error after correcting the solutions in the second step for each corresponding red dot. This gives an intuition of the behavior of proposed method for all kinds of test cases. This plot shows that most of the test cases show a very good accuracy after error correction even when the predicted solutions just based on ML estimated POD coefficients have large errors. There are a very few test cases that show errors increasing after correction, that most probably can be attributed to random sampling, and should be solved after efficient sampling techniques. The average accuracy for all the test cases can be found in Table 3. Table 4 shows the optimal hyperparameters of the NN model through grid search.



(**a**) (**b**)

Figure 26. Error in the prediction of (**a**) the oil production rate and (**b**) water cut for all the test cases before and after the error correction of the solutions.

Table 3. Homogeneous reservoir case 1: the average accuracy of the oil production rate and water cut for all test samples.

	Average Accuracy (%)
Oil production rate	95.28
Water cut	98.4

Table 4. Error NN model Hyperparameters found through grid search for homogeneous reservoir case.

Number of Nodes	**L2 Regularization Parameter**
8	0.1

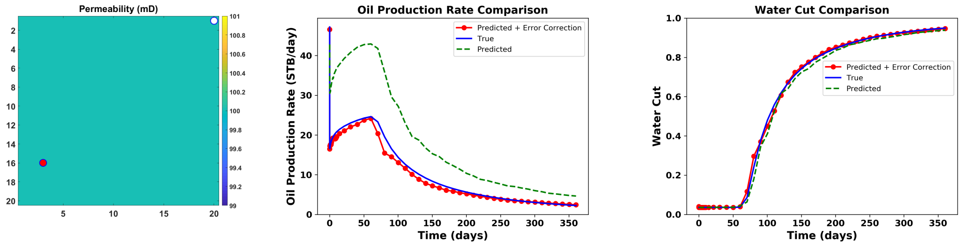
Now, we show the same example, but using fewer training sample well locations for predicting the basis coefficients and the construction of global ROB (this should be a lower-quality basis than before) and using same number of training points for the error correction model. Since it is challenging to pick the number of sample points for training and ultimately choosing appropriate basis dimensions, such an analysis is useful to determine if the error correction model can provide reasonable results even if the global ROB is not a very good representation of the parameter domain. The ML-based PMOR model is thus constructed with 50 sample well locations for this example.

Table 5 shows the optimal hyperparameters and the corresponding train and test accuracies obtained from the trained Random Forest model to predict the basis coefficients for both pressure and saturation. As can be expected, the train accuracy decreases as the ML model does not have a large data set to learn the physics which eventually also shows a decrease in the test accuracy as compared to the previous test case.

Table 5. Hyperparameters chosen by 5-fold cross-validation for Random Forest Regressor using 50 training samples.

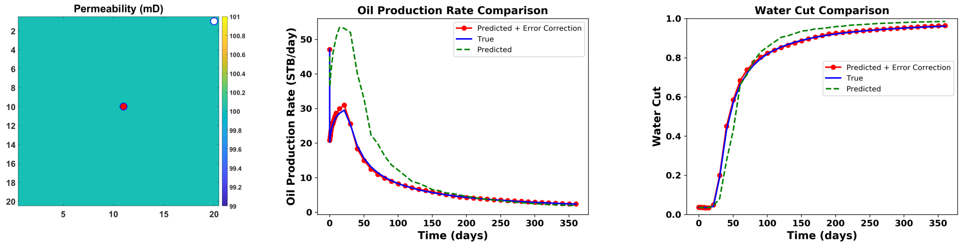
	RF Regression	**Train Accuracy**	**Test Accuracy**
Pressure	$N_{fmax} = 2$, $N_l = 2$	99.55	97.55
Saturation	$N_{fmax} = 2$, $N_l = 2$	98.71	90.79

Figures 27 and 28 show two of the test cases for this example by comparing the oil production rates and water cut. It still uses the same hyperparameters for the NN model as shown in Table 4. Figure 29 shows the errors in prediction of oil production rates and water cut for all these cases. The se results show a good agreement in the solutions thus showing the validity of error correction model for the cases when the global basis is not very accurate for the entire parameter domain but still can capture the overall trend of the solutions. The average accuracy for all the test cases can be found in Table 6. However, these observations may not hold true when the quality of global basis is compromised enough to explain the appropriate physics of flow.



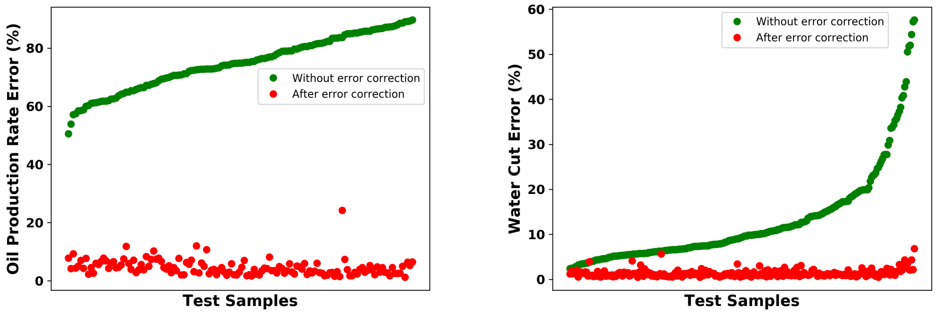
(**a**) (**b**) (**c**)

Figure 27. (**a**) Test case with producer well location at (3,16). (**b**) aomparison of oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone using 50 samples (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).



(**a**) (**b**) (**c**)

Figure 28. (**a**) Test case with producer well location at (11,10). (**b**) Comparison of oil production rate and (**c**) comparison of water cut, predicted the using global PMOR method alone using 50 samples (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).



(**a**) (**b**)

Figure 29. Error in the prediction of (**a**) the oil production rate and (**b**) water cut for all the test cases before and after the error correction of the solutions.

Table 6. Homogeneous reservoir case 2: Average accuracy of the oil production rate and water cut for all test samples.

	Average Accuracy (%)
Oil production rate	95.2
Water cut	98.3

8.2. Heterogeneous Reservoir Model—Single Well Case

The methodology shows promising results for a small homogeneous reservoir model. Now, we apply the same methodology to a heterogeneous reservoir model which is a section of layer 50 of the SPE10 benchmark model. Again, this is a two-phase flow (oil-water) reservoir model with one injector well and one producer well as shown in Figure 30. The reservoir model is discretized with a Cartesian grid of size 20 ft × 20 ft × 50 ft, and it contains 2500 (50 × 50) active cells. We neglect the capillary and gravity effects. The initial reservoir pressure is 4200 psi and the initial water saturation is considered 0. The injector and producer are BHP controlled at a constant pressures of 7000 psi and 2500 psi respectively. For the current scope of work, we believe, this model with about 5 orders of range in permeability and an injector and a producer, should be a good case with reasonable complexity to demonstrate the validity of proposed global PMOR strategy. Again we only consider that the producer is the parameter of interest which means the injector location is fixed and the producer changes location in the reservoir. The simulation is run for a duration of 3 years.

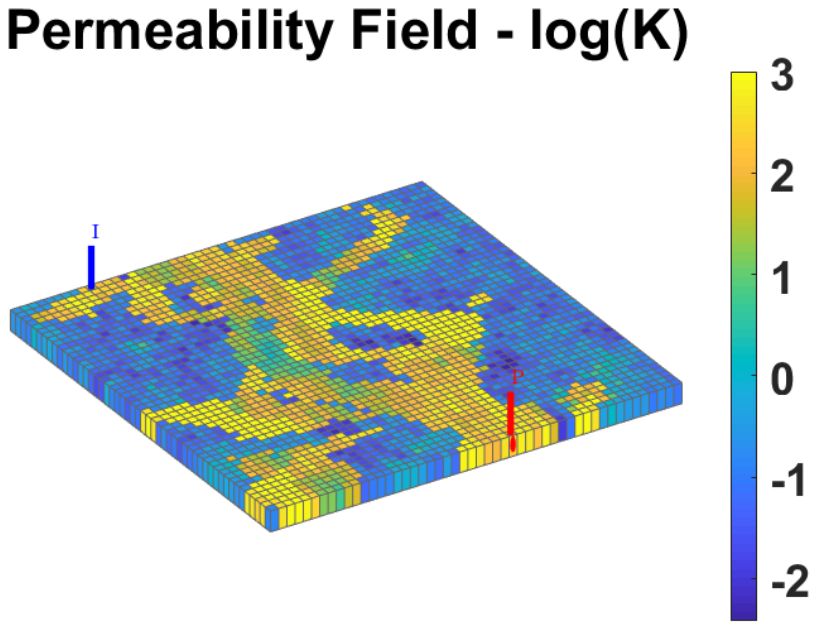


Figure 30. Heterogeneous log permeability field (section of SPE10 model layer 50) with one producer and one injector.

For the training of the global PMOR model, we consider the producer well locations only on the high permeability grid blocks majority of which lie on the channel connecting the injector and producer. We use all the grid blocks with greater than 10 mD as candidate well locations as shown in Figure 31. This is employed to observe the water cut for most of the well locations. Thus here, this global PMOR strategy is employed on a subset of parameter domain. The se are around 1000 sample locations out of which we use 200 randomly sampled locations for training the ML-based global PMOR model and 100 samples chosen for training the error correction model. We note here that for such highly heterogeneous model like this, even using the ROB computed at one well location produces very inaccurate result if used in the neighboring grid block. So with the current workflow we test all the possible well locations along the channel as our test cases to identify its performance.

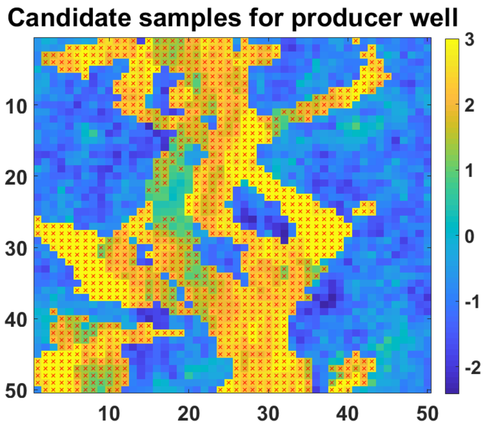


Figure 31. Samples considered as potential well locations shown by red crosses. Injector location is fixed at location (10,1) (not shown here) and hence not sampled.

We show the randomly sampled data set for training the global PMOR ML model in Figure 32a. Figure 32b shows the cumulative energy of singular values for the given sample set. For the first example, we use 99% energy criteria for pressure basis dimension and 90% energy criteria for saturation basis dimension. Later in the section, we analyze the accuracy of the method with changing energy criteria for these states. We observe the steep cumulative energy curve for pressure and a more concave curve for saturation. This is expected since all the well locations are just considered on the high permeability area of the reservoir. The observations about energy curves follows the same explanation as that of the homogeneous case if we consider well locations throughout the spatial domain of reservoir.

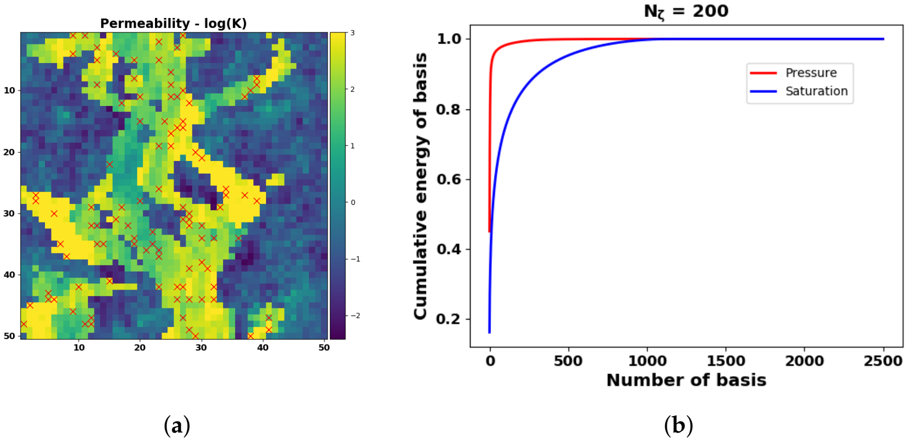


Figure 32. (**a**) Training samples randomly chosen for basis coefficient prediction. (**b**) Cumulative energy of eigenvalues for pressure (red) and saturation basis (blue).

Since we consider the heterogeneous case here, an additional feature corresponding to the well block permeability is added which was not included for the homogeneous model before. We first consider a test case, as shown in Figure 33 with producer well location at (28,50). Figures 34 and 35 show pressure and saturation solutions obtained by predicted POD basis coefficients with the true solutions for two of the test cases at different times respectively. We also show the error in pressure solution in Figure 36 to verify the same observations as in the homogeneous case where the maximum error is obtained at producer well locations. By observing the saturation solutions, we can see that there are small saturation errors near the waterfront. The time at which all the solutions are reported are chosen to be somewhere in the middle and at the end of simulation run time. As can be observed, the error is very small throughout the spatial domain and maximum at the well location which is the same observation as before.

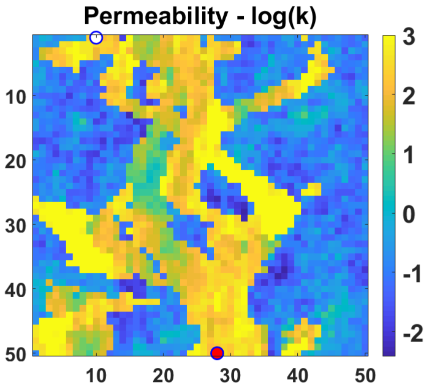
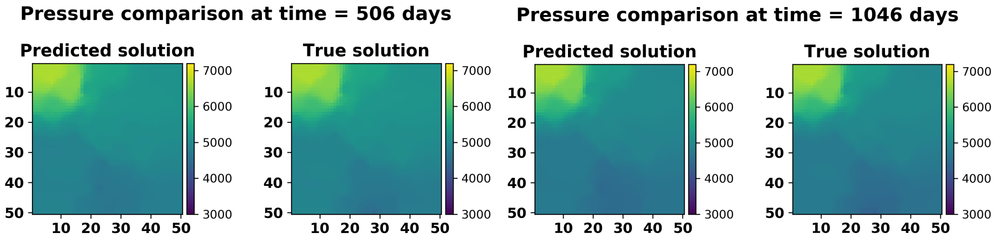
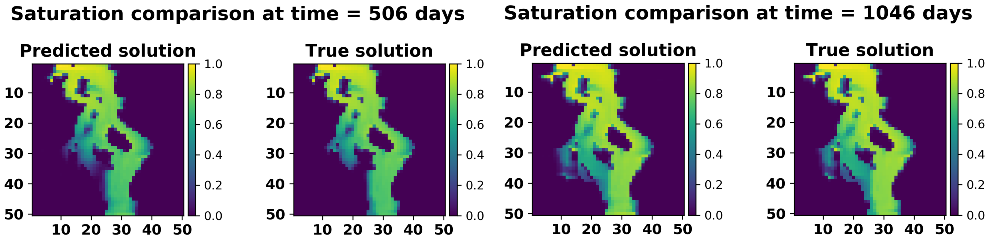


Figure 33. Test case with producer well at grid block (28,50).



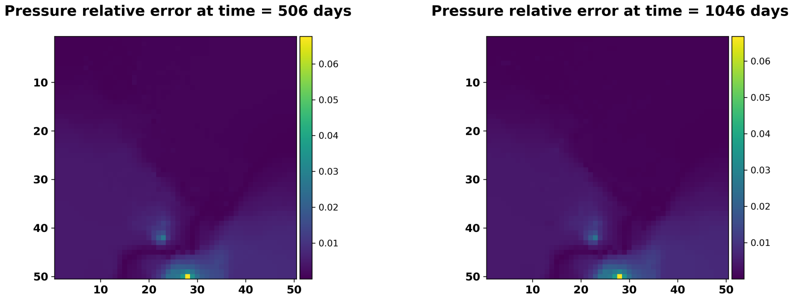
(**a**) (**b**)

Figure 34. Pressure solution comparison at (**a**) Time = 506 days and (**b**) Time = 1046 days.



(**a**) (**b**)

Figure 35. Saturation solution comparison at (**a**) Time = 506 days and (**b**) Time = 1046 days.



(**a**) (**b**)

Figure 36. Relative error in pressure at time = (**a**) 506 days (**b**) 1046 days.

Here, we also show the comparison of basis coefficients for pressure and saturation as predicted by the ML algorithm at the same time steps to determine the validity of ML models (Figures 37 and 38). We can see that the ML model produces a very small error in prediction of the coefficients for some of the pressure and saturation basis. Hence to check if the discrepancy in the predicted solution is due to the ML model or the quality of basis, we also compare the plots of relative error in ML predicted solution and the orthogonal projection error computed using Equation (43). This comparison for pressure in Figure 39 shows that the orthogonal projection error is maximum at the well location. Hence majority of the error in the output quantity of interest that is dependent on well block states is expected due to the quality of basis. Thus, it is in line with our discussion for homogeneous case. As we see that it is challenging to find a single global basis that is representative of the controllability properties at all the well locations, we use the error correction formulation in the second step to adjust for the error in state solutions at the grid block. Figures 40–42 show the test cases for different well locations comparing the oil production rates and watercut. We can see that the corrected solutions for some cases like test case 2 may not be very accurate but still shows much improvement after error correction. Similarly, for test case 3, where we see a lot of irregularities in the solutions obtained just by coefficient prediction, show significant improvement after error correction.

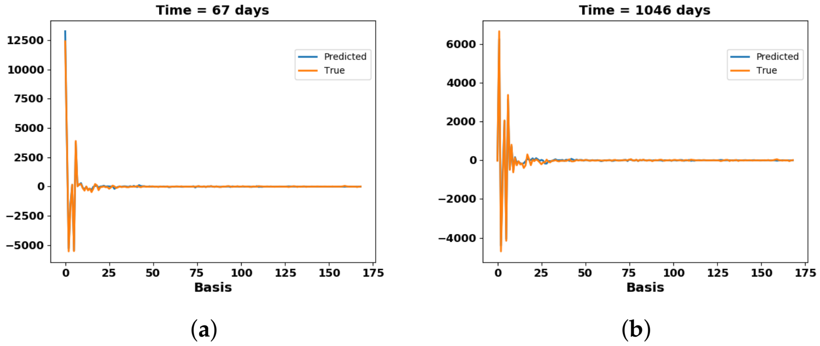


Figure 37. True and ML predicted pressure basis coefficient comparison at time = (**a**) 506 days and (**b**) 1046 days.

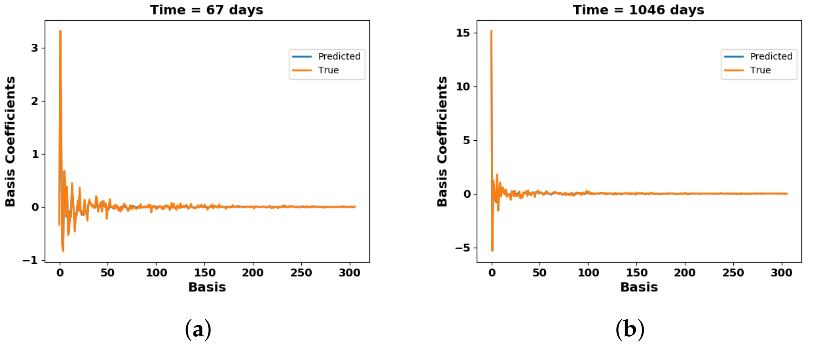


Figure 38. True and ML predicted pressure basis coefficient comparison at time = (**a**) 506 days and (**b**) 1046 days.

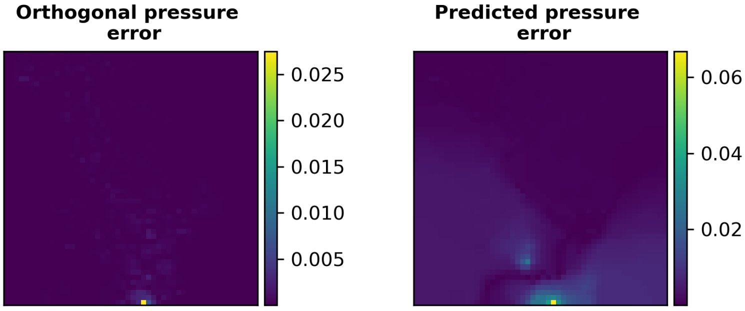
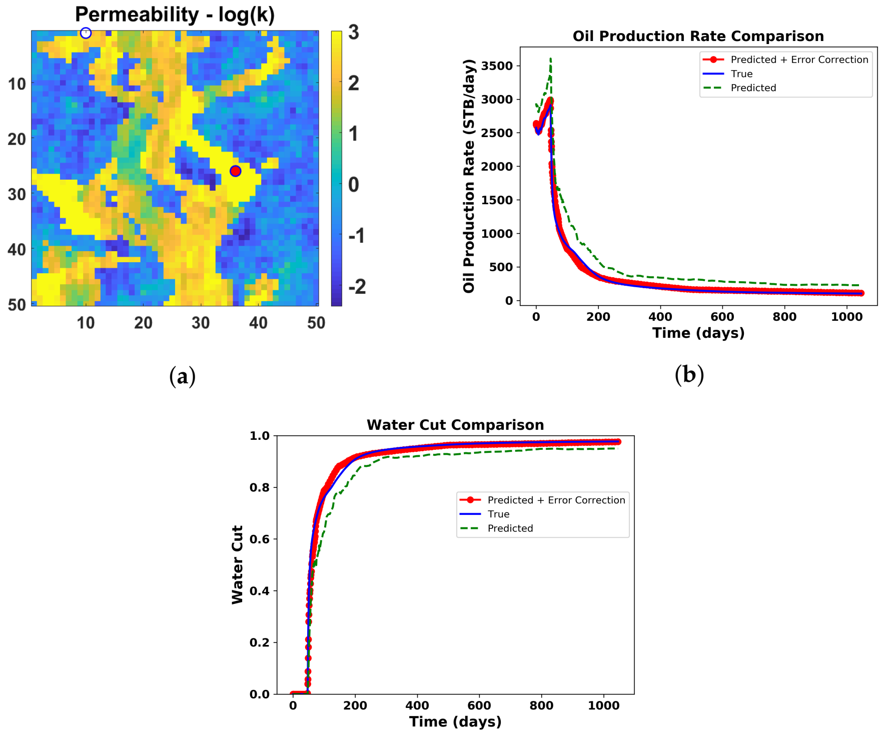


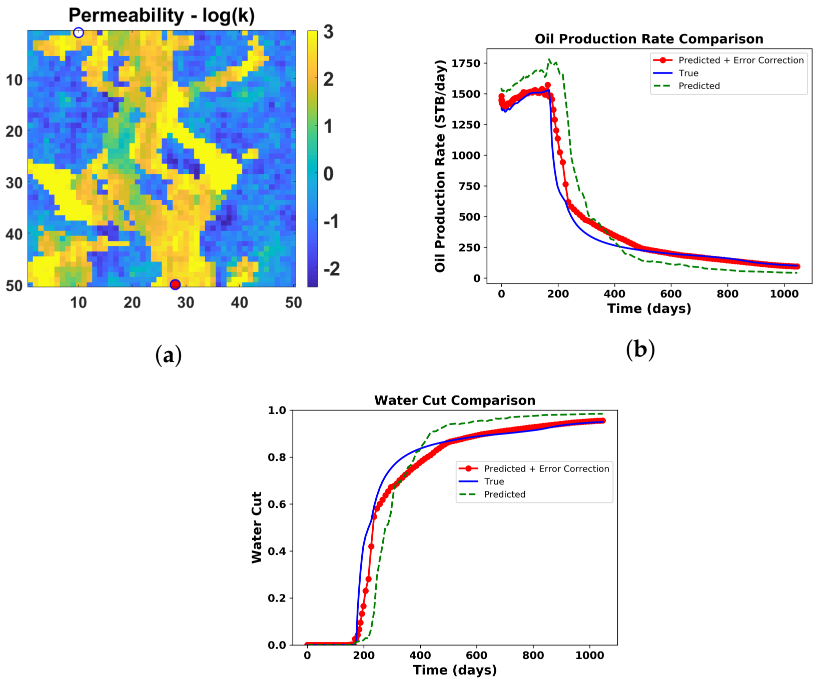
Figure 39. Comparison of the relative error in pressure due to orthogonal projection (**left**) and ML-predicted solution (**right**).



(**a**) (**b**)

(**c**)

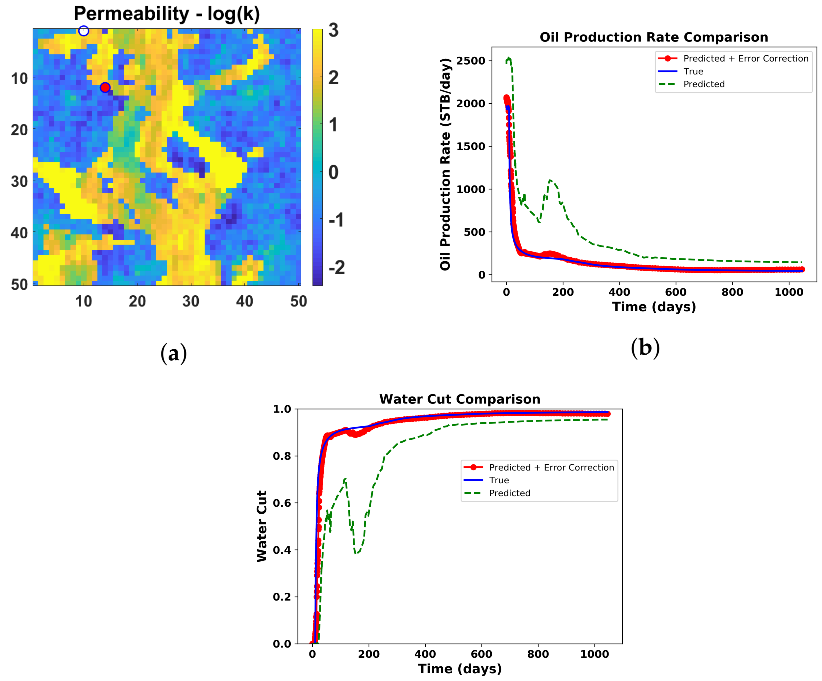
Figure 40. (**a**) Test case 1 with producer well location at (36,26). (**b**) Comparison of oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).



(**a**) (**b**)

(**c**)

Figure 41. (**a**) Test case 2 with the producer well location at (28,50). (**b**) Comparison of the oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).

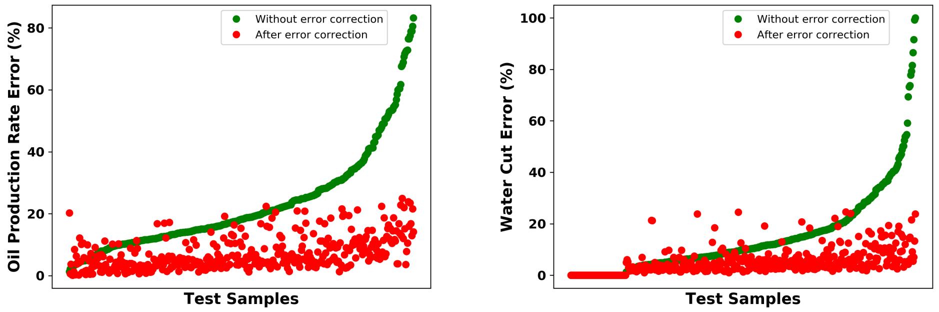


(**a**) (**b**)

(**c**)

Figure 42. (**a**) Test case 3 with producer well location at (14,12). (**b**) Comparison of oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).

For a better intuition about the results for this case study, we show a plot of accuracy in prediction for oil production rate and water cut for all the test cases (Figure 43). The samples are arranged in increasing order of errors obtained just from predicted basis coefficients and for each sample we show the error after employing the error correction model. It can be observed that after error correction, the accuracy of predictions in oil production rate and water cut increase significantly. There are some test cases that produces higher error after correction which were found to be the cases where no water cut was observed and very low oil production rates. A better sampling strategy can solve this issue. Table 7 shows the optimal hyperparameters of the NN model obtained through grid search. The average accuracy for all test cases after error correction is shown in Table 8.



(**a**) (**b**)

Figure 43. Error in the prediction of (**a**) the oil production rate (**b**) water cut for all the test cases before and after the error correction of the solutions.

Table 7. Error NN model hyperparameters found through grid search for the heterogeneous model with higher basis dimensions in the first step.

Number of Modes	L2 Regularization Parameter
10	0.4

Table 8. Heterogeneous reservoir case 1: average accuracy of oil production rate and water cut for all test samples.

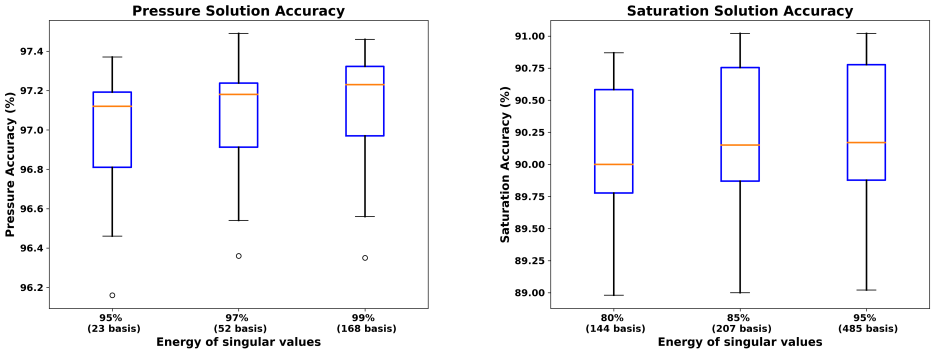
	Average Accuracy (%)
Oil production rate	92.6
Water cut	95.04

Table 9 shows the time comparison in seconds for these cases between fine-scale simulations run in Matlab and the proposed PMOR model with error correction, both run on a local 8-core machine. We see the speed-ups of about 50× for these test cases. This does not include time required to train the PMOR models.

Table 9. Time (seconds) comparison for the two test cases between fine-scale simulation and reduced-order model with error correction.

	Fine-Scale Simulation	PMOR + Error Correction
Test Case 1	65 s	1.3 s
Test Case 2	52 s	1 s

It is also important to decide on the energy criteria used for choosing the basis dimensions, since it is not trivial to make this decision a priori. As discussed before, there is a trade-off in accuracy and computational time between the decision to select the number of training samples and the basis dimensions. To get another perspective, we also show a box plot of the pressure and saturation solution accuracies as predicted from the basis coefficients for a range of energy criteria (Figure 44). Each box is constructed with 10 different training sets. As can be expected, with increasing number of basis, we see an increase in the accuracies of solutions. However, we also see a drastic reduction in the number of basis with decreasing percentage of singular values selected, especially for pressure. The accuracy shown here is the that when compared at all the grid blocks, which is not very different in magnitude between different basis dimensions. However, it was observed that the accuracy of solutions at the well grid block increases significantly with increasing basis dimension.



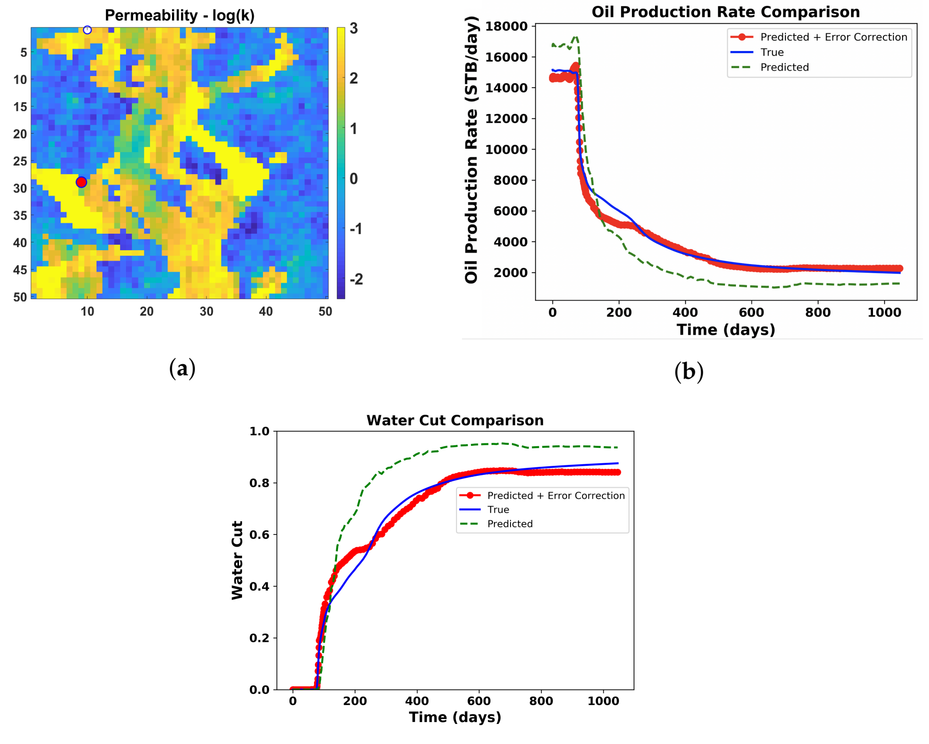
(**a**) (**b**)

Figure 44. Box plot of the accuracies using the predicted POD coefficients of (**a**) pressure for 95%, 97% and 99% energy of pressure singular values and (**b**) saturation for 80%, 85% and 95% energy of saturation singular values.

The case shown above corresponds to the pressure and saturation basis dimensions selected using energy criteria of 99% and 90%, respectively. Thus, we now evaluate, if the error correction model can also produce reasonable results if lower energy basis dimensions criteria is chosen. For this reason we consider another case with pressure and saturation basis dimension selected using energy criteria of 95% and 80% respectively (Note that these energy are chosen such that they capture most of the high energy singular values as shown in Figure 32b. If we choose even lower, the predicted basis coefficients are not expected to capture the physics of states behavior adequately and hence lead to inaccurate solutions). Figures 45 and 46 show two test cases comparing the oil production rate and water cut before and after error correction. In general, the solutions are a good match after error correction. The optimal NN error model hyperparameters for this model are shown in Table 10. The error in prediction, for all the test samples, before and after error correction is shown in Figure 47a,b. It can be seen that the error from predicted coefficients now increase at a higher rate which is expected due to much lower basis dimensions. However, after error correction, most of the test cases show much better accuracies. The average accuracy for oil production rate and water cut is shown in Table 11. It is lower than that predicted in the case of higher basis dimension (Table 8), but still not significantly less. Thus, it is up to one to choose appropriate basis dimensions such that significant computational speed-ups are obtained at the cost of losing accuracy.

Table 10. Error NN model hyperparameters found through grid search for the heterogeneous model with lower basis dimensions in the first step.

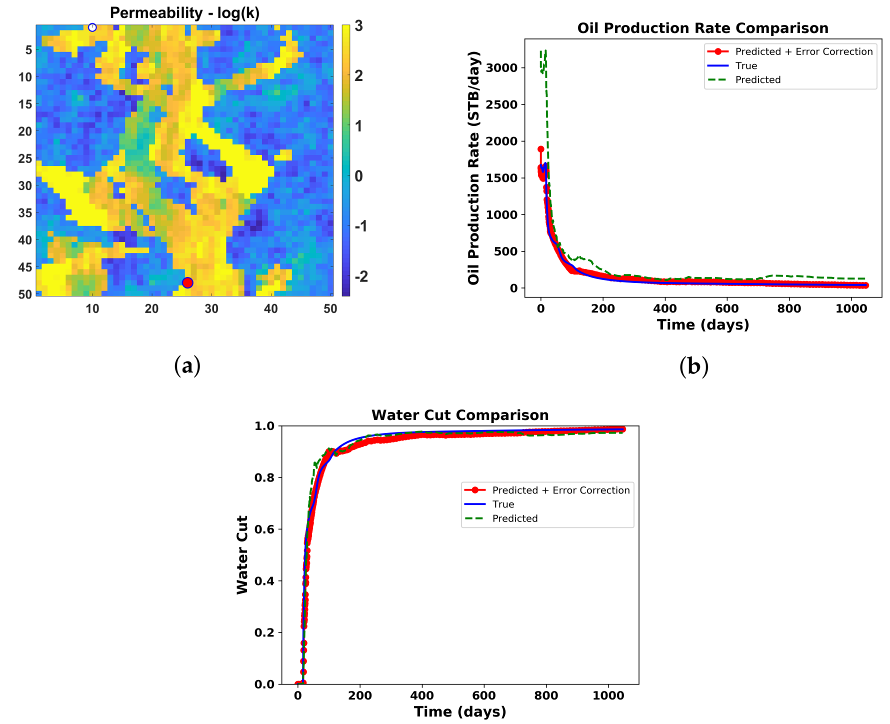
Number of nodes	L2 Regularization Parameter
10	0.2



(**a**) (**b**)

(**c**)

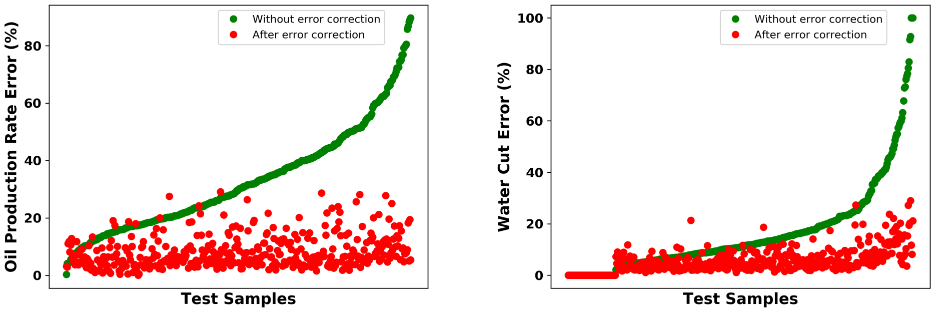
Figure 45. (**a**) Test case 1 with the producer well location at (9,29). (**b**) Comparison of the oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of error correction model (red circled line) with the true solution (blue line).



(**a**) (**b**)

(**c**)

Figure 46. (**a**) Test case 2 with the producer well location at (26,48). (**b**) Comparison of oil production rate and (**c**) comparison of water cut, predicted using the global PMOR method alone (dotted green line) and after the implementation of the error correction model (red circled line) with the true solution (blue line).



(**a**) (**b**)

Figure 47. Error in the prediction of (**a**) the oil production rate and (**b**) water cut for all the test cases before and after the error correction of the solutions.

Table 11. Heterogeneous reservoir case 2: average accuracy of the oil production rate and water cut for all test samples.

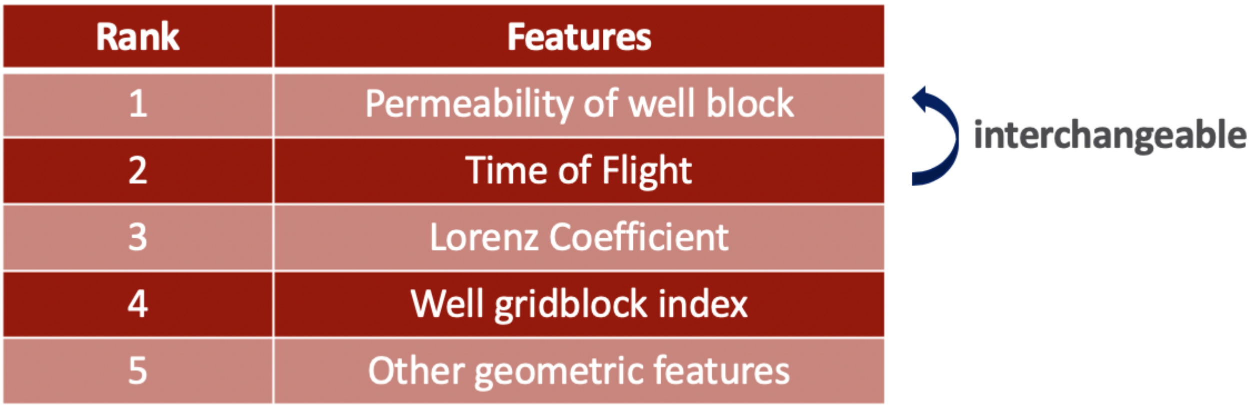
	Average Accuracy (%)
Oil production rate	91.74
Water cut	94.6

Table 12 shows the time comparison in seconds for these cases between the fine-scale simulations run and the proposed PMOR model with error correction, both run on a local 8 core machine. We see speed-ups of about 100× for these test cases, since the model outputs have lower dimensions and hence are less complex.

Table 12. Time (seconds) comparison for the two test cases (lower-dimensional basis) between fine-scale simulation and reduced-order model with error correction.

	Fine Scale Simulation	PMOR + Error Correction
Test Case 1	56 s	0.7 s
Test Case 2	46 s	0.5 s

Figure 48 shows the rank of features from the Random Forest model to compute coefficients, which clearly indicates the importance of physics-based features. We ran several cases with different combination of training samples and the order of this rank remained the same, except the permeability of the well block, with the time of flight being interchangeable but top-ranked.



Rank	Features
1	Permeability of well block
2	Time of Flight
3	Lorenz Coefficient
4	Well gridblock index
5	Other geometric features

interchangeable

Figure 48. Feature Rank as calculated by the Random Forest model.

8.3. Heterogeneous Reservoir Models—Multiple Wells Case

The results for a single producer and injector show the validity of the proposed methodology, but for practical implementation we need to see if this method shows such promising results. It is mainly the future work that needs to be addressed, but we show a simple case study as a beginning for future directions. We know from the MOR literature that MOR techniques, when used within optimization workflows, require many simulation runs corresponding to new parameter sets during each iteration. The training set of parameters for MOR development is designed such that the parameters are not drastically different from those expected to be encountered in the test set. For well locations, as we see from the above results, it is very difficult to determine how different is one location to the other, especially for heterogeneous reservoirs such as the section of the SPE10 model considered. Additionally, the parameter domains for the well locations increase significantly for large reservoir models, in that each grid block in the reservoir is a potential well location. This becomes even more challenging, as we have more wells where we can have different combinations of well locations across all the grid blocks. In these scenarios, it just becomes impractical to devise PMOR strategies that can satisfy all possible combinations of well locations in the reservoir model. Thus, training a PMOR model must be tied to the optimization strategy used and training samples must be designed accordingly. For example, since well location optimization is a very challenging problem given numerous possible well configurations, researchers resort to quick evaluations or diagnostics of the reservoir to rule out many areas in the reservoir as potential well locations [24,40]. Then, given this information, complex optimization routines using simulation runs can be sought for a systematic decision-making process.

The method currently proposed is suitable for any well location in the reservoir, and hence further adds to the training effort by collecting more samples. However, in the future, when this workflow is applied with optimization routines, the training strategy should be designed keeping the optimization strategy in mind. As a preliminary idea, we show a simple case for the same model as considered for single well case, but now we consider two producers and one injector. We keep the injector location fixed and the producers can change locations. With 2500 grid blocks and the injector location fixed, we can have 2498 × 2498 = 6,240,004 producer combinations. If both the producers have the same well properties and are produced with the same BHP schedule—i.e., both the well are equivalent—we have 6,240,004/2 = 3,120,002 different well configurations. Thus, the optimization strategies are sought such that these possible combinations are reduced a priori by setting some constraints. We only consider a simplistic case here as a beginning to validate the method for multiple wells. Figure 49 shows the reservoir model with an injector and two regions (shown in red) where producer 1 and producer 2 locations are randomly sampled. We choose 20 samples in each region and hence we can have 400 combinations of producer 1 and producer 2 pairs. The BHP schedules for both producers are kept the same and have identical well properties. The simulation is run for a duration of 500 days. The ML model is trained to predict the POD basis coefficients for pressure and saturation using 50 sample points randomly chosen out of 400. One hundred training samples are randomly chosen for an error correction ANN model. Note that the change here, as compared to single-well formulation, lies in the input features defined for ML models. Now the inputs include geometric and flow diagnostics-based features for both the wells in the ML model to predict the basis coefficients, and the error correction model has the same features representing both producer wells along with the reduced-order states and pore volumes injected.

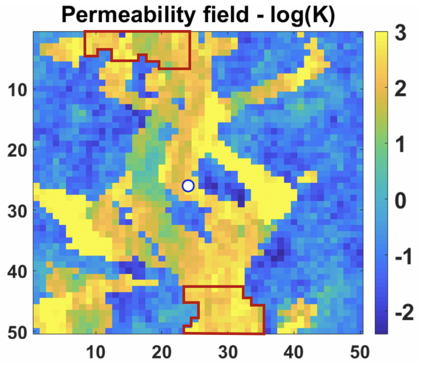
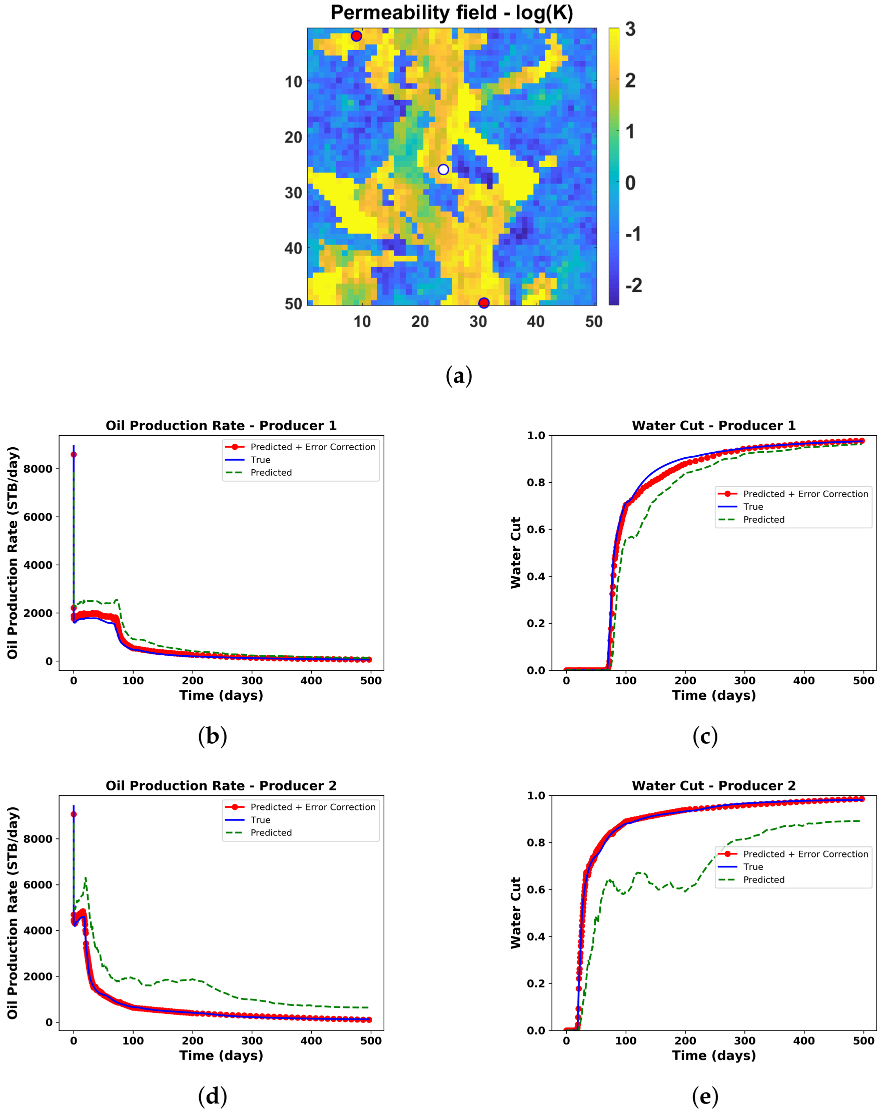


Figure 49. Reservoir model with an injector at location (24,26) and red regions showing the regions where the producer 1 (top of the reservoir) and producer 2 (bottom of the reservoir) locations are sampled.

Figures 50 and 51 show two of the test cases with different producer 1 and 2 pairs than were used in training. For test case 1, the results show a good accuracy after error correction, which shows erratic behavior for producer 2 using just the predicted basis coefficients. Similarly, test case 2 is an example where even without error correction the results are a good match to the true solutions, which improve further after error correction. Figure 52 shows the test accuracies for all the 250 test samples of oil production rates and water cut for both the producers. As we can see, the accuracy after error correction improves significantly for many test cases.



(**a**)

(**b**) (**c**)

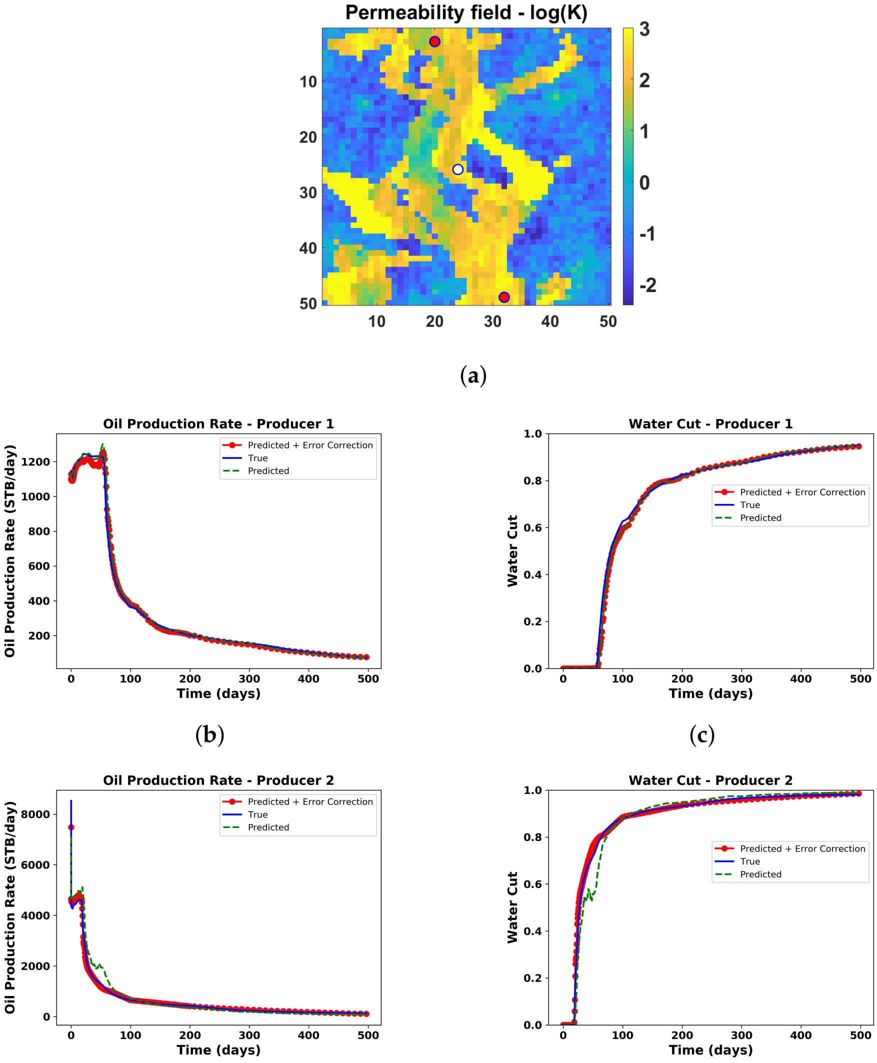
(**d**) (**e**)

Figure 50. (**a**) Test case 1 with producer 1 at (9,2) and producer 2 at (31,50). (**b**) Comparison of oil production rate for producer 1, (**c**) comparison of water cut for producer 1, (**d**) comparison of oil production rate for producer 2, and (**e**) comparison of water cut for producer 2, predicted using the global PMOR method alone (dotted green line) and after the implementation of the error correction model (red circled line) with the true solution (blue line).

Table 13 shows the comparison of time between fine-scale simulation and the proposed PMOR model with error correction. The speed-up obtained for these cases was about 100×. Note that these times are just for the test runs. It does not include the training time, which must be the area of focus for reduction in the future.

Table 13. Time (seconds) comparison for the two test cases between fine-scale simulation and reduced-order model with error correction.

	Fine Scale Simulation	**PMOR + Error Correction**
Test Case 1	102 s	1 s
Test Case 2	90 s	0.8 s



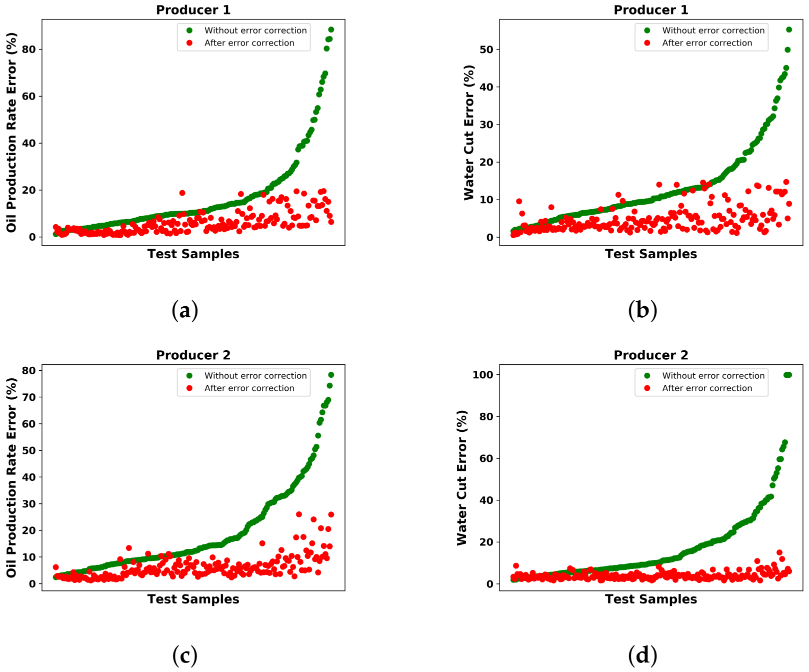
(**a**)

(**b**) (**c**)

(**d**) (**e**)

Figure 51. (**a**) Test case 2 with producer 1 at (20,3) and producer 2 at (32,49). (**b**) Comparison of oil production rate for producer 1, (**c**) comparison of water cut for producer 1, (**d**) comparison of oil production rate for producer 2, (**e**) comparison of water cut for producer 2, predicted using the global PMOR method alone (dotted green line) and after the implementation of the error correction model (red circled line) with the true solution (blue line).

From these results, we validate the applicability of the proposed workflow as a starting point for the case of multiple wells. We believe that the explanation on its used with the optimization procedure described above should be taken into consideration for practical implementation. As we move to the cases with many wells, the feature space becomes high-dimensional, that might lead to slower training procedures. A possible way to avoid this would be the observation that the time of flight, well block permeability, Lorenz coefficients, and indices of the well grid blocks, which are the most important features representing well locations, be used for POD basis coefficient prediction; then, during the error correction step, we may use all the geometric features along with physics-based features. This can reduce the dimensionality of the feature space and hence lead to a faster training procedure.



(**a**) (**b**)

(**c**) (**d**)

Figure 52. Error in the prediction of (**a**) the oil production rate and (**b**) water cut for producer 1 and (**c**) oil production rate and (**d**) water cut for producer 2 for all the test cases before and after the error correction of the solutions.

8.4. Advantages and Limitations

The two-step method described above requires many fine-scale simulations to train the ML models but can easily be used with high-performance computing facilities. With the advent of parallel computing resources and cloud architecture, the training procedure can be extremely fast. One may argue for running just fine-scale simulations, but it should also be considered that the proposed technique has been designed for well location optimization problems where each optimization iteration may be dependent on the solution of the previous iteration depending on the optimization strategy used, and hence cannot make full use of parallel resources. Keeping the expensive training procedure in mind, one may we may also design better sampling methods that reduce the number of sampling points significantly. We also note here that the methodology described is used for any parameter in the parametric domain. That is, the PMOR method is strictly global in the sense that the test cases may be significantly different from the training cases, which is a likely problem to encounter in the new well locations, as a new well location can change the dynamics of the problem significantly. This is not usually the case with the PMOR methods developed for the well control optimization problem, where the training procedure is not designed such that it is drastically different from the test cases or they may fail or lead to unstable solutions.

The advantages of the proposed method are manyfold. One is that this PMOR strategy is completely non-intrusive in nature, as it just requires data (pressure and saturation states, in our case) from any simulator and does not require the evaluation of any non-linear functions, such as Jacobian and residuals. Second, the ML procedure described here is extremely fast in predicting the solutions at a given time in the simulation run. For the case of the heterogeneous reservoir above, a computational speed-up of around 50× (case 1 with large basis dimensions) to about 100× (case 2 with lower basis dimensions) was observed for the test examples when run on a local 8-core machine. Each forward ML prediction (coefficients and error correction) is very fast, and the proposed method also allows multiple time steps to be computed in parallel. Since each time step is entered as a feature in the ML model, the solution at the current time step is independent of the solution at the previous time step, as in the reservoir simulation. Thus, we do not rely on the time steps for solution convergence and running the time steps in series with the proposed method. This also helps us to avoid convergence issues that may occur due to a few bad predictions at some time steps. However, one limitation of this method is that POD may not be the best way to reduce the pressure solutions for changing well locations, and thus, if replaced with a better ROM technique, the same workflow can be further improved. Additionally, the proposed workflow does not necessarily obey mass conservation, and it should be a research focus in the future to handle this problem.

9. Conclusions

In this paper, we introduced a two-step predict and correct non-intrusive global parametric model order-reduction method for changing well locations based on Machine Learning techniques. This method requires just the pressure and saturation states at different time steps and a global basis is constructed from the representative well locations in the reservoir. In the first step, the Machine Learning model is trained to learn the coefficients of POD basis for the well locations, as defined by a set of geometric and flow diagnostics-based features. We also introduce time as a feature to account for the temporal behavior of the coefficients. The trained ML model is then used to predict the POD coefficients of the states for new well locations. The ML model used here is Random Forest, which is much faster to train compared to other algorithms such as ANN, especially for large dimensional outputs, and has the ability to define complex input–output relations. The ML model predicts the basis coefficients extremely fast, with speed-ups of about 50× (for high-dimensional basis)–100× (for low-dimensional basis). We recognize the problem of using POD for reducing the pressure state for changing well locations, since each new location is a new lowest pressure point, as can be justified by the controllability theory. However, with a global ROB the ML model could capture the overall solution trend with a bias at the well location. Due to a systematic discrepancy in the solution, in that it captures the solution behavior, we considered fixing this discrepancy through error correction model. Thus, in the second step of the workflow we construct an ML-based error correction model that captures the state disparity given reduced model solutions as a feature. ANN was used in this case, which is faster to train in the second step, since it is a single-output model and was proved to reduced the error of prediction drastically for most test cases. The proposed workflow has a limitation, since it requires many training samples to define the global parameter space. This should be looked into in future research using efficient sampling techniques in order to reduce the number of samples required significantly and use local parameter regions to develop multiple local models in a similar way to that proposed. We also note that the proposed method must be tied with the optimization strategy used, which would help us design a better training strategy, as it is impractical to expect the model to be accurate for all possible well configurations in the reservoir.

Author Contributions: Conceptualization, H.Z. and E.G.; methodology, H.Z. and E.G.; software, H.Z.; validation, H.Z.; formal analysis, H.Z.; investigation, H.Z.; resources, E.G.; data curation, E.G.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z. and E.G.; visualization, H.Z.; supervision, E.G.; project administration, E.G.; funding acquisition, E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by Energi Simulation through the Energi Simulation Research Chair at Texas A&M.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ROM	Reduced-Order Model
ROB	Reduced-Order Basis
φ	Porosity of the reservoir
ρ_l	Density of the fluid per unit volume
$\widetilde{m_l}$	Mass flow rate
q_l	Volumetric flow rate
$\mathbf{v}_l$	Darcy velocity of liquid
μ_l	Viscosity of fluids
k_{rl}	Relative permeability of fluids
K	absolute permeability
S_l	Saturation of phase l
p_l	Pressure of phase l
$\mathbf{x}$	Pressure and saturation state vector
N_d	Degrees of freedom
ζ	well location parameter
η	error tolerance
N_ζ	Number of well location configurations
$\mathbf{T}$	Transmissibility Matrix
$\mathbf{G}$	Gravity Vector
$\mathbf{Q}$	Volumetric source/sink
$\mathbf{R}$	Residual Vector
$\mathbf{D}$	Accumulation Matrix
$\mathbf{J}$	Jacobian Matrix
N_t	Number of simulation time steps
$\boldsymbol{\Phi}$	Reduced-Order Basis
ϕ	Reduced basis function
t	Time
r	Reduced dimensionality of a vector or matrix
ϵ_{ROM}	Reduced-Order Model error
$\mathbf{X}$	Snapshot matrix of pressure and saturation
$\mathbf{U}$	Left singular matrix
$\mathbf{V}$	Right singular matrix
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
$\mathbf{c}_{\mathbf{x}}(t, \zeta)$	Matrix of basis coefficients at different time steps t and parameters ζ
PVI_r	Pore volumes injected
$\mathbf{x}_{wb}$	State solutions at well blocks
$\tilde{\mathbf{x}}_{wb}$	Reduced model solution from step 1
$\Delta\mathbf{x}_{wb}$	Error in the well block state solution

References

1. Jansen, J.D.; Brouwer, R.; Douma, S.G. Closed Loop Reservoir Management. In Proceedings of the SPE Reservoir Simulation Symposium, Galveston, TX, USA, 2–4 February 2009. [CrossRef]
2. Shirangi, M.G.; Durlofsky, L.J. Closed-Loop Field Development Under Uncertainty by Use of Optimization with Sample Validation. *SPE J.* **2015**, *20*, 908–922. [CrossRef]
3. Antoulas, A. Approximation of Large-scale Dynamical Systems. In *SIAM Advances in Design and Control*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005.
4. Hinze, M.; Volkwein, S. *Dimension Reduction of Large-Scale Systems*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 45, pp. 261–306. [CrossRef]
5. King, M.J.; Macdonald, D.G.; Todd, S.P.; Leung, H. SPE 50643-Application of Novel Upscaling Approaches to the Magnus and Andrew Reservoirs. In Proceedings of the European Petroleum Conference, The Hague, The Netherlands, 20–22 October 1998.
6. March, A.; Willcox, K. Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives. *AIAA J.* **2012**, *50*. [CrossRef]
7. Kennedy, M.C.; Hagan, A.O. Bayesian calibration of computer models. *J. R. Stat. Soc.* **2001**, *63*, 425–464. [CrossRef]
8. Knill, D.L.; Giunta, A.A.; Baker, C.A.; Grossman, B.; Mason, W.H.; Haftka, R.T.; Watsonft, L.T. Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design. *J. Aircraft* **1999**, *36*. [CrossRef]

9. Doren, J.F.M.V.; Markovinovic, R. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comp. Geosci.* **2006**, *10*, 137–158. [CrossRef]
10. Cardoso, M.A.; Durlofsky, L.J.; Sarma, P. Development and application of reduced-order modeling procedures for subsurface flow simulation. *Int. J. Num. Method. Eng.* **2008**, 1322–1350. [CrossRef]
11. Cardoso, M.A.; Durlofsky, L.J. Use of Reduced-Order Modeling Procedures for Production Optimization. In Proceedings of the SPE Reservoir Simulation Symposium, Woodlands, TX, USA, 2–4 February 2009; pp. 2–4. [CrossRef]
12. He, J.; Sætrom, J.; Durlofsky, L.J. Enhanced linearized reduced-order models for subsurface flow simulation. *J. Comp. Phys.* **2011**, *230*, 8313–8341. [CrossRef]
13. Trehan, S.; Durlofsky, L.J. Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to PDE-constrained optimization. *J. Comput. Phys.* **2016**, *326*, 446–473. [CrossRef]
14. Chaturantabut, S.; Sorensen, D. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764. [CrossRef]
15. Gildin, E.; Ghasemi, M.; Protasov, A.; Efendiev, Y. SPE 163618 Nonlinear Complexity Reduction for Fast Simulation of Flow in Heterogenous Porous Media. In Proceedings of the Reservoir Simulation Symposium, Woodlands, TX, USA, 18–20 February 2013.
16. Ghasemi, M.; Yang, Y.; Gildin, E.; Efendiev, Y.; Calo, V. Fast Multiscale Reservoir Simulations using POD-DEIM Model Reduction. In Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, USA, 25 February 2015. [CrossRef]
17. Sorek, N.; Zalavadia, H.; Gildin, E. SPE-182652-MS Model Order Reduction and Control Polynomial Approximation for Well-Control Production Optimization. In Proceedings of the SPE Reservoir Simulation Conference, Montgomery, TX, USA, 20–22 February 2017.
18. Tan, X.; Gildin, E.; Trehan, S.; Yang, Y.; Hoda, N. Trajectory-Based DEIM TDEIM Model Reduction Applied to Reservoir Simulation. In Proceedings of the SPE Reservoir Simulation Conference, Montgomery, TX, USA, 20–22 February 2017. [CrossRef]
19. Ghasemi, M.; Ibrahim, A.; Gildin, E. Reduced Order Modeling In Reservoir Simulation Using the Bilinear Approximation Techniques. In Proceedings of the SPE Latin America and Caribbean Petroleum Engineering Conference, Maracaibo, Venezuela, 21–23 May 2014. [CrossRef]
20. Ghasemi, M.; Gildin, E. Model Order Reduction in Porous Media Flow Simulation using Quadratic Bilinear Formulation. *Comp. Geosci.* **2016**, *20*, 723–735. [CrossRef]
21. Jiang, R.; Durlofsky, L.J. Implementation and detailed assessment of a GNAT reduced-order model for subsurface flow simulation. *J. Comp. Phys.* **2019**, *379*, 192–213. [CrossRef]
22. Zalavadia, H.; Sankaran, S.; Kara, M.; Sun, W.; Gildin, E. *A Hybrid Modeling Approach to Production Control Optimization Using Dynamic Mode Decomposition*; Society of Petroleum Engineers: Richardson, TX, USA, 2019. [CrossRef]
23. Centilmen, A.; Ertekin, T.; Grader, A.S. SPE 56433 Applications of Neural Networks in Multiwell Field Development. In Proceedings of the SPE Annual Technical Conference & Exhibition, Houston, TX, USA, 3–6 October 1999.
24. Taware, S.V.; Park, H.Y.; Datta-Gupta, A.; Bhattacharya, S.; Tomar, A.; Kumar, M.; Rao, H. Well Placement Optimization in a Mature Carbonate Waterflood using Streamline-based Quality Maps. In Proceedings of the SPE Oil and Gas India Conference and Exhibition, Mumbai, India, 28–30 March 2012. [CrossRef]
25. Casenave, F.; Ern, A.; Lelièvre, T. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Adv. Computat. Math.* **2015**, *41*, 961–986. [CrossRef]
26. Barthelmann, V.; Novak, E.; Ritter, K. High dimensional polynomial interpolation on sparse grids. *Adv. Computat. Math.* **2000**, *12*, 273–288. [CrossRef]
27. Amsallem, D.; Farhat, C. An Online Method for Interpolating Linear Parametric Reduced-Order Models. *SIAM J. Sci. Comput.* **2011**, *33*, 2169–2198. [CrossRef]
28. Hesthaven, J.S.; Ubbiali, S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.* **2018**, *363*, 55–78. [CrossRef]
29. Swischuk, R.; Mainini, L.; Peherstorfer, B.; Willcox, K. Projection-based model reduction: Formulations for physics-based machine learning. *Comp. Fluid.* **2018**, 1–14. [CrossRef]
30. Zalavadia, H.; Gildin, E. Th A2 08 Parametric Model Order Reduction for Adaptive Basis Selection Using Machine Learning Techniques during Well Location Opt. In Proceedings of the 15th European Conference on the Mathematics of Oil Recovery (ECMOR XV), Amsterdam, The Netherlands, 29 August–1 September 2018.
31. Ertekin, T.; Abou-Kassem, J.H.; King, G.R. *Basic Applied Reservoir Simulation*; Society of Petroleum Engineers: Richardson, TX, USA, 2001.
32. Aziz, K.; Aziz, K.; Settari, A. Petroleum Reservoir Simulation. Applied Science Publishers 1979, 476p. *Adv. Chem. Eng. Sci.* **2016**, *6*, 4.
33. Chen, Z.; Huan, G.; Ma, Y. *Computational Methods for Multiphase Flows in Porous Media*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006; p. 523. [CrossRef]
34. Lie, K.A. *An Introduction to Reservoir Simulation Using MATLAB / GNU Octave User Guide for the Matlab Reservoir Simulation*; Cambridge University Press: Cambridge, UK, 2018.
35. Bao, A.; Gildin, E. Data-Driven Model Reduction Based on Sparsity-Promoting Methods for Multiphase Flow in Porous Media. In Proceedings of the SPE Latin America and Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, 18 May 2017. [CrossRef]

36. Breiman, L.E.O. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Datta-Gupta, A.; King, M.J. *Streamline Simulation: Theory and Practice*; Society of Petroleum Engineers: Richardson, TX, USA, 2007.
38. Datta-Gupta, A.; King, M.J. A semianalytic approach to tracer flow modeling in heterogeneous permeable media. *Adv. Water Res.* **1995**, *18*, 9–24. [CrossRef]
39. Shahvali, M.; Mallison, B.; Wei, K.; Gross, H.; Energy, C.; Company, T. An Alternative to Streamlines for Flow Diagnostics on Structured and Unstructured Grids. *SPE J.* **2012**, *17*, 768–778. [CrossRef]
40. Møyner, O.; Krogstad, S.; Lie, K.A. The Application of Flow Diagnostics for Reservoir Management. *SPE J.* **2015**, *20*, 306–323. [CrossRef]
41. Shook, G.M.; Mitchell, K.M. *A Robust Measure of Heterogeneity for Ranking Earth Models: The F PHI Curve and Dynamic Lorenz Coefficient*; Society of Petroleum Engineers: Richardson, TX, USA, 2009. [CrossRef]
42. Kohavi, R.; John, G.H. Wrappers for Feature Subset Selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
43. Trehan, S.; Durlofsky, L.; Carlberg, K. Error Estimation for Surrogate Models of Dynamical Systems Using Machine Learning. *Int. J. Num. Method. Eng.* **2016**, 1–31. [CrossRef]
44. Goodfellow, I.J.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.