APPLICATIONS OF MACHINE-LEARNING IN RESERVOIR CONNECTIVITY DETECTION AND OPTIMIZATION

A Dissertation

by

DEEPTHI SEN

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Akhil Datta-Gupta
Committee Members,	Michael J King
	Eduardo Gildin
	Bani Mallick
Head of Department,	Jeff Spath

May 2022

Major Subject: Petroleum Engineering

ABSTRACT

Reliable quantification of well connectivity is a crucial aspect in forming a good understanding of a reservoir, which in turn helps in formulating future development plans such as rate optimization and offset wells. This assumes an even greater importance when applied to high-capital projects such as CO2 EOR and polymer floods. Conventional methods for assessing well connectivity include tracer tests and numerical simulation-based techniques such as streamlines. However, these methods of connectivity detection tend to be either computation-intensive (i.e. numerical simulation) or resource-intensive (such as tracer tests).

This dissertation makes three major contributions related to machine-learning applications for connectivity detection and rate optimization. Firstly, I propose a novel approach for connectivity quantification and rate optimization during a waterflood under geologic uncertainty in reservoir properties such as permeability and porosity. A machine-learning (ML) based approach which is quick and scalable for rate optimization over multiple geologic realizations is proposed instead.

Secondly, a machine-learning framework is built on the statistical recurrent unit (SRU) model that interprets well-based injection/production data into inter-well connectivity without relying on a geologic model. Furthermore, a streamline-based validation procedure is also proposed which provides physics-based backing to the results obtained from data analytics.

Thirdly, this dissertation proposes a workflow that integrates unsupervised machine learning and streamline techniques to select representative geologic realizations based on their flow features. The workflow may be used to identify key wells for implementing optimized rate schedules, while taking into account the uncertainty in the geologic model.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Akhil Datta-Gupta, and my committee members, Dr. Michael King, Dr. Eduardo Gildin, Dr. Bani Mallick for their guidance and support throughout the course of this research. I would also like to thank Dr. Debjyoti Banerjee for serving as a substitute committee member in place of Dr. Mallick during my final examination. I would like to express my sincere gratitude to Dr. Hongquan Chen for his mentorship and contribution to my research.

Thanks also go to my closest friends – Dilly, Joshiba and Akshi, my colleagues and the department faculty and staff for making my grad school experience at Texas A&M University truly enjoyable.

Finally, thanks to my mother and father for their encouragement and to my husband Adi for his love and support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professors Akhil Datta-Gupta (advisor), Michael J. King and Eduardo Gildin of the Department of Petroleum Engineering and Professor Bani Mallick of the Department of Statistics.

The streamline-tracing software used in Chapter II and rate optimization algorithm used in Chapter IV were developed by Dr. Hongquan Chen.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by Texas A&M University Joint Industry Projects (JIP), Model Calibration and Efficient Reservoir Imaging (MCERI).

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION[1]

Reliable quantification of well connectivity is a crucial aspect to forming a good understanding of a reservoir, which in turn helps in formulating future development plans such as rate optimization and offset wells (Chen et al. 2020b). This assumes an even greater importance when applied to high-capital projects such as CO2 EOR and polymer floods.

Conventional methods for assessing well connectivity include tracer tests (Zhang et al. 2016, Suarsana and Badril 2011) and numerical simulation-based techniques such as streamlines (Datta-Gupta and King 2007). Streamlines explicitly reveal the flow paths in the reservoir and quantify the interaction between injectors and producers via well-pair flux allocations. Extensive literature has shown the capacity of streamlines over a variety of grid systems, including Cartesian grids (Pollock 1988), corner point grids (Cordes and Kinzelbach 1992; Jimenez et al. 2010), unstructured grids (Prevost et al.

---

[1] Part of this chapter is reprinted with permission from Sen, Deepthi, et al. "Data-Driven Rate Optimization Under Geologic Uncertainty." SPE Annual Technical Conference and Exhibition. OnePetro, 2020. Copyright 2020 Society of Petroleum Engineers. Further reproduction is prohibited without permission.

[1] Part of this chapter is reprinted with permission from Sen, Deepthi, et al. "Machine learning based rate optimization under geologic uncertainty." Journal of Petroleum Science and Engineering 207 (2021): 109116.

[1] Part of this chapter is reprinted with permission from Chen, Hongquan, et al. "Model-Free Assessment of Inter-Well Connectivity in CO2 WAG Projects Using Statistical Recurrent Unit Models." SPE Annual Technical Conference and Exhibition. OnePetro, 2021. Copyright 2021 Society of Petroleum Engineers. Further reproduction is prohibited without permission.

2002; Rasmussen 2010; Zhang et al. 2012), and dual porosity single permeability models (Chen et al. 2020c).

However, these conventional methods tend to be either computation-intensive (i.e. numerical simulation) or resource-intensive (such as tracer tests). In recent years, data-driven methods that derive valuable insights from commonly recorded field data have proved to be quite successful in applications in the oil and gas domain related to drilling (Veettil and Clark 2020, Jahani et al. 2021), petrophysics (Sen, Ong, et al. 2020b), reservoir engineering (Yousef et al. 2006, Tian and Horne 2019, Sen, Chen, et al. 2020, Liu et al. 2019) and production. The current study explores various methodologies to leverage the power of machine-learning for building workflows that can perform tasks such as connectivity detection and rate optimization in an efficient way. The study is broadly aimed at inferring the reservoir connectivity between injectors and producers in a field under recovery operations such as waterflood or CO_2 water alternating gas (WAG) injection.

Chapter II : Data-Driven Rate Optimization under Geologic uncertainty

Waterflood is carried out by injecting water into the reservoir through injectors to increase/maintain reservoir pressure while oil is being extracted at the producers. Well rates optimization is critical to a successful waterflood project via improved injection efficiency and reduced water cycling. However, waterflood optimization can be a challenging task due to the number of parameters involved, the computational load required for repeated numerical reservoir simulation and the inherent uncertainty in the reservoir models. Chapter II of this dissertation describes the development of a neural network based proxy model that estimates the streamline-based connectivity between injectors and producers during a waterflood. This method is also an attempt at using a data-driven technique to infer an aspect of flow physics (i.e. the TOF) under the constraints of flow rates and heterogeneity, and using this to perform rate optimization in a quick manner. The computational efficiency of the developed proxy model makes it suitable for rate optimization under geologic uncertainty wherein the optimization has to be run on a large number of realizations.

Chapter III: Model-Free Assessment of Inter-well Connectivity using Statistical Recurrent Unit Models

The statistical recurrent unit (SRU) was introduced by Oliva et al. (2017) as an un-gated alternative to more complex RNN architectures such as LSTMs (Hochreiter and Schmidhuber 1997) and GRUs (Chung et al. 2015). The SRU was shown to capture long term dependencies at least as much as the latter models, simply by using moving averages of temporal information at different scales. Therefore, the SRU architecture is simpler and more interpretable compared to LSTMs and GRUs, enabling easier training and incorporation of additional constraints such as well locations. In this chapter, we propose an SRU based framework for inferring inter-well connectivities. The proposed workflow contains two key components: SRU model specifically framed to the CO2 WAG problem and the inter-well connectivity assessment based on the SRU model and variable importance calculation.

Variable importance is an assessment of the contribution of each input on the output (Breiman 2001). It can be a promising method of inferring well connectivity from a data-driven model. That is, the variable importance computed from an explainable data-driven model, should ideally correspond to the actual influence that an input signal (such as those from an injector) has on the output (production at producer). In this way, an explainable and predictive data-driven model may be used to derive insights on the actual reservoir connectivity in a quick and efficient way, without the need to run computationally costly reservoir simulations.

Chapter IV: Identification of Key Wells for Optimization Considering Geologic Uncertainty

In Chapter II, it is established that the optimal schedule based on one individual geologic model may not necessarily result in favorable outcomes for the real field due to the geologic inconsistencies between the real and the model. Therefore, the geologic uncertainty is typically represented by an ensemble of history matched realizations of the reservoir. Subsequently, optimal schedules for each well shall also be expressed as a distribution rather than single values. The corresponding uncertainty assessment is crucial to decision making regarding field implementation. In this chapter, we focus on formulating and accelerating the generation of the distribution of optimal rates of each well in waterflood projects, while considering geologic uncertainty.

We start with the generation of history-matched geologic realizations, given a set of observed data. Subsequently the k-means algorithm is used to cluster the time-of-flight field at the end of history, to group the realizations into N_{clust} sets, each with its representative realization (centroid). In this manner, the geologic uncertainty described by N_{real} ($N_{real} = 200$ for SAIGUP case) realizations are represented using merely $N_{clust} =$

15 centroid realizations. Rate optimization is performed solely on the centroid realizations and the distribution of rate changes assigned to each well was shown to be similar to the values obtained by optimizing the entire ensemble. The uncertainty analysis generated using the TOF-based centroid realizations may be used to inform the selection of key wells for implementation of rate change.

CHAPTER II

DATA-DRIVEN RATE OPTIMIZATION UNDER GEOLOGIC UNCERTAINTY[2]

Waterflood is by far the most extensively used improved oil recovery method post primary depletion. This is carried out by injecting water into the depleting reservoir through injectors to increase/maintain reservoir pressure while oil is being extracted at the producers. This process has been explored by studies at various scales, ranging from nanoscale (Xiong et al. 2019) to field scale management (Chen et al. 2020b) due to the strong impact of heterogeneity over the subsurface flow. The presence of high-permeability streaks in a highly heterogeneous reservoir may lead to early water breakthrough at the producers, resulting in premature rate decline and reduced oil recovery per barrel of water injected. Hence, optimization of well rates during a waterflood is critical to an effective waterflood project via improved injection efficiency and reduced water cycling. However, waterflood optimization can be a challenging task due to the number of parameters involved, the computational load required for repeated numerical reservoir simulation and the inherent uncertainty in the reservoir models.

---

[2] Part of this chapter is reprinted with permission from Sen, Deepthi, et al. "Data-Driven Rate Optimization Under Geologic Uncertainty." SPE Annual Technical Conference and Exhibition. OnePetro, 2020. Copyright 2020 Society of Petroleum Engineers. Further reproduction is prohibited without permission.

[2] Part of this chapter is reprinted with permission from Sen, Deepthi, et al. "Machine learning based rate optimization under geologic uncertainty." Journal of Petroleum Science and Engineering 207 (2021): 109116.

Asheim (1988) combined two-phase reservoir simulation with numerical optimization via total differentiation of the simulator equation. Optimization was performed to maximize net present value (NPV) subject to constraints on reservoir flow dynamics, total flow capacity and well constraints.

Sudaryanto and Yortsos (2001) developed an injection optimization workflow for single-phase incompressible flow based on optimal control theory. This was tested on a rectangular bounded system with two injectors and one producer. The optimal strategy for the test cases considered was found to be of the bang-bang type where the injectors operate at extreme values in the allowable range. Balaji et al. (2017) incorporated geomechanics into a bang-bang type optimization scheme.

Zhai et al. (2016) inferred interwell connectivity between well rate allocations through numerical tracer experiments. The time dependence of oil cut in the production was modeled using a sigmoid decline. This proxy model was used to perform rate optimization on a two-phase system.

Nævdal et al. (2006) developed a closed-loop control approach using ensemble Kalman filters (EnKF) in order to maximize the NPV subject to constraints. This approach requires a simulator to be run for each member in the ensemble, so that a control vector that maximizes the objective function can be computed by solving the adjoint equation using steepest descent. Wang et al. (2009) developed a closed loop reservoir management workflow that maximizes the NPV and tested and compared three kinds of optimization techniques – steepest descent, simultaneous perturbation stochastic approximation (SPSA) and EnKF. Steepest descent was found to be the most effective for the test cases studied. Chen et al. (2010) also applied an EnKF with covariance localization for data assimilation in the SPE Brugge case (Peters et al. 2010) and performed rate optimization on the mean model from the ensemble. Other non-linear filtering schemes such as the unscented Kalman filter (Wan and Van Der Merwe 2000) and various kinds of particle filters (Gustafsson et al. 2002) may also be employed for similar closed-loop control approaches to waterflood optimization. In particular, Raihan and Chakravorty (2008a, 2008b) have developed an efficient non-linear filtering scheme that overcomes issues such as particle depletion, which may potentially be incorporated into a waterflood optimization setup.

Field-scale waterflood optimization using rate control has been carried out by delaying the time of waterfront breakthrough at the producers. Grinestaff (1999) and Grinestaff and Caffrey (2000) introduced the concept of streamlines to production optimization by deriving qualitative insights from streamline simulation of the North West Fault Block in Prudhoe Bay. Brouwer et al. (2001) performed rate optimization by reducing the distribution in the arrival times of the flood front for a time-independent flow field. This approach was extended to time-dependent flow fields in Brouwer and Jansen (2002). Alhuthali et al. (2007) proposed a methodology that involves equalizing the arrival time of the waterfront at producers in separate subregions of the reservoirs. The arrival time was calculated by running a reservoir simulator and tracing streamlines explicitly. Chen et al. (2020b) proposed a streamline-based gradient-free method for maximizing oil production at the end of production time under constraints such as fieldwide injection and production rates, minimum well producing pressure and allowable ranges on well rates. Using this method, it is required to solve the flow equation and trace streamlines using the computed fluxes. The optimization proceeds by equalizing the well-pair efficiencies rather than the arrival times as proposed by Alhuthali et al. (2007). Van Essen et al. formulated an approach to optimize well rates under geologic uncertainty wherein the objective function was computed as the expected value of cumulative revenue over several geologic realizations. Alhuthali et al. (2008) applied this methodology to develop a streamline-based rate optimization scheme under geologic uncertainty that equalizes the arrival times in a probabilistic sense over multiple realizations. The methods described above requires solving the flow partial differential equation (PDE) and this may not be computationally feasible for large-scale field applications when one has to optimize over multiple realizations.

In recent years, machine-learning based methods have found considerable success in applications related to drilling (Veettil and Clark 2020), petrophysics (Sen et al. 2020b) and reservoir engineering (Yousef et al. 2006). The computational demand associated with performing field-wide reservoir simulation has prompted the use of machine-learning based proxy models that mimic the reservoir response, given a set of inputs. Proxy modeling is especially useful in applications such as field-wide waterflood optimization, which may otherwise require several iterations and multiple simulation runs per iteration for convergence. Tian and Horne (2016) proposed a machine-learning based model for quantifying injector-producer connectivity using a modified Pearson correlation coefficient between injection and production histories. The computed connectivities were validated using a machine-learning based multiwell testing scheme that learns the variation in injection pressures with respect to various features defined on the production rates. Even though this workflow provides useful insights on the reservoir geology without assuming a prior model, the trained model cannot be directly used for rate optimization since this does not function as a proxy model that can predict waterflood performance given the rates and a geologic model.

One of the most commonly used proxy models for waterflood optimization– the capacitance resistance model (CRM) was developed by Yousef et al. (2006) for predicting production trends in mature fields under waterflooding. As a first-principle model, the CRM treats the reservoir system as an RC electrical circuit wherein the resistance and the capacitance represent the interwell connectivity and drainage volume, respectively. These parameters are estimated by fitting injection and production data to a CRM equation. This concept has been explored in detail and the results of its applicability under various conditions are available in the literature (Cao et al. 2014, Sayarpour et al. 2008). In particular, Prakasa et al. (2017) and Kansao et al. (2017) applied CRM-based techniques for waterflood optimization. However, since the CRM model captures the field heterogeneity information implicitly from well history, it is not possible to train a single CRM over multiple realizations of geology. Lee et al. (2010) developed a method for improved CRM model fitting using hybrid constraint nonlinear optimization. However, the method was only tested for small test cases. Lin et al. (2010) used CRM to detect the existence and orientation of high permeability streaks between injectors and producer by characterizing the lag time for each injector-producer pair. This information may be used to gain insights for the rate optimization process.

Liu et al. (2012) modeled the reservoir as a collection of independent subsystems, each consisting of one injector and one producer. The reservoir response was modeled as an aggregation of impulse responses to injection rates. The rate optimization was formulated as the maximization of a linear objective function (total oil produced) in the discrete frequency domain, subject to linear constraints. Nwachukwu et al. (2018) used a machine-learning based approach to predict NPV given the well locations, well rates and well-block properties. The primary inputs were used to generate secondary features such as 'diffusive time of flight' (as a measure of connectivity) and well angle, which were subsequently fed into a gradient boosted machine learning model (XGBoost). Diffusive time of flight, defined as the travel time of the pressure wave front, is a measure of well connectivity and is calculated using the Fast Marching Method (FMM). They also studied the effects of geological uncertainty by using an ensemble of realizations. However, they did not extend the workflow to larger field cases.

In this work, we adopt the workflow proposed by Alhuthali et al. (2007) and Alhuthali et al. (2008) in performing rate optimization by equalizing arrival times over multiple realizations by employing a machine-learning based proxy model for estimating the arrival time, given the well rates and reservoir property fields. Our method is also an attempt at using a data-driven technique to infer an aspect of flow physics (i.e. the TOF) under the constraints of flow rates and heterogeneity, and using this to perform rate optimization in a quick manner. This is in contrast with approached that use proxy models to directly predict the NPV, which may be susceptible to several operational and economic factors. The rest of the chapter is organized as follows: In Section 2, we outline the approach to rate optimization that has been implemented in this chapter. Section 3 details the mathematical background behind our workflow. Section 4 describes the results obtained on applying the workflow first to a simple 2D heterogeneous case and then to a 3D field case. Section 5 summarizes the chapter and the directions for future efforts.

Approach

Consider a heterogeneous two-phase reservoir with N_I injectors, injecting at rates $\boldsymbol{q_I} = [q_{I_1}, \dots, q_{I_{N_I}}]$ and N_P producers, producing at rates $\boldsymbol{q_P} = [q_{P_1}, \dots, q_{P_{N_P}}]$. We wish to compute the optimal rates $\boldsymbol{q_I^*}$ and $\boldsymbol{q_P^*}$ that maximize the cumulative oil production. This is achieved by maximizing the sweep efficiency of the waterflood defined as

$$\eta = \frac{PV_{swept}}{PV_{total}} \tag{1}$$

Here PV_{total} is the total pore volume of the reservoir model and PV_{swept} is the sum of pore volumes of gridblocks that have been ‘swept’ by the fluid front at a given time. Alhuthali et al. (2007) showed that η is maximized when the arrival times of the injected fluid front at the producers are equalized.

The arrival time of the injected fluid front at producer P_i is represented by the minimum time of flight (TOF), calculated as the average TOF of the fastest 20% streamlines that end in P_i. This quantity is denoted by τ_i (Alhuthali et al. 2007). The TOF represents the travel time of a neutral tracer along the streamlines. Typically, the evaluation of τ_i requires calculating the flux in the reservoir system and subsequent streamline-tracing using Pollock's method (Pollock 1988). We circumvent the need to compute the flux and streamline tracing by using a trained proxy model that directly predicts τ_i given the well rates and geology. This is a key differentiating aspect of this work, wherein we develop a machine-learning based proxy for a physics-informed intermediate output, which is subsequently used for well rate optimization.

Once we have the trained model, we may use it in place of the forward simulator in order to compute the rates that equalize the TOFs at the producers. Additionally, due to the speed of the forward run of the proxy model, we can easily compute well rates that equalize the TOF in a probabilistic sense over multiple geologic realizations. The overall workflow is illustrated in Figure 1. The workflow is discussed more closely in the rest of this section.

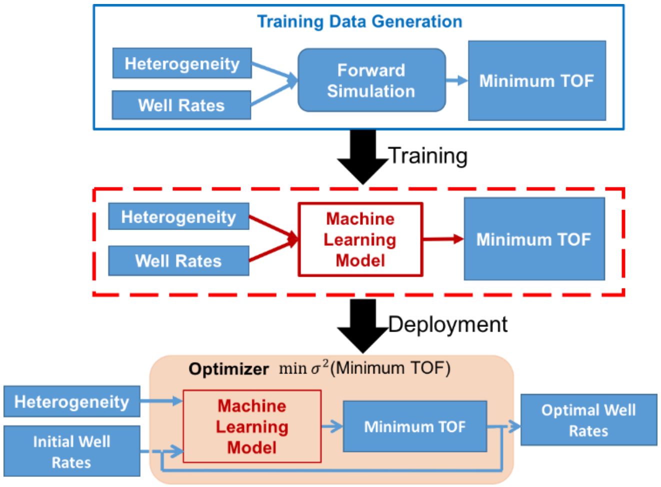


Figure 1 Proposed workflow for rate optimization using proxy model for time of flight (TOF) connectivity. (Reprinted with permission from Sen et al., 2020a)

Proxy model building

Assume that the uncertainty in geology is characterized by N_F heterogeneous fields $\boldsymbol{F_i}$, $1 \leq i \leq N_F$ where $\boldsymbol{F_i}$ is a vectorized representation of the gridblock-wise reservoir properties such as permeability and porosity. In order to build the proxy model, we use grid-connectivity transformation (GCT), explained in detail under 'Mathematical Formulation', to compute a low-dimensional representation of each of the geologic characterization.

The proxy model f that is used in this work is such that $\boldsymbol{\tau} = f(\boldsymbol{x})$ where $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_{N_P}]$ and $\boldsymbol{x} = [\boldsymbol{q_I}|\boldsymbol{q_P}|\boldsymbol{v_i}]$. Here, $\boldsymbol{v_i}$ denotes the GCT basis coefficient vectors of the geologic fields $\boldsymbol{F_i}$ considered. A representation of the training dataset for the proxy model, demarcating the inputs and outputs is given in Figure 2.

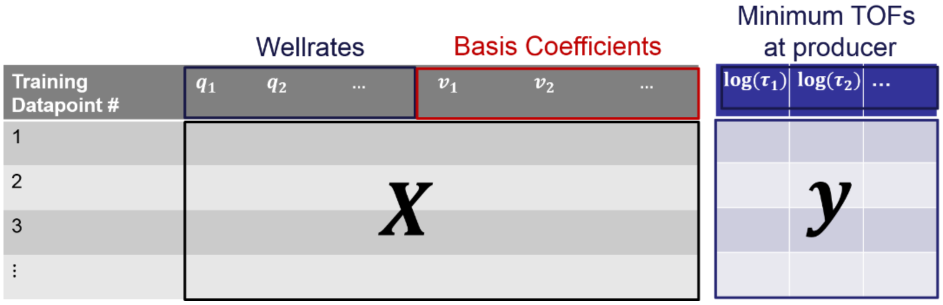


Figure 2 Schematic representation of the training dataset for proxy model building showing inputs and outputs. (Reprinted with permission from Sen et al., 2020a)

Dataset Generation

The input dataset for training the model, $\boldsymbol{X}$, consists of N_{train} observations (rows) of $\boldsymbol{x}$. The corresponding output dataset $\boldsymbol{y}$ consists of N_{train} rows of $\boldsymbol{\tau} \in \mathbb{R}^{1\times N_P}$, computed by running the reservoir simulator and tracing streamlines for each instance in $\boldsymbol{X}$. Since τ for each of the producers may be different by several orders of magnitude, log-transformed τ is used to form $\boldsymbol{y}$ for better fit of the training data.

Artificial Neural Network

Artificial neural nets (multilayered percepteron (MLP)) were used to build the proxy model for predicting the TOF in this study, although any suitably trained algorithm would be admissible. The proxy model was constructed as an MLP, with N_{MLP} layers and n_{neu_i} neurons in the i^{th} layer ($1 \leq i \leq N_{MLP}$) as shown in Figure 3. The neurons in the output layer ($i = N_{MLP}$) have linear activation functions. All other neurons use a ReLU activation function σ_{ReLU}, where

$$\sigma_{SS}(x) = max(0, x) \tag{2}$$

The number of neurons in the output layer ($i = N_{MLP}$) is equal to the dimensionality of the output. Dropout layers are added after each layer of neurons to limit overfitting. Dropout works by randomly deactivating a fraction of the neurons during the training phase. Dropout layers are bypassed when the model is used for prediction. The model is assembled using Keras library with Tensorflow backend (Chollet 2015).

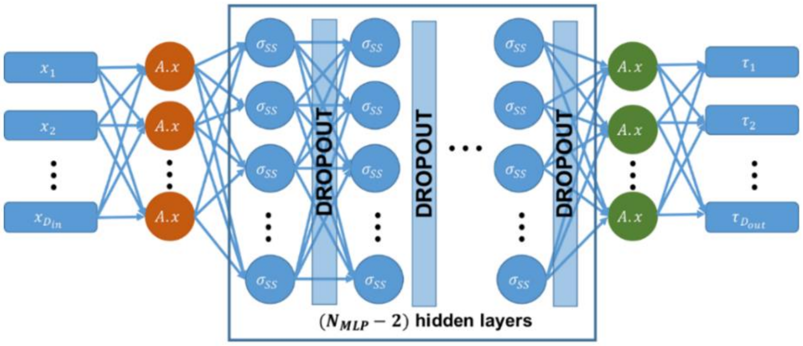


Figure 3 A fully connected multi-layer percepteron used to build the proxy model. The neurons in the hidden layers have ReLU activation whereas the output layer is linearly activated. (Reprinted with permission from Sen et al., 2020a)

K-fold cross-validation

The selection of the best model parameters is guided by its generalization performance, which indicates the prediction capacity of the model on a test dataset, independent of the dataset on which the model was trained on. K-fold cross-validation is a commonly used method for method selection in machine-learning literature (Hastie et al. 2009). It provides an estimate of the expected generalization error of the model over several training datasets. During k-fold cross-validation (Mishra and Datta-Gupta 2017), the entire training dataset is randomly split into k segments. A prediction of each of the k segments is generated by training the model on the remaining k-1 segments. This way, the average loss is computed over the entire training dataset (Figure 4). This is repeated for different models, by varying the model parameters such as the number of hidden layers and number of neurons and the average loss is recorded. The model parameter set that minimizes the average loss is chosen. Finally, the chosen model is re-trained on the entire dataset.

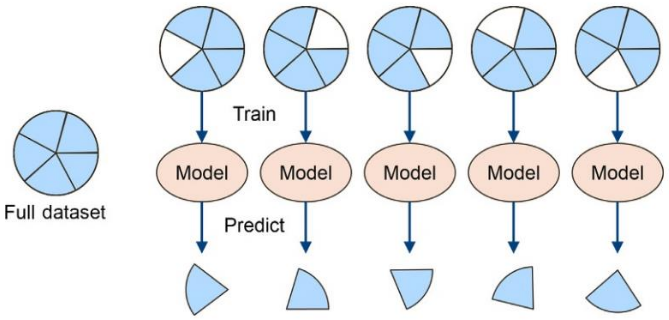


Figure 4 An illustration of k-fold cross validation for k=5 (Mishra and Datta-Gupta 2017).

Rate Optimization Using Proxy Model

The rate optimization framework used in this chapter follows the workflow of Alhuthali et al. (Alhuthali et al. 2008, Alhuthali et al. 2007). For optimizing the rates for a single realization, we first divide the wells into a number (N_{group}) of subgroups based on the flux allocations using spectral clustering as explained in a later section. Subsequently, we compute the well rates that minimize the sum of variances of $\tau's$ in each of the subgroups. In such a case, we assume that we know the geologic characterization with fair degree of certainty.

This rate optimization framework was extended to incorporate geologic uncertainty by modifying the objective function as the expectation of the variances over multiple geologic realizations. Sequential Least Square algorithm implemented in the python library (Scipy) is used for solving the minimization problem with constraints.
The details of rate optimization are provided below.

Mathematical Formulation

Streamline Tracing

Streamline tracing involves computing the path traced by a tracer particle that enters the reservoir through an injector till it reaches a termination point: a producer, an aquifer or a stagnant region. Given a steady state flow-field within the reservoir, streamlines are fixed in time. The time-of-flight (TOF) from an injector at location l is defined as the time a neutral tracer would take to travel from the injector to l. Mathematically, it is defined as

$$\boldsymbol{\tau_l = \int \frac{\phi}{u} ds} \tag{3}$$

where u is the Darcy velocity, ϕ is the porosity and s represents streamline connecting l to the injector. Also, τ_l is a function of the interstitial velocity given by $\frac{\phi}{u}$, which is in turn dependent on the pressure at each gridblock, and hence on the permeability and porosity fields and wellrates.

In practice, once the reservoir simulator computes the pressure field and subsequently fluxes in the reservoir, streamline tracing is performed using the Pollock's method (Pollock 1988).

Rate optimization

We adopt Alhuthali's workflow that involves equalizing the arrival time of the waterfront at producers in distinct subregions of the reservoir. This effectively maximizes the sweep efficiency of the waterflood project at breakthrough. The equalization of TOF, as a means to maximize sweep and thereby achieve improved oil recovery, is preferred in this study since this bypasses the need to run the simulator for the entire duration of well activity. TOF is computed from the flux field obtainable from a single timestep simulation with the assigned well rates. This makes the workflow quite fast, despite the need to run the simulator to generate the training set for cases where the streamline configuration remains fairly static.

Each sub-region is associated with a well subgroup located within it. In Alhuthali et al. (2007), these subregions and associated subgroups of wells are chosen by the user. In the proposed work, the user specifies the number of subgroups, N_{group}. Spectral clustering, explained in a latter section, is used to choose the wells in each subgroup. The original procedure broadly involves two steps:

a. Flow simulation and subsequent streamline tracing

b. Computation of the difference e between the arrival time at each producer P_i $(1 \leq i \leq N_P)$ and the desired arrival time for the subgroup to which P_i belongs. Thus, given then P_i belongs to m^{th} subgroup G_m $(1 \leq m \leq N_{group})$:

$$\boldsymbol{e_i = \tau_{desired,m} - \tau_i} \tag{4}$$

Here $\tau_{desired,m} = \frac{\sum_{i:P_i \in G_m} \tau_i}{N_{G_m}}$ where N_{G_m} denotes the number of producers in G_m. The arrival time τ_i is calculated as the average TOF of the fastest 20% streamlines at i^{th} producer, as explained in Alhuthali et al. (2007).

The minimization problem for finding optimal well rates for N_I injectors $\boldsymbol{q_I} = [q_{i1}, \dots, q_{iN_I}]$, and N_P producers $\boldsymbol{q_P} = [q_{p1}, \dots, q_{pN_P}]$ is formulated as

$$\min_{\boldsymbol{q}} \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{e^T e} \tag{5a}$$

subject to N_{eq} equality constraints and N_{ineq} inequality constraints:

$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0} \tag{5b}$$

$$\boldsymbol{g}(\boldsymbol{q}) \leq \boldsymbol{0} \quad where\, \boldsymbol{h}\colon \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}^{\boldsymbol{N_{eq}}} \; and\, \boldsymbol{g}\colon \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}^{\boldsymbol{N_{ineq}}} \; and\, \boldsymbol{q} = [\boldsymbol{q_I} | \boldsymbol{q_P}].$$

Here

$$\boldsymbol{e^T e} = \sum_{\boldsymbol{m=1}}^{\boldsymbol{N_{group}}} \sum_{\boldsymbol{i:P_i \in G_m}} \left(\boldsymbol{\tau_{desired,m}} - \boldsymbol{\tau_i}(\boldsymbol{q})\right)^{\boldsymbol{2}} \tag{6}$$

In field applications, the constraints can be related to wells, group of wells, gathering stations or the field as a whole (Taware et al. 2017).

This rate optimization framework was extended to uncertain geologies by modifying the objective function as the expectation of the residual over multiple geologic realizations (Alhuthali et al. 2008).

Therefore, for the j^{th} realization in a pool of N_{real} realizations,

$$(\boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e})_{\boldsymbol{j}} = \sum_{\boldsymbol{m}=\mathbf{1}}^{\boldsymbol{N}_{\boldsymbol{group}}} \sum_{\boldsymbol{i}:\boldsymbol{P}_{\boldsymbol{i}}\in\boldsymbol{G}_{\boldsymbol{m}}} \left(\boldsymbol{\tau}_{\boldsymbol{desired,m}} - \boldsymbol{\tau}_{\boldsymbol{i}}(\boldsymbol{q})\right)^{\mathbf{2}} \tag{7}$$

The expected value and standard deviation of the squared residual are given by

$$\mathbb{E}(\boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e}) = \frac{\sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{N}_{\boldsymbol{real}}}(\boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e})_{\boldsymbol{j}}}{\boldsymbol{N}_{\boldsymbol{real}}} \tag{8}$$

The minimization problem is then formulated as

$$\min_{q} f(\boldsymbol{q}) = \mathbb{E}(\boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e}) \tag{9}$$

Low dimensional projection of heterogeneous field

A key feature of this work is the representation of the heterogeneous field using a compact set of orthogonal basis functions that are constructed from the grid-connectivity information of the reservoir model. This representation is referred to as grid connectivity transformation (GCT) of the heterogeneous field (Bhark et al. 2011). The linear mapping of the high-dimensional field to the spectral domain provides an efficient way to parameterize the heterogeneity using a small number of features vectors. However, it is to be noted that any low dimensional representation method may be used to parameterize the heterogeneous field. The simplest way to parameterize spatial properties would be to transform these into another domain via methods such as principal component analysis (PCA) based algorithms (Sarma et al. 2007, Ma et al. 2011), Karhunen–Loève transform (KLT) (Reynolds et al. 1996), sparse dictionary learning (Khaninezhad et al. 2010) and discrete cosine transformation (DCT) (Jafarpour and McLaughlin 2009). Dorn and Villegas (2008) used a level-set method to parameterize heterogeneity in a reservoir and subsequently perform history-match. More recently, Park and Caers (2020) applied PCA to parameterize features of very high dimensions and used scree plot to select the optimal number of basis functions, similar to the approach adopted in our study. Additionally, deep-learning based algorithms such as variational autoencoders (VAE) (Gundersen et al. 2020) and convolutional neural networks (CNN) (Alakeely and Horne 2020) have been applied successfully in reservoir characterization problems. For a comparison of recent reservoir parameterization methods, the reader may refer to Zhang et al. (2021). However, deep-learning-based algorithms for parameterization entail significant computational effort and a large dataset size to successfully train a complex neural network. Therefore, for the purpose of our study, we use GCT which is a simpler and faster algorithm that only requires an SVD decomposition and grid structure.

In order to apply GCT, the connectivity of reservoir grid is characterized as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, wherein the vertice set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ represent the N grid cell centers and the edge set $\mathcal{E}$ represents all connections between the vertices in $\mathcal{G}$. An edge between vertices $\mathscr{v}_i$ and $\mathscr{v}_j$ is assumed to exist if the i^{th} and j^{th} grid-cells share a face. Given $\mathcal{G}$, the graph connectivity is described by Laplacian matrix L which may be constructed as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{10}$$

where W is the similarity matrix that indicates the weights of connection between all vertices in $\mathcal{G}$. For our application,

$$\boldsymbol{W_{ij}} = \begin{cases} \mathbf{1} & \boldsymbol{if\ connection\ exists\ between\ \mathscr{v}_i\ and\ \mathscr{v}_j} \\ \mathbf{0} & \boldsymbol{otherwise} \end{cases} \tag{11}$$

In eqn 10, D denotes the degree matrix of the graph, which denotes the strength of connections in each vertex of the graph. D is in the form of a diagonal matrix, whose i^{th} diagonal element is the row-wise sum of the i^{th}column of W.

Eigen-decomposition of L yields a set of N orthogonal basis vectors $\boldsymbol{\Phi_0} = [\Phi_1|\Phi_2| \ldots |\Phi_\mathrm{N}]$ where each basis vector Φ_i represents a harmonic of the grid structure. Any spatial field $\boldsymbol{F}$ can be linearly mapped to any Φ_i to yield a GCT basis coefficient $\boldsymbol{\nu_i}$ that represents the amplitude of $\boldsymbol{F}$ in the modal frequency corresponding to Φ_i**.** Thus $\boldsymbol{F}$ may be reparameterized in terms of

$$\boldsymbol{\nu} = \boldsymbol{\Phi^T F} \tag{12}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N \times N_B}$ is a subset of $\boldsymbol{\Phi_0}$**,** containing N_B out of the original N basis vectors, such that $N_B \ll N$. Hence the high-resolution spatial field $\boldsymbol{F} \in \mathbb{R}^{N \times 1}$ is represented as a vector $\boldsymbol{\nu} \in \mathbb{R}^{N_B \times 1}$**.**

Spectral Grouping of Wells

We propose the use of spectral clustering to form well groups (Narasingam and Kwon 2017) for proxy modeling and subsequent rate optimization, instead of a user-specified grouping scheme as presented by Alhuthali et al. (2007). Spectral grouping is performed based on the injection rate-allocation, which is defined as the fraction of injected fluid from an injector allocated to each producer.

During training phase, we record N_{train} values of the 'rate allocation' matrix R defined by $R_{ij} = \frac{q_{ij}}{q_i}$ where q_{ij} is the volume of fluid that flows to the j^{th} producer from the i^{th} injector. This information is readily available from the streamlines and the associated fluxes along the streamlines. The average rate allocation matrix R_{mean} is calculated over the available training dataset. The connectivity between the wells is characterized by a graph $\mathcal{G}_{well} = (\mathcal{V}_{well}, \mathcal{E}_{well})$. Here, $\mathcal{V}_{well}$ is the set of vertices in the graph, which represent the $N_P + N_I$ wells. $\mathcal{E}_{well}$ represents all connections in $\mathcal{G}_{well}$. An edge between vertices $v_{well,i}$ and $v_{well,j}$ is assumed to exist if the i^{th} and j^{th} wells have a positive value for rate allocation. It is assumed that no edge exists between injectors and between producers and all connections are undirected.

Therefore, the similarity matrix W_{well} and Laplacian L_{well} for $\mathcal{G}_{well}$ can be represented as

$$\boldsymbol{W_{well}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{R_{mean}} \\ \boldsymbol{R_{mean}^T} & \boldsymbol{0} \end{bmatrix} \tag{13}$$

$$\boldsymbol{L_{well}} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{R_{mean}} \\ -\boldsymbol{R_{mean}^T} & \boldsymbol{diag}(\boldsymbol{sum}(\boldsymbol{R_{mean}}, \mathbf{1})) \end{bmatrix} \tag{14}$$

The eigen decomposition of L_{well} can be written as

$$\boldsymbol{L_{well}} = \boldsymbol{U \Lambda U^T} \tag{15}$$

The first k eigenvectors (corresponding to the k smallest eigenvalues) may be used to cluster $\mathcal{V}_{well}$ into k subgroups (Von Luxburg 2007). The value of k is predecided by the user.

Applications

2D Heterogeneous Case

To start with, the proposed algorithm was applied to a 2D 50x50 heterogeneous field, with uncertainty in permeability characterization. A pool of 1000 realizations of the permeability field was generated using Sequential Gaussian Simulation (SGS) for different values of variogram range and azimuth. For illustrative purposes, a few of the realizations are shown in Figure 5. The porosity is set to a constant value of 0.2 across the field.

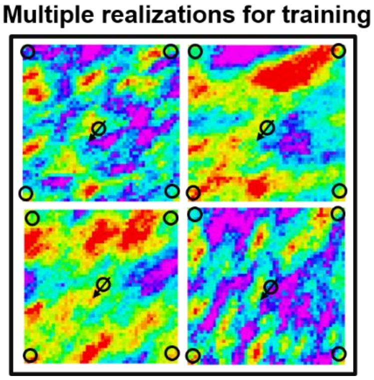


Figure 5 A pool of equiprobable k_X realizations are generated using SGS for training the model (reprinted with permission from Sen et al., 2020a)

Table 1 Parameter distributions used in dataset generation for 2D case proxy model

Parameter	Distribution
Variogram range (ft)	$\text{Discrete}[10,20,30]$
Variogram azimuth (°)	$\text{Discrete}[25,65]$
Production rate, q_P (bbl/d)	$U[500,4000]$
Injection rate, q_I (bbl/d)	$U[1000,8000]$

Proxy model building

The model was trained on 600 realizations from the generated pool of 1000 realizations. For every training datapoint, one of the 600 realizations was chosen randomly, and well rates were assigned randomly from distributions given in Table 1. Flow simulation was then carried out using ECLIPSE (Schlumberger 2018) and streamline tracing was performed using our in-house post-processing software DESTINY (Chen et al. 2020b).

A total of 1000 training datapoints were generated in this way. The detailed process is given below.

GCT Transformation

The first step of our workflow consists of representing each of the heterogeneous fields as a linear combination of GCT basis functions. The GCT basis functions for the 50x50 grid is computed by setting up the graph Laplacian matrix and its subsequent eigen-decomposition. It is to be noted that this is independent of the property field and needs to be done just once during the entire workflow. We compute the first 300 basis functions, which correspond to the smallest 300 eigenvalues of the graph Laplacian.

As explained earlier, each of the property fields may be expressed as a linear combination of these basis functions and its basis coefficients (which are property-dependent). This concept is illustrated in Figure 6. Here the original permeability field is expressed as a linear combination of the property dependent basis functions $v_1, v_2, \ldots$ and the GCT basis functions that represent a particular modal frequency of the grid structure. A higher resolution representation is obtained by increasing the number of basis functions used in the representation.

However, a more efficient way to represent the field would be to choose the basis functions that are associated with higher magnitudes of basis coefficients. This may be done by sorting the basis coefficients for a particular field, in the descending order of magnitude and choosing the first few basis coefficients (and its associated basis functions). In this work, since we have several realizations of permeability, we compute the “average” permeability across all training realizations. The drop-off in basis coefficient magnitude (of the mean permeability field) with an increasing number of sorted coefficients is shown in Figure 7. We choose the first 10 basis functions (corresponding to the highest 10 basis coefficients) out of the computed 300. As seen in Figure 7, the drop off is very steep up till around 4 basis functions. Hence we can safely assume that 10 basis functions would capture most of the heterogeneity. The additional buffer of 6 basis functions was included keeping in mind the strong dependence of the predicted variable to the heterogeneity. We see that the accuracy of the model improves with increasing number of basis functions. Using merely 4 basis functions may yield a good reconstruction of the field. However, the information captures from 4 basis functions may be insufficient for TOF prediction due to the latter’s strong dependence on heterogeneity.

In case we want to represent two or more properties, this procedure is to be repeated for each property. Therefore, if we were to represent porosity, we would use the basis functions corresponding to the highest basis coefficients for the ensemble mean of the porosity.

Subsequently, each of the permeability realizations is expressed in terms of these 10 basis functions. The reconstruction of a reference permeability field using sorted basis functions is shown in Figure 8.

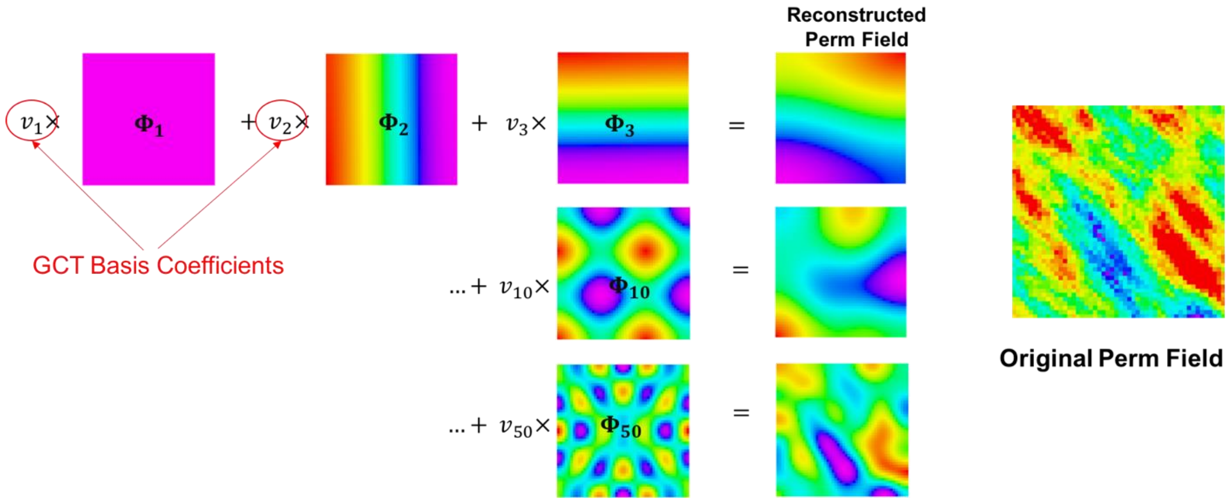


Figure 6 Representation of a k_X realization in terms of GCT basis functions and coefficients for the 50x50 grid (reprinted with permission from Sen et al., 2020a)

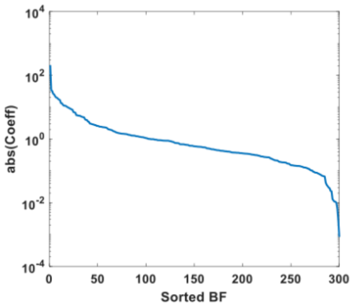


Figure 7 Drop-off in magnitude of basis-coefficient with increasing number of basis functions. Only the first 10 are chosen as input features to proxy model for 2D case (reprinted with permission from Sen et al., 2020a)

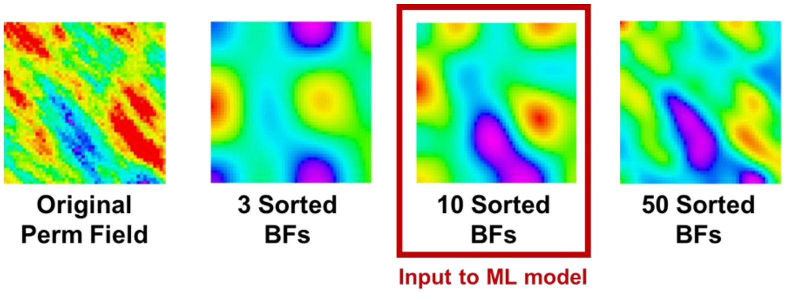


Figure 8 Reconstructed perm field using basis functions sorted based on magnitude of basis coefficients of the ensemble mean field of the training dataset (reprinted with permission from Sen et al., 2020a)

Streamline tracing

The next step in the workflow is to assign randomly sampled rates (both injection and production) all the wells. The reservoir simulations are performed using ECLIPSE running on Intel® Xeon® CPU with a base processor speed of 2.2 GHz. Once the flow field is computed, DESTINY is used to trace streamlines. DESTINY calculates the TOF for each of the streamlines that end at each producer. The average time of flight of the fastest 20% of all the streamlines ending at the i^{th} producer is calculated and stored as τ_i.

Model Selection using k-fold cross-validation

Six different architectures of neural networks tested are shown in Table 2. We perform 5-fold cross-validation, as described under the section ‘Proxy model building’, by means of which the dataset is split into 5 sections randomly. At each iteration of 5-fold cross-validation, a model is trained on 4 out of 5 folds and validated on the remaining fold. In other words, each of the models is trained on a dataset under an 80%-20% training/validation split. Furthermore, 5-fold cross-validation ensures that each datapoint serves as validation exactly once. The individual and total RMSE calculated from this validation performance across the entire dataset is given in Table 2.

We note that there is a reduction in the total RMSE with increasing model complexity. A model consisting of a single hidden layer with 50 neurons is chosen since the RMSE stabilizes at this point. While one could choose a more complex network, the increase in accuracy achieve by doing so is meagre.

The neural network was trained for 500 epochs using the Adam optimizer with a learning rate of 0.001. The batch size was chosen to be 400. The predictions on the entire dataset via 5-fold cross-validation is shown in Figure 9. The primary reason for the scatter in the predictions is the non-linearity of the predicted variable (TOF), and the parsimonious representation of heterogeneity through basis functions which focuses on preserving the large scale continuity rather than small scale variations in the heterogeneity. We also see some bias in the predictions of higher values of TOF. However, the impact of these predictions on optimization is expected to be marginal because the higher TOFs tend to represent the stagnant areas in the velocity field.

Table 2 Model Selection using 5-fold cross-validation

Architecture	**RMSE**				**Total RMSE**
	P1	**P2**	**P3**	**P4**	
[10]	0.42	0.44	0.49	0.53	0.94
[20]	0.4	0.42	0.46	0.49	0.88
[50] (Chosen)	0.4	0.4	0.45	0.46	0.85
[100]	0.39	0.39	0.44	0.45	0.84
[50,50]	0.39	0.39	0.44	0.45	0.84
[100,100]	0.39	0.38	0.43	0.44	0.83

Table 3 Neural network architecture for proxy modeling (2D case)

Layer no.	$\boldsymbol{N_{neurons}}$	**Activation**	**Dimensions**	
			Input	**Output**
1 (Input layer)	0	N/A	N/A	$N_I + N_P + N_{basis} = 15$
2	50	ReLU	15	50
3 (Output layer)	$N_P = 4$	Linear	50	4

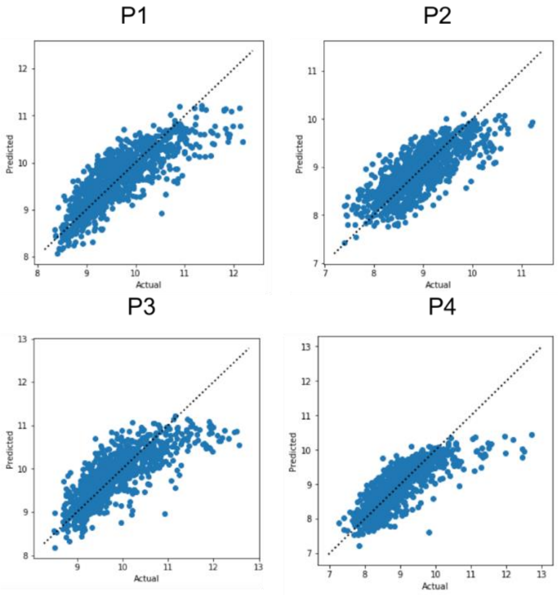


Figure 9 Model fit on test set using MLP with a single hidden layer with 50 neurons Rate Optimization (reprinted with permission from Sen et al., 2020a)

Rate optimization by equalizing the arrival times was performed using the SCIPY (library) implementation of the sequential least squares algorithm on Python. The trained model was used to compute the minimum times of flight at each producer.

Optimization without Uncertainty

The workflow was first tested on a single realization (shown in Figure 10 (a)) under the assumption that there is no uncertainty in the underlying geology. The waterflood performance is quantified in terms of the total cumulative oil recovery. In addition, we impose constraints on field-wide injection and production rates at values q_{tot_I} and q_{tot_P}. Hence the optimization problem is posed as described earlier:

$$\min_{[\boldsymbol{q_I}|\boldsymbol{q_P}]} \|\boldsymbol{e}(\boldsymbol{q_I}, \boldsymbol{q_P})\|_2 \tag{16a}$$

$$\boldsymbol{s.t.}\ \sum_{i=1}^{N_I} q_{I_i} = q_{tot_I}\ ;\ \sum_{j=1}^{N_P} q_{P_j} = q_{tot_P} \tag{16b}$$

The base unoptimized case for the experiment is where all the injection rates and production rates are set to be equal. Hence for the unoptimized case, $q_{I1} = 1000\ bbl/d$ and $q_{P1} = q_{P2} = \cdots = q_{P4} = 250\ bbl/d$. These are also the initial guesses to the optimization problem. The values of q_{tot_I} and $q_{tot_P}$are both set to $1000\ bbl/d$, setting the voidage-replacement ratio at unity. It is to be noted that the unit voidage-replacement ratio is not a requirement for optimization.

The resulting oil saturation after running the forward simulator with the assigned optimized rates (Figure 10 (b)) for 20 years is shown in Figure 10 (c). The cumulative oil recovery is also plotted with respect to time for the equal rates and optimized cases. The times of earliest breakthrough for each case is marked as BT1 (equal rates case) and BT2 (optimized case). This clearly shows the delayed breakthrough due to the equalization of arrival times at the producers.

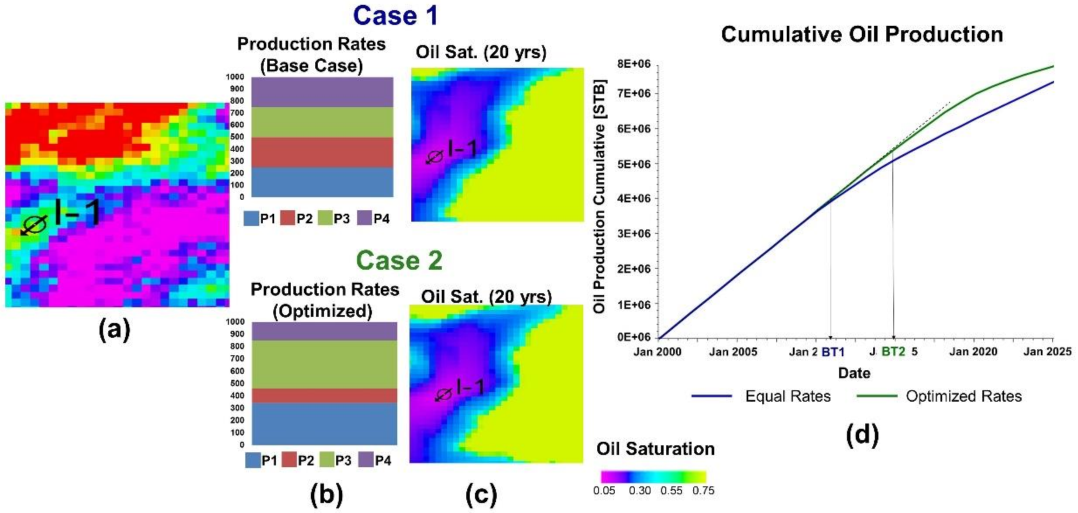


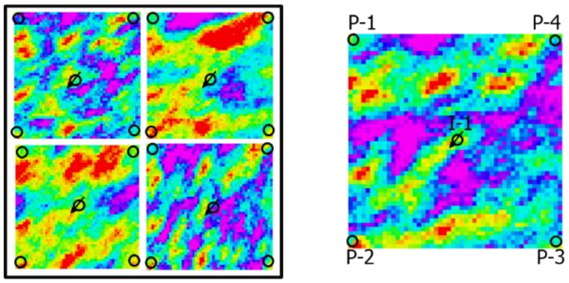
Figure 10 For a single realization (a), considering no uncertainty, we compare the equal rates base (b top) case and the optimized rates (b bottom) case. The oil saturation profiles (c) and cumulative oil production plots (d) show the improvement in the sweep and thereby oil recovery after optimization. The earliest breakthrough time is marked on (d) as BT1 (equal rates case) and BT2 (optimized case) (reprinted with permission from Sen et al., 2020a)

Optimization under Uncertainty

The trained model was then used to obtain a well rate scheme optimized in a probabilistic sense over multiple realizations. We do so by solving the problem posed in eqn 9:

All 600 realizations used in training was used for optimization as well (Figure 11 (a)). At each iteration, a pool of 500 realizations from the training set is chosen with replacement in order to compute the value of the objective function. Doing so ensures a sufficiently large ensemble of residuals such that the probability density function (PDF) of the residuals have converged in distribution (via the central limit theorem) at each iteration. Once convergence is reached, the objective function $\mathbb{E}(\boldsymbol{e}^T\boldsymbol{e})$ is evaluated and the optimization proceeds as per sequential least squares algorithm.

The optimized rates obtained by solving eqn 9 was tested on a new test realization outside the training dataset (Figure 11(b)). For comparison, a blind realization chosen randomly from the training dataset was optimized and these rates were applied on the test realization.



Multiple realizations for optimization
(a)

Test realization
(b)

Figure 11 (a) The objective function for optimization under uncertainty is computed by sampling multiple realizations with replacement from the training pool (b) The obtained optimal well rates were tested on a new test realization outside of the training pool. (reprinted with permission from Sen et al., 2020a)

The cumulative oil production profiles for the equal rates case (Case 1) and the optimization case based on a single blind realization (Case 2) are represented by the blue and red curves in Figure 12. The resulting oil saturation profile from running an extended simulation for 15 years is given in Figure 13 (Case 1 and Case 2). The breakthrough in Case 2 is observed to happen earlier than in Case 1. This highlights the fact that rate optimization without considering geological uncertainty can lead to early breakthrough and poorer oil recovery than an unoptimized case. The cumulative oil recovery plot using well rates obtained by optimizing over multiple realizations is shown by the green curve in Figure 12. The earliest breakthrough is delayed compared to the preceding cases (BT3 in comparison to BT1 and BT2) and the resulting oil saturation profile (Figure 13 (Case 3)) shows more symmetry and increase in sweep.

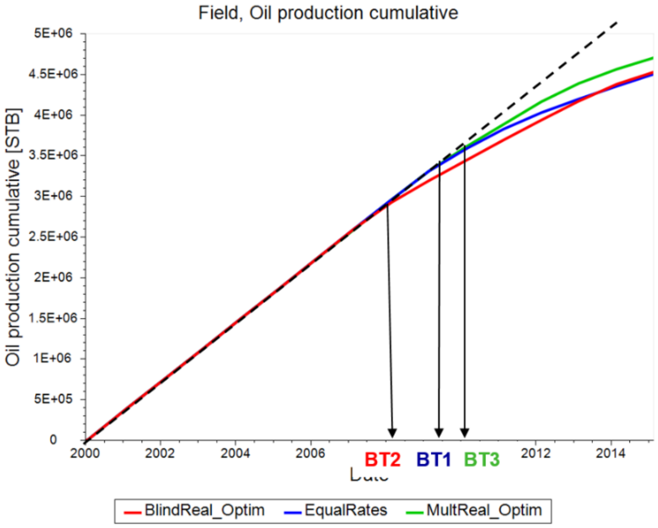


Figure 12 A comparison of cumulative oil production for optimization under uncertainty – equal rates case (blue), blind realization optimization case (red) and multiple realization optimization case (green) (reprinted with permission from Sen et al., 2020a)

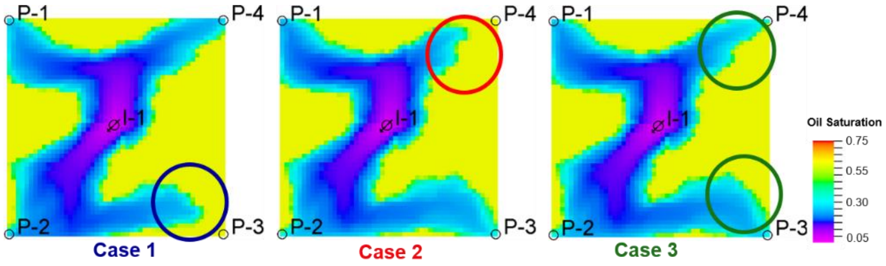


Figure 13 Comparison of oil saturation profiles at the end of 15 years for equal rates case (Case 1), blind realization optimization case (Case 2) and multiple

realization optimization case (Case 3) (reprinted with permission from Sen et al., 2020a)

A bulk of the computational requirement for the workflow is at the training dataset generation stage where the forward simulator has to be run to generate each datapoint (Table 4). However, at the deployment phase, a single forward run of the proxy model is faster than the conventional simulator-based approach by almost three orders of magnitude (Table 5). This makes the proxy model well-suited for multiple realization optimization where the evaluation of the objective function at each iteration requires several forward simulations for a converged result.

Table 4 Computational requirement for training data generation

Task	**CPU Time (s) (2D Case)**
Eclipse run + saving data	6
Streamline tracing + saving	2
Overheads	1
Total	9
Total for 1000 datapoints	**2.5 hrs**

Table 5 Computational requirement at deployment phase

Task	**CPU Time (s) (2D Case)**
Proxy model	0.001
Eclipse	2
Streamline Tracing	0.02

3D Brugge Case

Field Description

The machine-learning based rate optimization workflow was tested on the Brugge benchmarking case described by Peters et al. (2010). The Brugge dataset is a publicly available dataset containing 104 realizations of x-permeability ($PERMX$), z-permeability ($PERMZ$), porosity (ϕ), net to gross ratio (NTG) and initial water saturations ($SWCON$) of a faulted reservoir with 20 producers and 10 injectors. The field described by the model is comprised of a half dome which is elongated east-west and with a large fault at its northern boundary. The field also has a smaller internal fault positioned at an angle of $\sim 20°$ to the northern boundary. The reservoir property fields and zone thicknesses are representative of a North Sea Brent-type field. The size of the reservoir model is $139 \times 48 \times 9$ gridblocks. The reservoir may be divided into roughly 4 zones in the z-direction:

- Zone 1: Layers 1 – 2
- Zone 2: Layers 3 – 4
- Zone 3: Layers 5 – 6
- Zone 4: Layers 7 – 9

For our application, we consider uncertainty in x-permeability, k_X, and porosity, ϕ .Therefore, the NTG, water-oil contact and relative permeability curves are kept constant across all realizations. Furthermore, z-permeability k_Z is computed from a k_X by assuming a constant $\frac{k_Z}{k_X} = 0.1$. Figure 14 shows three of the ϕ and k_X realizations.

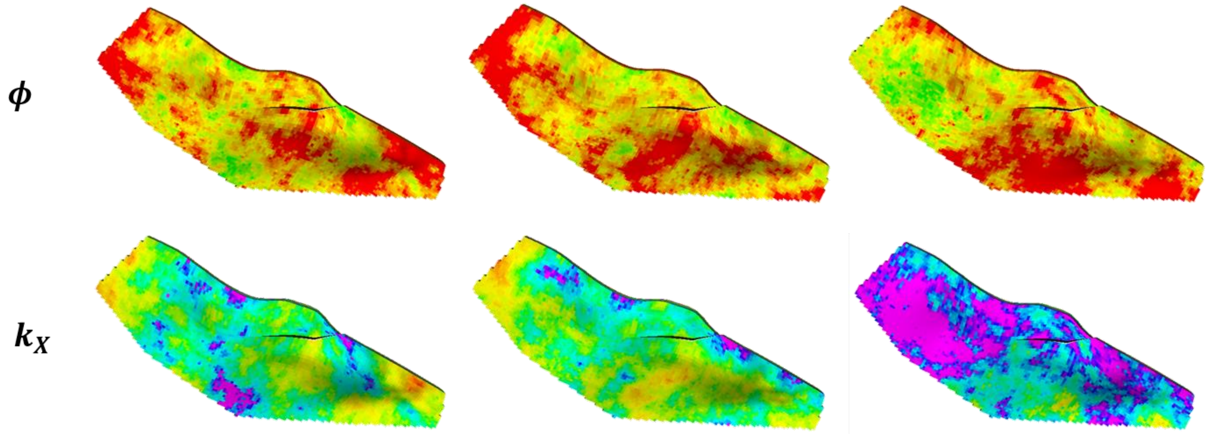


Figure 14 Three of 104 realizations of porosity and permeability available in the Brugge benchmarking dataset (reprinted with permission from Sen et al., 2020a)

Proxy model building

For generating each training datapoint, a realization was sampled from a pool of 104 realizations (available from (Hoffimann)) and well rates were assigned randomly. Flow simulation was then carried out using ECLIPSE and streamline tracing was performed using our in-house post-processing software DESTINY. A total of 1000 datapoints were generated in this way. The detailed process is given below.

GCT transformation

As with the 2D case, the first step of our workflow consists of representing each of the heterogeneous fields as a linear combination of GCT basis functions. Each of the property fields may be expressed as a linear combination of these basis functions and its basis coefficients (which are property-dependent). This concept is illustrated in Figure 15.

In order to better represent the heterogeneity, the basis coefficients of the ensemble mean of the training pool was sorted based on magnitude. The drop in magnitude of the basis coefficients with increasing number of basis functions for porosity and permeability are shown in Figure 16. It is clear from Figure 16 that the drop-off levels off by around 20 basis functions. However, for our purpose, the basis functions corresponding to the highest 30 values of coefficients were chosen for representing each of the individual realizations, in order to ensure that most of the heterogeneity information is captured. The reconstruction of a reference perm field using the first 30 basis functions is shown in Figure 17.

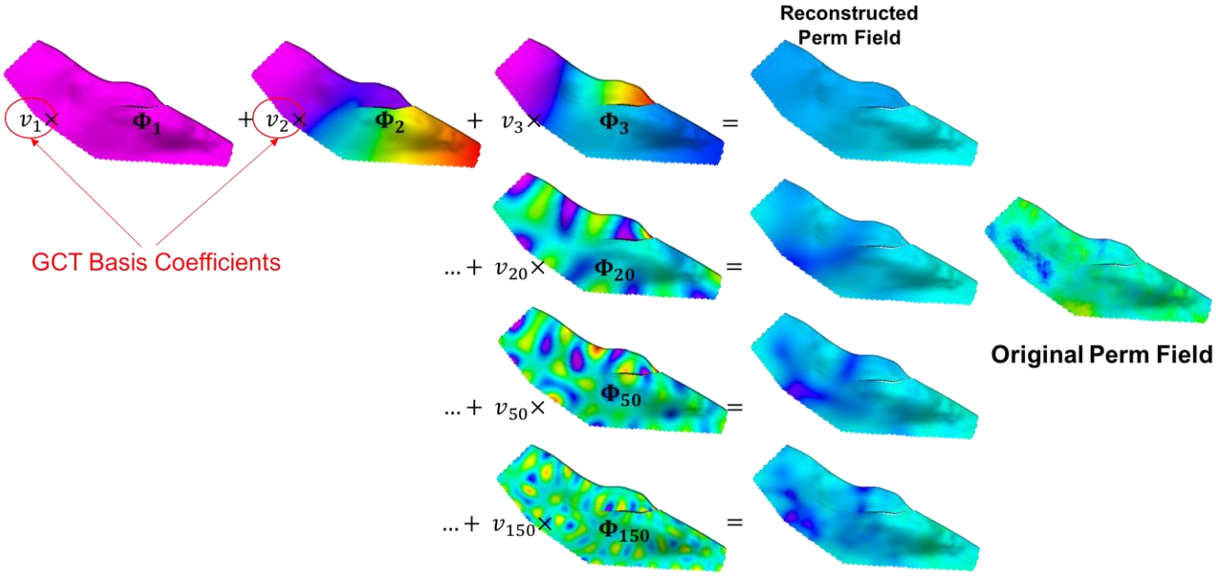


Figure 15 Representation of a k_X realization in terms of GCT basis functions and coefficients. (reprinted with permission from Sen et al., 2020a)

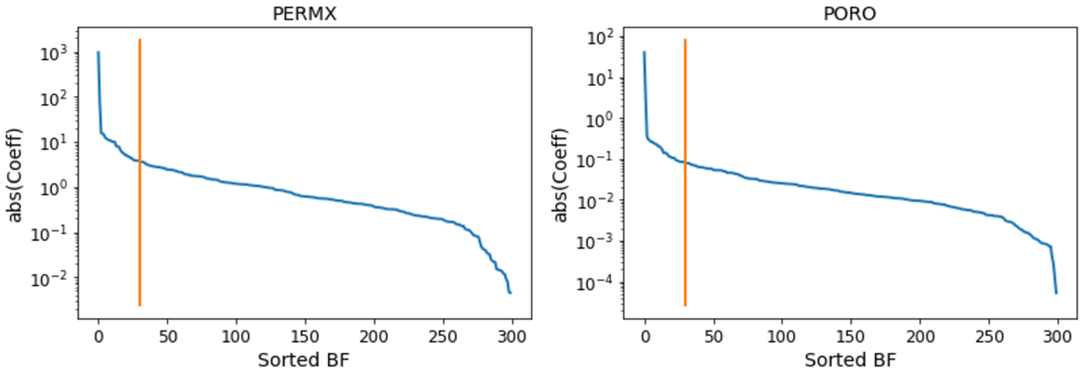


Figure 16 The drop-off in the magnitude of basis coefficients is plotted against the number of basis functions, for permeability and porosity of the Brugge ensemble mean. N_{basis} is chosen to be 30 for both fields (marked by orange line). (reprinted with permission from Sen et al., 2020a)

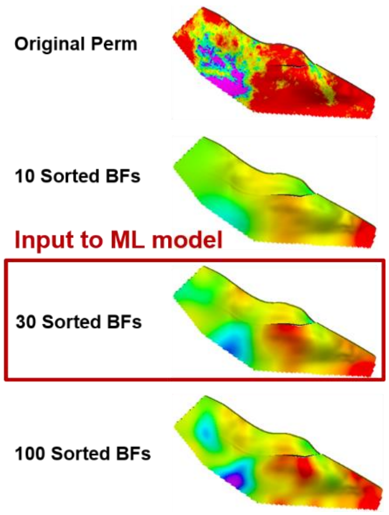
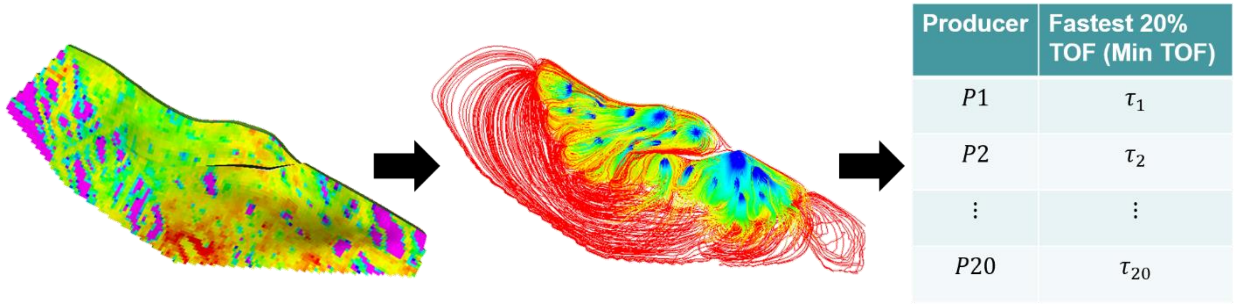


Figure 17 The GCT basis functions are sorted based on the magnitude of their corresponding basis coefficient in the ensemble mean of the training set. Only the first 30 are chosen as input features to the proxy model (reprinted with permission from Sen et al., 2020a)

Streamline tracing

The next step in the workflow is to assign rates (both injection and production) randomly to all the wells. Well rates are sampled from uniform distributions – $U[500,4000]bbl/d$ for producers and $U[1000,8000]bbl/d$ for injectors. The reservoir simulations are performed using ECLIPSE running on Intel® Xeon® CPU with a base processor speed of 2.2 GHz. Once the flow field is computed, DESTINY is used to trace streamlines. DESTINY calculates the TOF for all streamlines that end at each producer. The average time of flight of the fastest 20% of all the streamlines ending at the i^{th} producer is calculated and stored as τ_i. Figure 18 illustrates this step.



Producer	Fastest 20% TOF (Min TOF)
$P1$	τ_1
$P2$	τ_2
$\vdots$	$\vdots$
$P20$	τ_{20}

Figure 18 Illustration of reservoir flow simulation, subsequent streamline tracing and generation of response dataset. (Reprinted with permission from Sen et al., 2020a)

A total of 1000 observations containing the following were generated:

- Basis coefficients for k_X ($\boldsymbol{\nu_{kx}}$)
- Basis coefficients for ϕ ($\boldsymbol{\nu_{\phi}}$)
- Well rates ($\boldsymbol{q_I}, \boldsymbol{q_P}$)
- Times of flights ($\boldsymbol{\tau}$)

Rate allocations for each case, were also computed and saved for running the spectral grouping algorithm.

Spectral Grouping

In order to impose physical constraints on the dependencies between producers and injectors learnt by the model, we treat the problem as a set of N_{group} smaller independent proxy models, each with an associated subgroup of injectors and producer. Therefore, we build N_{group} proxy models $f_1, \dots, f_{N_{group}}$ such that

$$\boldsymbol{\tau_j} = \boldsymbol{f_j}(\boldsymbol{x_j}), \quad \boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{N_{group}} \tag{18}$$

Here, for the j^{th} subgroup G_j, $\boldsymbol{\tau_j} = \left[\tau_{P_{k_P}}\right]$ and $\boldsymbol{x_j} = \left[\boldsymbol{q_{I_j}} \middle| \boldsymbol{q_{P_j}} \middle| \boldsymbol{\nu_{kx}} \middle| \boldsymbol{\nu_{\phi}}\right]^T$ where $\boldsymbol{q_{I_j}} = [q_{I_{k_I}}]$, $\boldsymbol{q_{P_j}} = [q_{P_{k_P}}]$. The subscripts k_P and k_I are such that $k_P = r\ if\ P_r \in G_j\ (1 \leq r \leq N_P)$ and $k_I = s\ if\ I_s \in G_j\ (1 \leq s \leq N_I)$.

The well groupings are formed using spectral clustering, as presented under the section 'Mathematical Formulation'. The average rate allocation R_{avg} was computed by taking the sum of each of the allocated fluid volumes to every producer from every injector and dividing by the sum of injection rates. A representation of the resulting R_{avg} is illustrated in Figure 19. The graph Laplacian matrix for rate allocation was then set up as described before. In our work, k is chosen to be 2, following Alhuthali et al. (2008). The clustering of the first two eigenvectors (corresponding to the smallest two eigenvalues) is performed using k-means clustering. The resulting well grouping is shown in Figure 20.

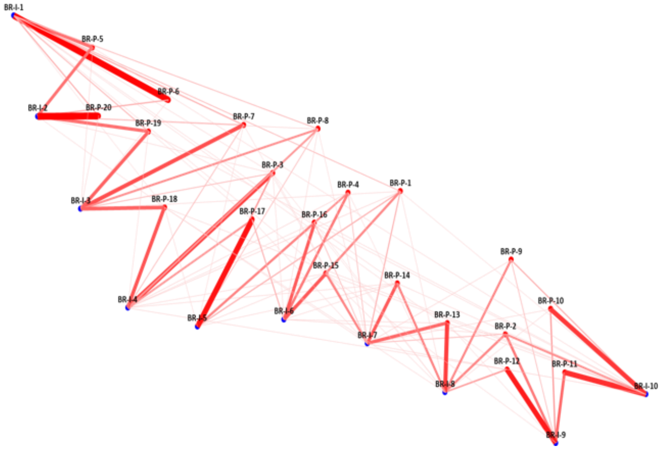


Figure 19 The average rate allocation from the training dataset. These are used to generate well groups by spectral clustering (reprinted with permission from Sen et al., 2020a)

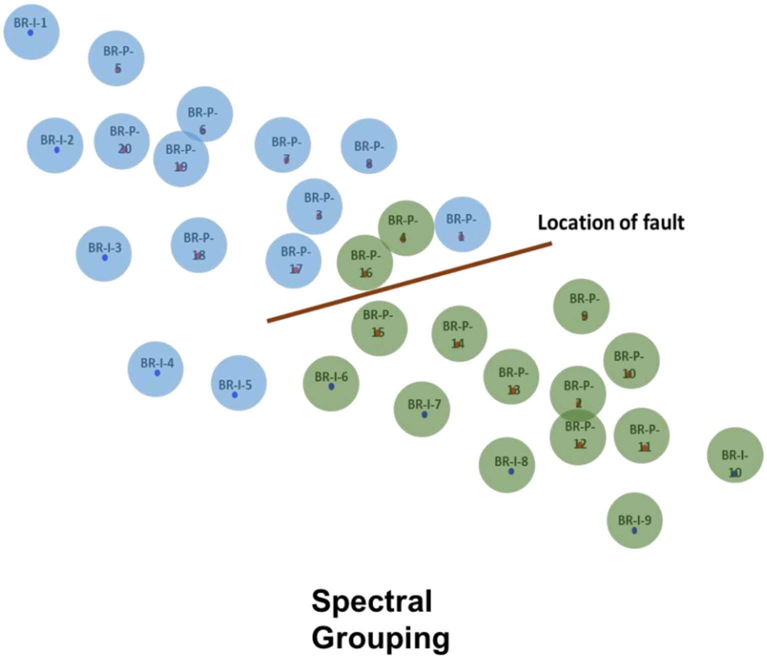


Figure 20 Results of spectral well grouping scheme for $k = 2$ (reprinted with permission from Sen et al., 2020a)

Model Selection using k-fold Cross-validation

5-fold cross-validation was used to choose the best model hyperparameters for the neural network model for each of the two well groups. The average RMSE is shown in Table 6. The neural network architecture chosen is given in . Each network was trained for 500 epochs using the Adam optimizer with a learning rate of 0.001. The batch size was chosen to be 400.

Table 7 Neural network architecture for proxy modeling (Brugge case)

Layer no.	$N_{neurons}$	**Activation**	**Dimensions**	
			Input	**Output**
1 (Input Layer)	0	N/A	N/A	$N_I + N_P + N_{basis} = 75$
2	100	ReLU	75	100
3 (Output Layer)	$N_P = 10$	Linear	100	10 (each group)

Table 6 Model selection using 5-fold cross-validation for the MLP corresponding to each well group. The architecture is represented as a tuple wherein the n^{th} element represents the number of neurons in the n^{th} hidden layer

Group1		**Group 2**	
Architecture	**Total RMSE**	**Architecture**	**Total RMSE**
[10]	1.30	[10]	1.55
[20]	1.23	[20]	1.45
[50]	1.22	[50]	1.38
[100] (Chosen)	1.21	[100] (Chosen)	1.37
[50,50]	1.21	[50,50]	1.37
[100,100]	1.21	[100,100]	1.37

Table 7 Neural network architecture for proxy modeling (Brugge case)

Layer no.	$N_{neurons}$	**Activation**	**Dimensions**	
			Input	**Output**
1 (Input Layer)	0	N/A	N/A	$N_I + N_P + N_{basis} = 75$
2	100	ReLU	75	100
3 (Output Layer)	$N_P = 10$	Linear	100	10 (each group)

The model fit obtained for the chosen model is shown in Figure 21. As explained in the 2D case, the primary reason that we do not obtain an excellent fit (>90%) is the high non-linearity of the predicted TOF with respect to heterogeneity as well as the loss in heterogeneity information due to the reduced order representation. The fit may be improved by increasing the number of basis functions used in the representation. However, the effect of this degradation in fit on the optimization is not significant as seen in Figure 24, wherein we compare the performance of the proxy model based optimization with a full-physics based optimization.

It is evident that the model performance degrades for producers with very high minimum times of flight. These are cases where the producer is practically disconnected from any of the injectors, such that the minimum time of flight recorded corresponds to a few number of highly tortuous streamlines. This degradation in performance barely affects rate optimization since the flux along these bundles are very small.

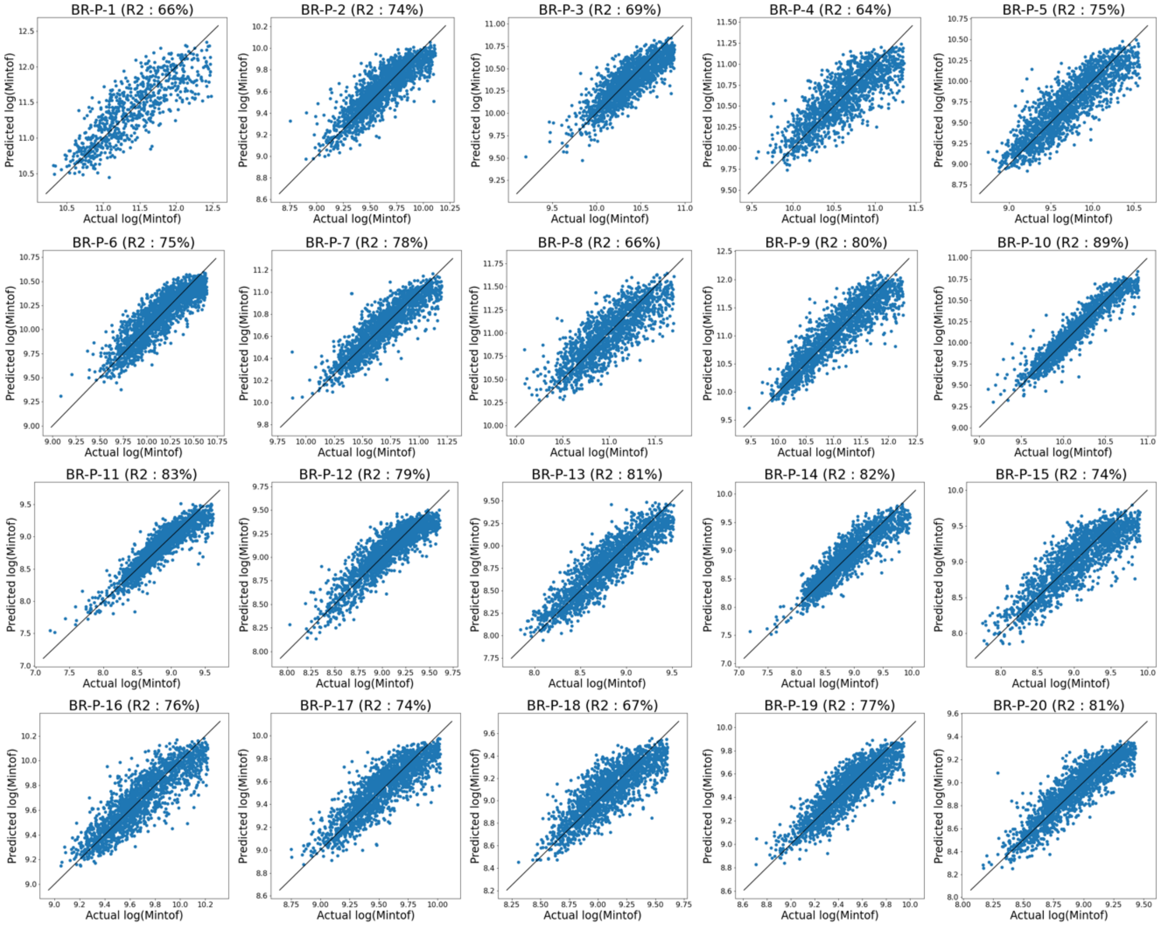

BR-P-1 (R2 : 66%)
BR-P-2 (R2 : 74%)
BR-P-3 (R2 : 69%)
BR-P-4 (R2 : 64%)
BR-P-5 (R2 : 75%)
BR-P-6 (R2 : 75%)
BR-P-7 (R2 : 78%)
BR-P-8 (R2 : 66%)
BR-P-9 (R2 : 80%)
BR-P-10 (R2 : 89%)
BR-P-11 (R2 : 83%)
BR-P-12 (R2 : 79%)
BR-P-13 (R2 : 81%)
BR-P-14 (R2 : 82%)
BR-P-15 (R2 : 74%)
BR-P-16 (R2 : 76%)
BR-P-17 (R2 : 74%)
BR-P-18 (R2 : 67%)
BR-P-19 (R2 : 77%)
BR-P-20 (R2 : 81%)

Figure 21 Model fit from 5-fold cross-validation (reprinted with permission from Sen et al., 2020a)

Rate Optimization

Rate optimization by equalizing the arrival times within each of the subgroups (obtained using spectral clustering) was performed using the ‘Scipy’ (Python library) implementation of the sequential least squares algorithm on Python. The trained model was used to compute the minimum times of flight at each producer.

Optimization without Uncertainty

The workflow was first tested on a single realization, under the assumption that there is no uncertainty in the geology. We impose a unit voidage replacement ratio constraint. Hence the optimization problem is posed as the following:

$$\min_{[\boldsymbol{q_I}|\boldsymbol{q_P}]} \|\boldsymbol{e}(\boldsymbol{q_I}, \boldsymbol{q_P})\|_2 \tag{16a}$$

$$\boldsymbol{s.t.} \ \sum_{i=1}^{N_I} q_{I_i} = q_{tot_I}; \sum_{j=1}^{N_P} q_{P_j} = q_{tot_P} \tag{16b}$$

The base unoptimized case for the experiment is where all the injection rates and production rates are set to be equal. Hence for the unoptimized case, $q_{I1} = q_{I2} = \cdots = q_{I10} = 3000\ bbl/d$ and $q_{P1} = q_{P2} = \cdots = q_{P20} = 1500\ bbl/d$. These are also the initial guesses to the optimization problem. The increased sweep after optimization is evident from the rightmost panel in Figure 22 where the TOF from injector of each gridblock on Layer 1 is plotted. The improvement in oil recovery is evident in a comparison of the oil saturation profiles at the end of 30 years resulting from the unoptimized and optimized cases, as shown in Figure 23. The region in red corresponds to the volume unswept by the flood front. The sweep efficiency is also plotted against time for the unoptimized base case and optimized case in Figure 24.

In order to compare the efficacy of using the proxy model for optimization with full-physics optimization, we performed streamline-based optimization (Chen et al. 2020b) on the same case. These results are shown in Figure 24, wherein we find that the improvement in recovery from using the full-physics method (~10% increment) is not substantial in comparison with the proposed method (~8% increment). Furthermore, the error in our proxy model (with an average R2 score of 76%, as seen in Figure 21) results in a ~2% reduction in cumulative recovery. One could train a more accurate proxy model by increasing the number of data points in the training process.

Furthermore, aerial representations of optimized rate changes resulting from the streamline-based optimization and those from the proposed workflow are given in Figure 25 (a) and (b). It is seen that the proposed workflow correctly identifies the 7 out of 10 wells for which the rates have to be increased (as per the streamline-based schedule). These include the injector I-8, along with producers P-1, P-2, P-8 and P-9 (located in the interior of the edge flood). Furthermore, both optimizations methods allocate reduced rates to the producers that are close to the edge injectors - BR-P-11 to BR-P-19, so as to equalize the arrival times of the flood fronts and thereby increase sweep.

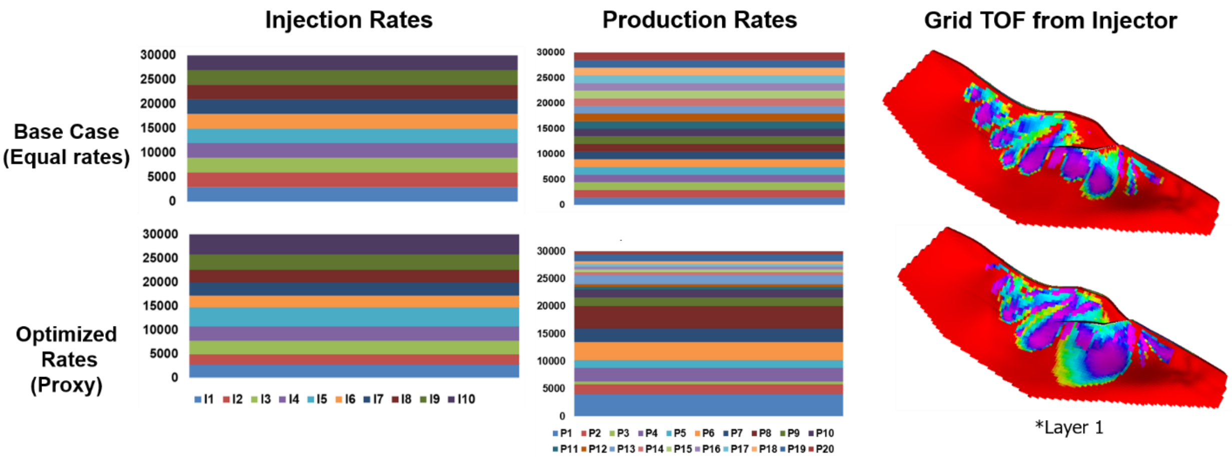


Figure 22 Injection and production rates with sweep visualization in Layer 1 in a Brugge realization before and after optimization. The area in red represents the ‘unswept’ regions in the model. (Reprinted with permission from Sen et al., 2020a)

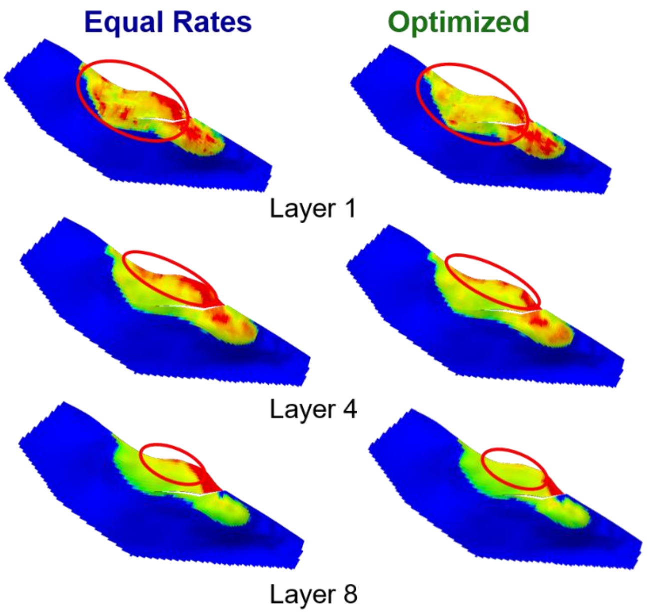


Figure 23 Oil saturation profile at the end of 30 years for the equal rates case compared with the optimized rates case (reprinted with permission from Sen et al., 2020a)

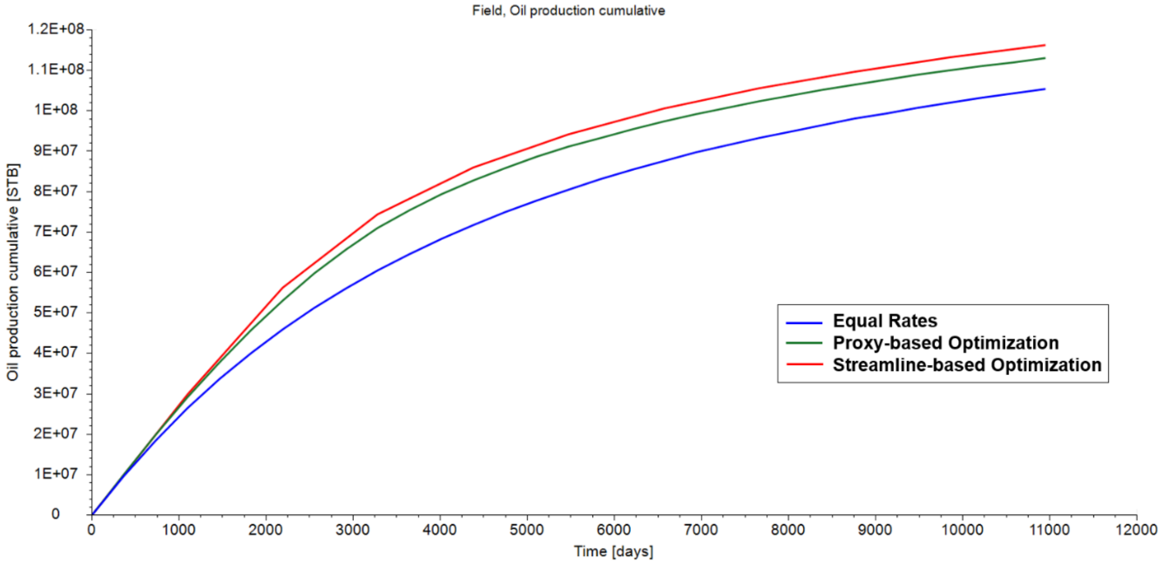


Figure 24 Cumulative oil production comparison in rate optimization based on a single geologic realization: the equal rates (with $q_{I1} = q_{I2} = \cdots = 3000\ bbl/d$ and $q_{P1} = q_{P2} = \cdots = 1500\ bbl/d$), optimal rates based on the proxy model, optimal

rates based on the simulation model (reprinted with permission from Sen et al., 2020a)

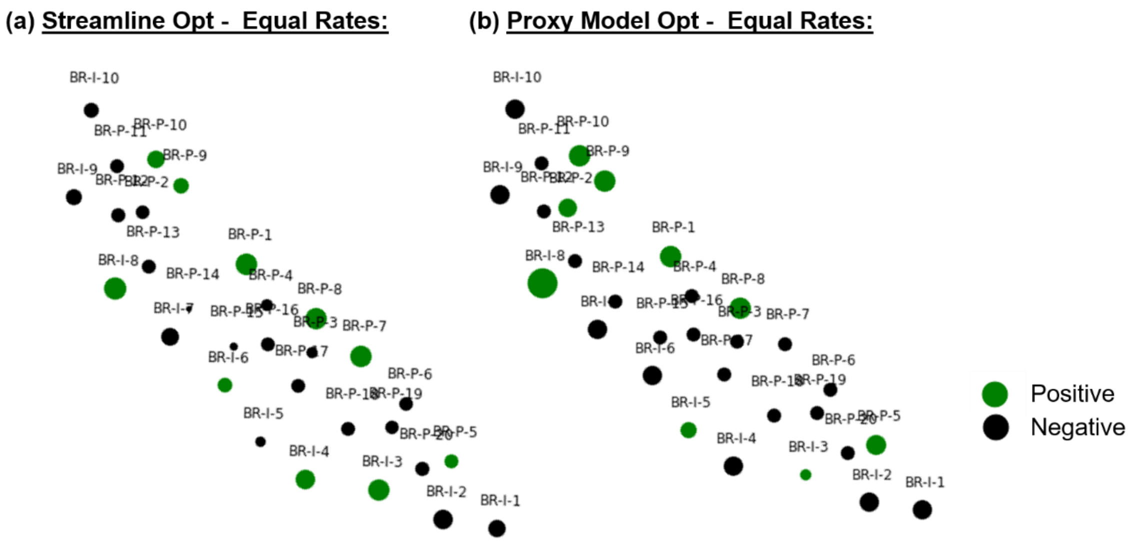


Figure 25 A comparison of rate changes calculated using (a) the streamline-based optimization method and (b) the proposed machine-learning based workflow are shown

Optimization under Uncertainty

The trained model was then used to obtain a well rate scheme optimized in a probabilistic sense over multiple realizations in order to account for uncertainty in geology. We do so by solving the problem posed in Eq 9.

All 36 realizations used in training was used for optimization as well. The optimization pool was made of 200 realizations, which was formed by sampling with replacement from the 36-realization training dataset. Doing so ensures a sufficiently large ensemble of residuals such that the PDF of residuals have converged in distribution (via the central limit theorem) at each iteration. Once convergence is reached, the objective function $\mathbb{E}(\boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e})$ is evaluated and the optimization proceeds as per sequential least squares algorithm.

The optimized rates obtained by solving Eq 9 was tested on a test realization outside the training dataset. The rate changes prescribed by the optimization (i.e. optimized rate minus the initial equal rate) to each well is shown in Figure 26, along with the residual oil at 20 years for each case. From Figure 26 we observe that:

a. The ‘outer’ producers (closest to the edge injectors) are assigned smaller rates whereas the ‘inner’ producers are assigned higher rates as compared to the equal rates base case. This reallocation of rates leads to increased oil recovery from the regions marked by blue ellipses in the oil saturation plot.
b. Along with the inner injectors BR-P-9 and BR-P-10, three injectors (BR-I-7, BR-I-8 and BR-I-10) are assigned higher rates, leading to increased recovery from the region marked with the red ellipse in the oil saturation plot.

In order to better visualize the effect of equalizing time-of-flight, the streamlines at the last timestep from both the unoptimized and optimized case is given in Figure 27. The unoptimized case (left) shows the presence of early breakthrough pathways, indicated by streamlines with small total time-of-flight (in blue), mostly between the edge injectors and the ‘outer’ producers. These have been removed in the optimized case (right) by time-of-flight equalization.

Moreover, there exist regions (marked in red ellipses) with remaining oil that are sparsely covered by streamlines with large total time-of-flight in the unoptimized case. After optimization, the increased rate assignment to injectors BR-I-7, BR-I-8 and BR-I-10, as well as the ‘inner’ producers (especially BR-P-1, BR-P-9 and BR-P-10) ensure that the streamlines are evenly distributed in these regions.

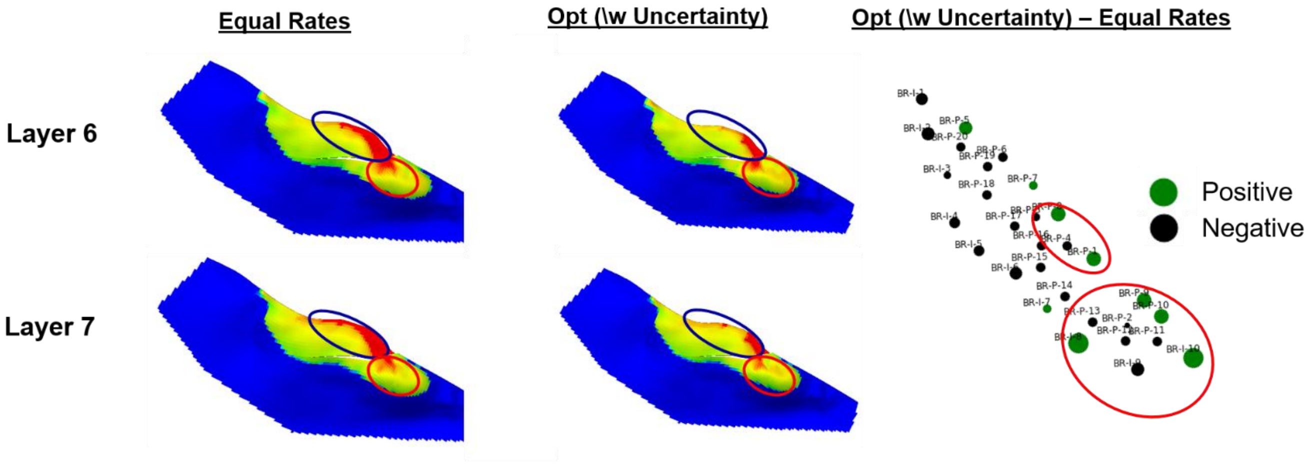


Figure 26 Well rates and remaining oil before and after optimization (reprinted with permission from Sen et al., 2021)

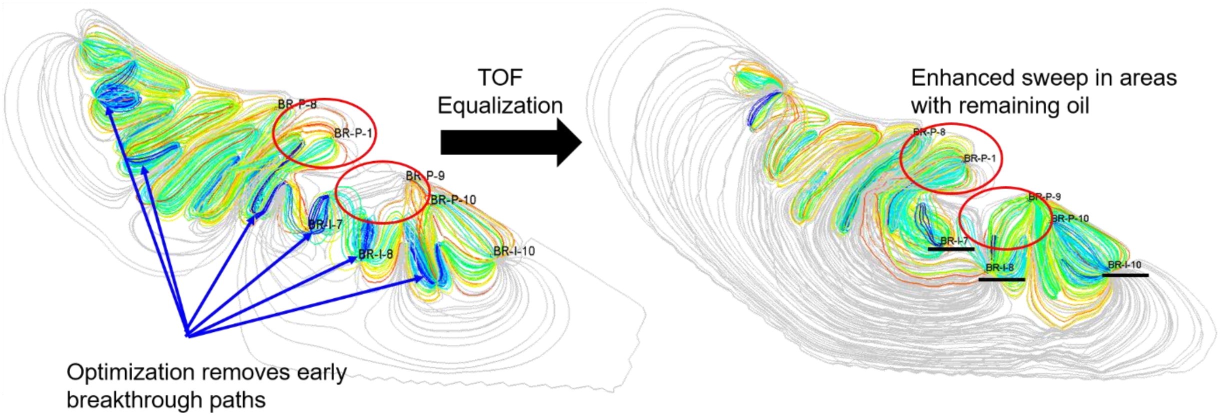


Figure 27 TOF analysis (reprinted with permission from Sen et al., 2020a)

For further comparison, a single realization chosen from the training dataset was optimized and these rates were applied on the test realization. The resulting cumulative oil production optimization under uncertainty is plotted in Figure 28 along with that from optimizing the single realization, as well as the unoptimized case. Both the optimized cases (based on a single realization and multiple realizations) show improved recovery as compared to the unoptimized case, with the latter optimization (considering uncertainty) being marginally better than the former. The benefit of considering uncertainty in the optimization is more apparent in Figure 29, when considering the distribution of cumulative oil recoveries obtained by implementing the optimized rates on multiple realizations in the ensemble. Figure 29 (a) and (b) shows the distribution and expected value of the cumulative oil recovery. The inclusion of uncertainty in the optimization increases the expected cumulative recovery (from the equal rates base case) by ~7%, whereas, without it, the improvement is only ~3%.

This is further illustrated in Figure 30, which shows the resulting oil saturation maps at various layers at the end of the simulation. An improvement in sweep as a result of optimization, in comparison with the unoptimized case, is visible at Layers 1, 6 and 8. Furthermore, the improvement in oil recovery as a result of accounting for multiple realizations, rather than a single realization, is also evident in Figure 28 and Figure 30. The improvement in sweep brought about by multiple realization optimization is evident especially in the oil saturations maps at Layers 1 and 6.

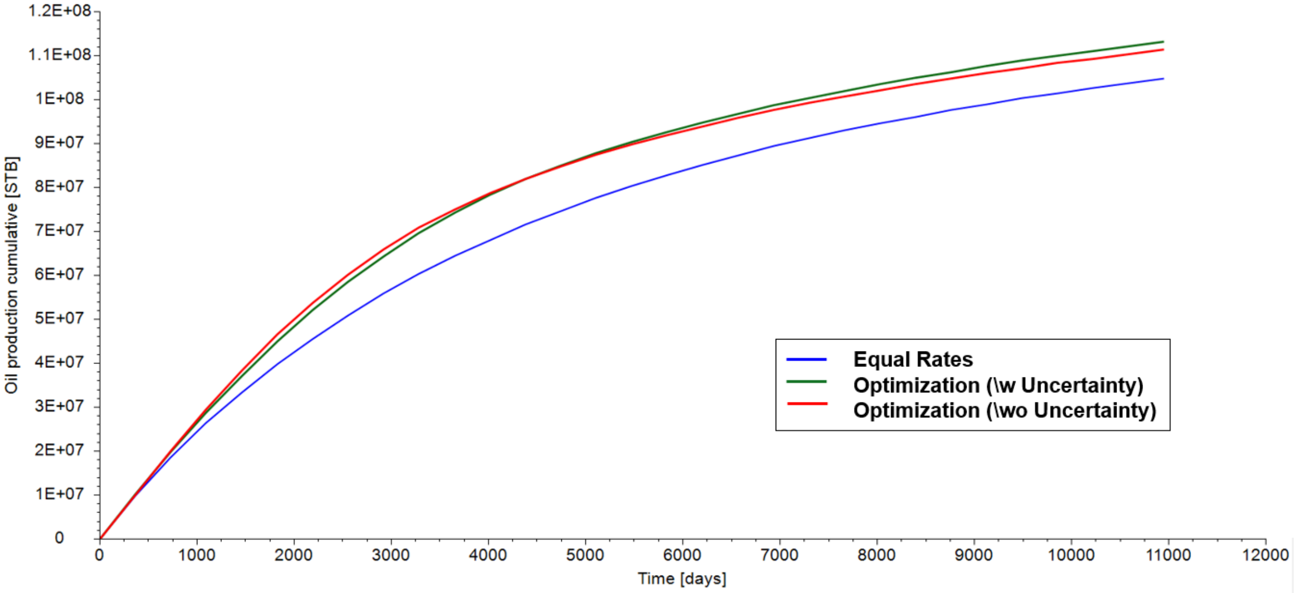


Figure 28 The cumulative oil production in a test case by using the optimization results from a single blind realization compared with that obtained by equalizing arrival time in a probabilistic sense (over multiple realizations) (reprinted with permission from Sen et al., 2020a)

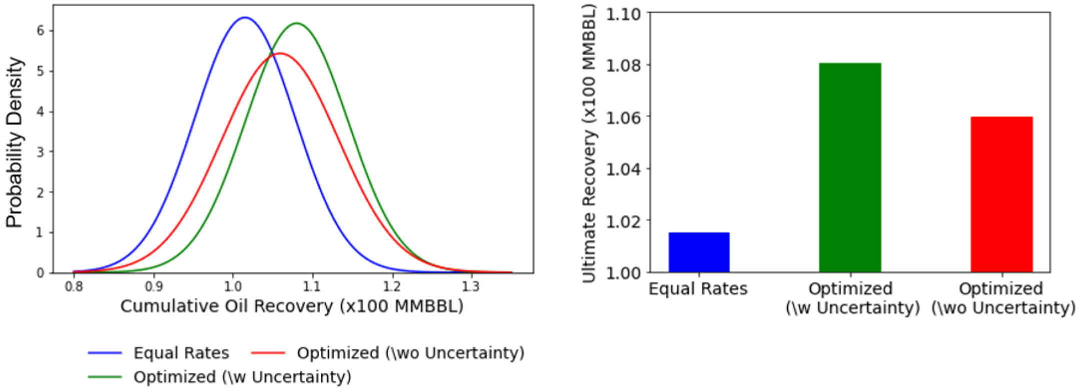


Figure 29 A comparison of the (a) distributions and (b) expected values of cumulative oil recovery resulting from using equals rates, optimized rates (considering uncertainty) and optimized rates (without considering uncertainty). (Reprinted with permission from Sen et al., 2021)

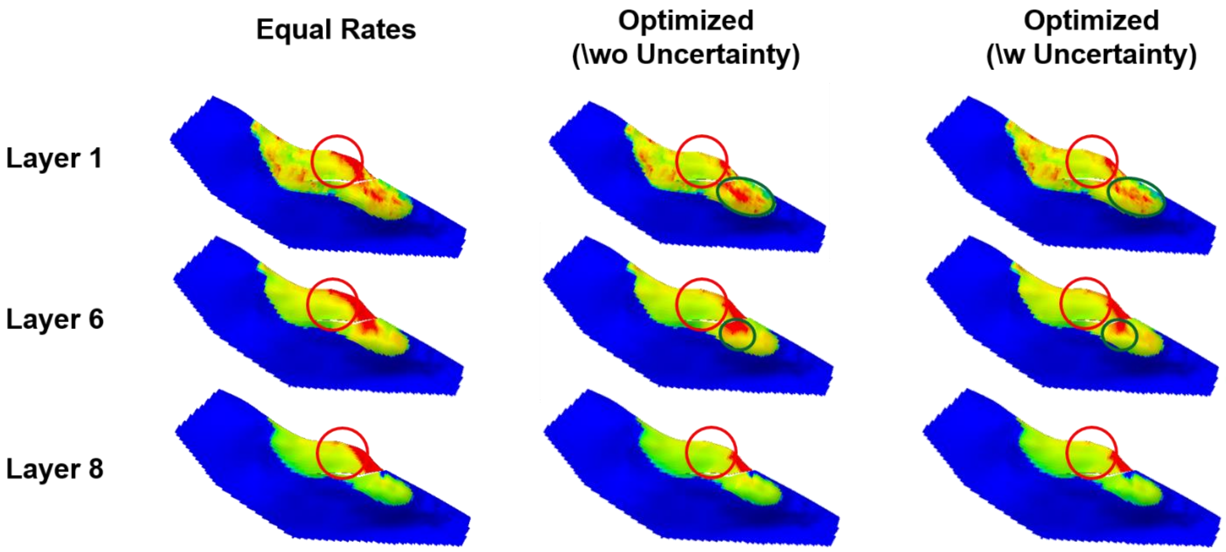


Figure 30 Oil saturation profile at the end of 30 years for the equal rates case compared with the optimized rates case. (Reprinted with permission from Sen et al., 2020a)

Similar to the 2D case, the generation of the training dataset accounts for the bulk of computational requirement of the workflow, with the generation of each datapoint by simulation and streamline tracing taking around 16 seconds on an Intel® Xeon® CPU. However, at the deployment phase, we are able to reduce the CPU time for the forward run from 4 seconds to 0.006 seconds (around 3 orders of magnitude), as shown in Table 8 and Table 9.

Table 8 Computational requirement for training data generation (Brugge Case)

Task	**CPU Time (s)**
Eclipse run + saving data	9
Streamline tracing + saving	5
Overheads	1
Total	16
Total for 1000 datapoints	**4.4 hrs**

Table 9 Computational requirement at deployment phase (Brugge Case)

Task	CPU Time (s)
Proxy model	0.006
Eclipse	4
Streamline Tracing	0.3

Conclusions

We developed and implemented a machine-learning based workflow for rate optimization for two-phase flow in geologic heterogeneous media. A key differentiating aspect of our workflow is the use of a proxy model in predicting a physics-informed intermediate output, i.e., the minimum TOF at the producer, as a measure of inter-well connectivity. Because there is considerable uncertainty in the underlying geology of a field under waterflood, we demonstrated the need for accounting for multiple geologic realizations while performing waterflood optimization. However, the optimization over multiple realizations is computationally intensive since it requires several forward runs of the reservoir simulator at each iteration. Our proposed workflow overcomes this limitation by mapping the minimum TOF to the well rates heterogeneity using a proxy model, thereby speeding up the forward run by three orders of magnitude. This speed-up enables us to perform the optimization by equalizing the minimum TOF considering multiple realizations. However, the application of the proposed methodology to problems that require streamline regeneration, such as varying well rates/constraints or addition of offset wells, would require specification of more inputs to the proxy model such as time or well locations/configurations along with well rates. The extension of this methodology to such problems shall be explored in the next stage of the study.

The key contributions of this chapter are listed below:

- The heterogeneous field properties were projected on a lower-dimensional space using grid-connectivity transformation and expressed in terms of the basis coefficients. The TOF at the producers were mapped to the well rates and the basis coefficients using a multilayered perceptron-based proxy model.
- The proxy model was used to estimate the TOF without having to solve the flow equation and trace streamlines. Fieldwide rate optimization was performed by equalizing the estimated TOF within the well subgroups.
- The proxy model incorporates the effects of heterogeneity on a physics-informed characterization of interwell connectivity (minimum TOF at producer). Furthermore, the forward run of the proxy model is faster than the conventional simulator by three orders of magnitude, which makes this suitable for optimization over multiple geologic realizations.
- The proxy model was used to perform rate optimization over multiple geologic realizations on a 2D heterogeneous case. The results show superior oil recoveries to the cases that were unoptimized or optimized over a single realization.
- The practical feasibility of the approach was demonstrated through application to the SPE benchmark Brugge field case. A systematic grouping of the wells and application of the algorithm to the well groups resulted in improved oil recovery compared to cases that were unoptimized or optimized over a single realization.

CHAPTER III

INTER-WELL CONNECTIVITY DETECTION IN CO2 WAG PROJECTS USING STATISTICAL RECURRENT UNIT MODELS[3]

Reliable quantification of well connectivity is a crucial aspect to forming a good understanding of a reservoir, which in turn helps in formulating future development plans such as rate optimization and offset wells (Chen et al. 2020b). This assumes an even greater importance when applied to high-capital projects such as CO_2 EOR and polymer floods.

Conventional methods for assessing well connectivity include tracer tests (Zhang et al. 2016, Suarsana and Badril 2011) and numerical simulation-based techniques such as streamlines (Datta-Gupta and King 2007). Streamlines explicitly reveal the flow paths in the reservoir and quantify the interaction between injectors and producers via well-pair flux allocations. Extensive literature has shown the capacity of streamlines over a variety of grid systems, including Cartesian grids (Pollock 1988), corner point grids (Cordes and Kinzelbach 1992; Jimenez et al. 2010), unstructured grids (Prevost et al. 2002; Rasmussen 2010; Zhang et al. 2011; Zuo et al. 2021) and dual porosity dual permeability models (Chen et al. 2020a).

---

[3] Part of this chapter is reprinted with permission from Chen, Hongquan, et al. "Model-Free Assessment of Inter-Well Connectivity in CO2 WAG Projects Using Statistical Recurrent Unit Models." SPE Annual Technical Conference and Exhibition. OnePetro, 2021. Copyright 2021 Society of Petroleum Engineers. Further reproduction is prohibited without permission.

However, these conventional methods tend to be either computation-intensive (i.e. numerical simulation) or resource-intensive (such as tracer tests). In recent years, data-driven methods that derive valuable insights from commonly recorded field data have proved to be quite successful in applications in the oil and gas domain related to drilling (Veettil and Clark 2020, Jahani et al. 2021), petrophysics (Sen, Ong, et al. 2020b), reservoir engineering (Yousef et al. 2006, Tian and Horne 2019, Sen, Chen, et al. 2020a and 2021, Liu et al. 2019) and production (Zhou et al. 2018, Pan et al. 2021).

One of the earliest instances of using injection and production data to infer well connectivity was introduced by Heffer et al. (1997) wherein a Spearman rank correlation based approach was used to identify preferential flow paths between injectors and producers. Tian and Horne (2016) introduced a modification to the Pearson's correlation coefficient (Benesty et al. 2009) to identify injection and production rates that vary concurrently. The modified Pearson's correlation coefficient cannot be used for prediction in a stand-alone fashion without being coupled with another predictive model. Albertoni et al. (2003) applied multivariate linear regression to injection and production data to infer inter-well connectivity. This approach had the added benefit of being a predictive tool that estimates total liquid production at a producer, given water injection rates. However, the authors did not extend this method to problems with varying GOR. Yousef et al. (2006) introduced a capacitance-resistance model (CRM) to include the effects of both flow rates and BHP in a similar framework to develop a predictive tool and infer well connectivity purely from data. However, the authors outline a number of pre-requisites for applying the CRM model, such as the absence of long producer shut-ins and constant GOR. These shortcomings may limit the use of CRM to a real-field WAG injection case. However, a number of modifications to the CRM have been developed by various authors, each for specific cases (Sayarpour 2008, de Holanda et al. 2018, Holanda et al. 2018).

Major limitation of the above-mentioned works is the inability to model general non-linearity in production trends arising from factors such as varying GOR and frequent shut-ins. A potential way to redress this shortcoming is by using a universal approximator like neural networks that can theoretically model any measureable function to an arbitrary degree of accuracy (Hornik et al. 1989). An extension of the neural network to sequential data, called the recurrent neural network (RNN), was first introduced by Hopfield (1982) and later by Rumelhart et al. (1986). From thereon, RNNs have been applied to various applications ranging from handwriting and speech recognition (Graves et al. 2008, Dutta and Sarma 2012) to regression applications such as trade forecasting (Dunis and Huang 2002) and hydrological forecasting (Coulibaly and Baldwin 2005). In the past, RNNs have been applied successfully to oil and gas problems such as formation top detection (Sen, Ong, et al. 2020), well-control optimization (Kim and Durlofsky 2021) and production prediction (Tian and Horne 2017 and 2019, Bao et al. 2020). However, connectivity detection based on RNN models is not yet demonstrated.

The statistical recurrent unit (SRU) was introduced by Oliva et al. (2017) as an un-gated alternative to more complex RNN architectures such as LSTMs (Hochreiter and Schmidhuber 1997) and GRUs (Chung et al. 2015). The SRU was shown to capture long term dependencies at least as much as the latter models, simply by using moving averages of temporal information at different scales. Therefore, the SRU architecture is simpler and more interpretable compared to LSTMs and GRUs, enabling easier training and incorporation of additional constraints such as well locations.

Variable importance is an assessment of the contribution of each input on the output (Breiman 2001). It can be a promising method of inferring well connectivity from a data-driven model. That is, the variable importance computed from an explainable data-driven model, should ideally correspond to the actual influence that an input signal (such as those from an injector) has on the output (production at producer). In this way, an explainable and predictive data-driven model may be used to derive insights on the actual reservoir connectivity in a quick and efficient way, without the need to run computationally costly reservoir simulations.

With the above in mind, we propose a SRU based framework for inferring inter-well connectivities. The proposed method contains two key components: SRU model specifically framed to the CO_2 WAG problem and the inter-well connectivity assessment based on the SRU model and variable importance calculation. The rest of this chapter is organized as follows. First, the proposed methodology is explained in detail, focusing on the overall workflow - the statistical recurrent unit modeling and the variable importance quantification. Second, we apply the proposed framework to the CO_2 WAG problem with selected inputs and customized loss function. Third, the specified workflow is tested using a field-scale CO_2-WAG injection case. Finally, we summarize our findings and explore avenues for future research.

Methodology

In this work, we develop an SRU-based framework, for estimating connectivity between injectors and producers using the time-elapsed well measurements. Furthermore, we demonstrate the validity of the inferences by comparing with a physics-based measure of connectivity – the streamline flux allocations.

Connectivity Inference Workflow

The general workflow for connectivity inference is shown in Figure 31. The first step (Figure 31(a)) is to fit a machine learning model to the data. In our application, we train an SRU model for each producer that predicts its gas production rate given the producing pressure and injector-wise injection rates. The next step (Figure 31(b)) is to ensure the reliability of the proxy model by testing its performance on a dataset that has not been seen during training (test set). A model with good predictive power on the test set is an indication that the connections that it learned during training are representative of the true connections that exist in the field. Once we have a reliable model, we can quantify the influence of any injector on the producer by computing the permutation variable importance (Breiman 2001) which quantifies the contribution of injector's rates in the accuracy of prediction of producer's gas production. In this way, a variable importance map can be plotted showing the major connections in the field (Figure 31(c)).

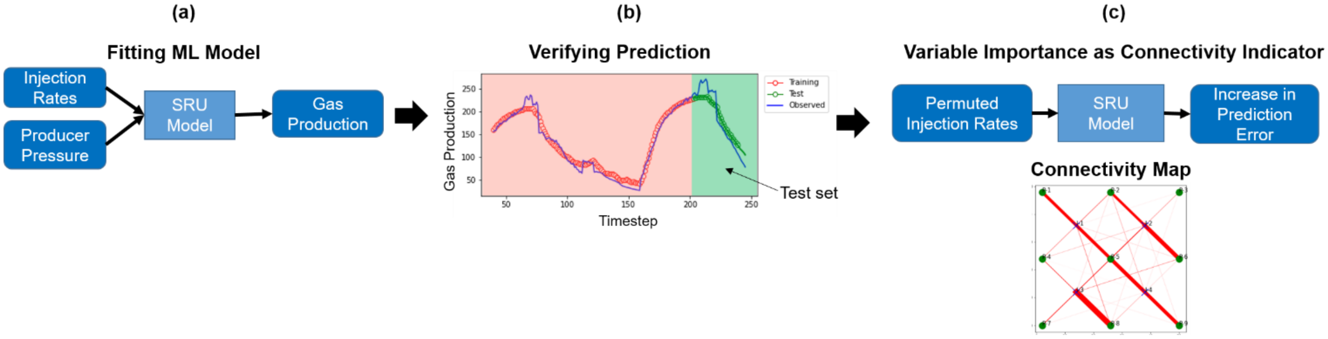


Figure 31 Connectivity inference workflow using machine learning (reprinted with permission from Chen et al., 2021)

The mathematical notations are introduced here for the ease of reading of subsequent sections. The dataset used for modeling a particular producer consists of the input matrix $X_{T\times M}$ and the output matrix $Y_{T\times 1}$. T is the number of time steps, M is the number of features (predictor/independent variables) which can be gas or water injection rate of selected injectors or pressure of the producer. When X is viewed as time series, $X_{T\times M} = [x_1, x_2, \dots, x_T]^T$, where $x_t = [z_1(t), z_2(t), \dots, z_M(t)]$ is a row vector ($1<t<T$) and $z_i(t)$ represents the *i-th* feature value at step *t*. When X is viewed as a set of features, $X_{T\times M} = [z_1, z_2, \dots, z_M]$, where $z_i = [z_i(1), z_i(2), \dots, z_i(T)]^T$ is a column vector ($1<i<M$). These two perspectives will be employed to explain SRU and permutation variable importance in the following sections. $Y_{T\times 1} = [y_1, y_2, \dots, y_T]^T$, y_t is the gas production rate of the producer at step *t* in current study.

Statistical Recurrent Network

A commonly used approach for modeling time-series data consists of computing the output as a function of the current information of the system, commonly called the ‘state’. Such an approach to time-series model is employed in various algorithms ranging from hidden Markov models (Sen et al. 2014) to different variants of RNN (Sherstinsky 2020, Hochreiter and Schmidhuber 1997, Chung et al. 2015). Broadly speaking, the RNN predicts the output at a time step as a function of the input at the current time step as well as the information passed on from the previous time step. Regardless of architectural variations, the RNN (Sherstinsky 2020) has two sets of inputs:

- Inputs pertaining to the current time step
- Information passed from the previous time step(s)

An RNN to predict production rate is shown in Figure 32, illustrating the general concept where the output at *t* is computed from the inputs at *t* as well as the recurrence, which in this case, is merely the output at *t*-1. In Figure 32, a weighted sum of the inputs from the current time step t $\left(z_1(t), \ldots, z_M(t)\right)$ and the output at $t-1$ (i.e. y_{t-1}) pass through an activation function (denoted σ) to produce the output y_t.

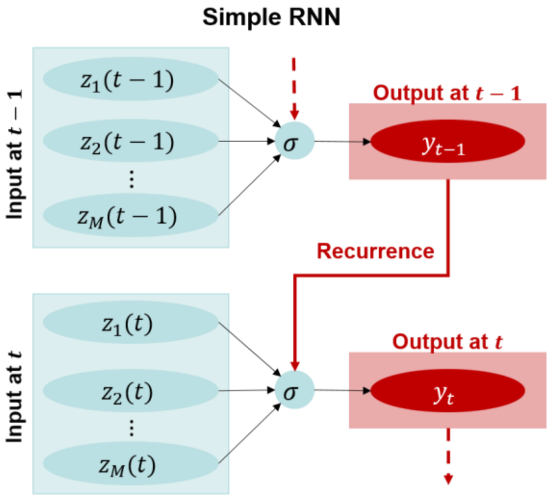


Figure 32 A schematic representation of a RNN to predict the inputs z_1,…z_M. The output at t is computed as a function of inputs at t and the previous output at t-1. This type of recurrence enables the RNN cell to compute the output as a function of the history. (Reprinted with permission from Chen et al., 2021)

The SRU used for our problem is based on the work by Oliva et al. 2017. This architecture adds a layer of complexity to the basic RNN (Figure 32) by introducing a mechanism to keep track of short term and long term dependencies between input and output by means of exponential moving averages. Given an input sequence $\{x_1, x_2, \dots, x_T\}$ and output sequence $\{y_1, y_2, \dots, y_T\}$, the SRU first computes an instantaneous state γ_t as a function of the input, as given below:

$$\gamma_t = f(x_t) \tag{1}$$

where $f(x_t)$ represents a neural network.

Subsequently, a cumulative state μ_t may be computed as the exponential moving average at different scales ($\alpha_s \in \{\alpha_1, \alpha_2 \dots\}$) as given below.

$$\mu_{t,\alpha_s} = (1 - \alpha_s)\gamma_t + \alpha_s \mu_{t-1,\alpha_s} \tag{2}$$

A schematic representation of the calculations involved are shown in Figure 33. Consider a time-series problem illustrated in Figure 33(a) with the input time-series $\{x_1, \dots, x_T\}$ and a corresponding time-series of the dependent variable $\{y_1, \dots, y_T\}$, which are to be predicted. Assuming a window size $N_w = 3$, the first prediction would be made at $t = 3$. The instantaneous states $\gamma_1, \gamma_2, \gamma_3$ are computed from the respective inputs x_1, x_2 and x_3 by passing these through a neural network NN_1. Subsequently, the cumulative states at each scale (number of scales is 2 in the illustration) is computed for $t = 1,2,3$ as in Eq. 2 and Figure 33(c), from the computed γ's. This computation may be represented as a weighted average of the γ's as shown in Figure 33(b). Ultimately, the output y_3 is computed by passing the cumulative states μ's through a neural network NN_2.

In this study, NN_1 and NN_2 are dense neural networks with rectified linear activation function (ReLU), which is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

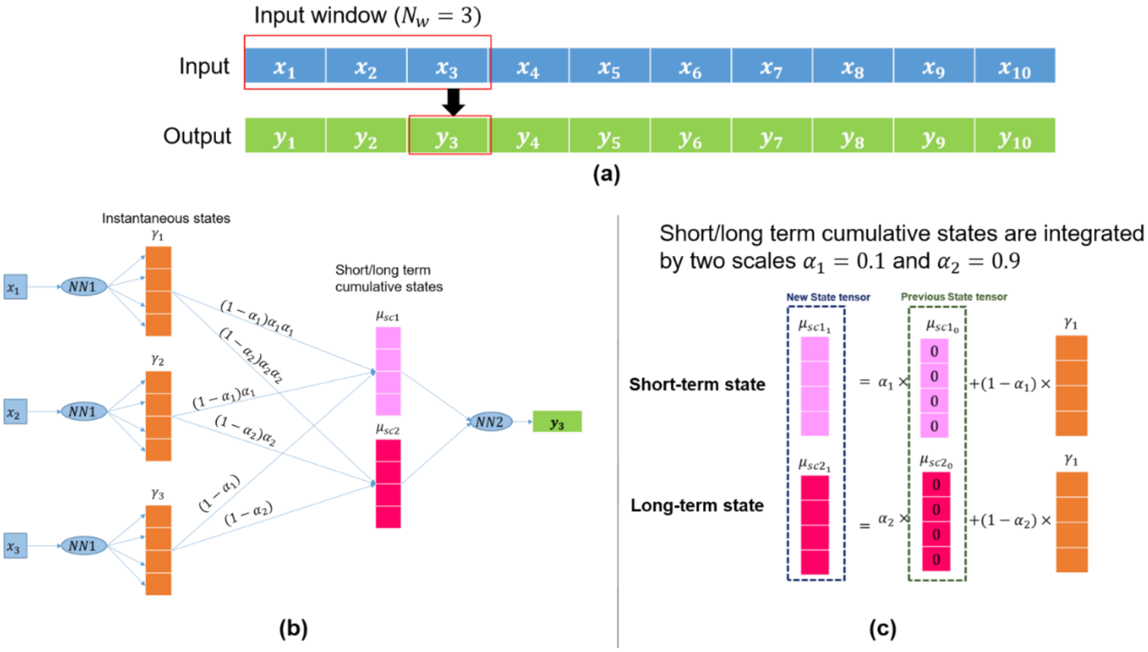


Figure 33 Detailed illustration of the SRU architecture (2 scales in the illustration) as applied to a time-series problem described in (a) with inputs $\{x_1, \dots x_{10}\}$ and predicted variable $\{y_1, \dots y_3\}$. (b) The instantaneous states $(\gamma_1, \gamma_2, \gamma_3)$ are computed from the inputs at each time step in the window, x1, x2 and x3. Next, the short term (μ_{sc_1}) and a long term (μ_{sc_2}) cumulative states are computed from the instantaneous states, capturing the short-term and long-term information. The output at t=3 is then a function of μ_{sc_1} and μ_{sc_2}. (c) The computation of μ_{sc_1} and μ_{sc_2} is performed recurrently, as an exponential moving average of the instantaneous state at t and the cumulative state at t-1. (reprinted with permission from Chen et al., 2021).

Permutation Variable Importance

Variable importance (VI) refers to any metric that quantifies the contribution of any feature z_i in the accuracy of prediction of Y. A wide variety of VI measures have been reported in literature. Commonly used VI metrics include those based on assumed linear relationships (Achen 1982, Bring 1996) and others that employ some kind of decomposition of the covariance matrix (Kruskal 1987, Budescu 1993, Zuber et al. 2011). A comprehensive study of various importance measures and its application was provided by Grömping (2015).

In the present work, we adopt a machine-learning specific VI scheme, first introduced by Breiman (2001) (as an application to random forest models). Herein, VI of a feature z_i is measured in terms of the increase in prediction error from using a random permutation of the rows of z_i. Therefore, given a dataset $[Y, X]$ where $X = [z_1, z_2, \dots, z_M]$, we first compute the original prediction error (loss). In this work, we define the loss as the mean squared error over the dataset:

$$e_{orig} = loss(Y, X) = Mean_t\left(\left(y_t - y_{pred,t}\right)^2\right) \tag{3}$$

Next, we randomly permute the rows of the feature z_i, to get z_i^{shuf}. Here, the superscript $shuf$ represents a single permuted ordering of the set of row indices $\{1,2, \dots, T\}$. The perturbed prediction error is then given as

$$e_{shuf,i} = loss(Y, X_{shuf,i}) \tag{4}$$

Where $X_{shuf,i} = [z_1, z_2, \dots, z_i^{shuf}, \dots, z_M]$. The shuffling and e_{shuf} computation may be done repeatedly to get the expected value of e_{shuf} over the permutations of z_i. The variable importance of z_i to Y is then computed as

$$VI_i = E\left(e_{shuf,i}\right) - e_{orig} \tag{5}$$

An illustration of the permutation variable importance is shown in Figure 34. Consider a dataset, *X* being gas injection rates at injectors I_1 and I_2 and pressure data at producer P_1, *Y* being the gas production rate at producer P_1. The prediction error may be computed as the mean squared difference between the observed values and the corresponding model predictions. In order to compute the variable importance of injector I_1 to P_1, we permute the values of input under I_1 and then re-predict using the model. The shuffling leads to an increase in the resulting prediction error. The new error is e_{shuf}. By repeatedly shuffling the columns of I_1 and computing e_{shuf}, we obtain $E\left(e_{shuf}\right)$. Now the variable importance of I_1 is the difference between $E\left(e_{shuf}\right)$ and e_{orig}. Higher this difference, greater is the importance of I_1 to P_1.

Since the predictions for different producers are made via different SRU models (with separate sets of NN weights), a comparison between VI's with different producers (e.g. $I_1 - P_1$ vs $I_1 - P_2$, or $I_1 - P_1$ vs $I_2 - P_2$) is inherently meaningless. In other words, the relative strength of connections inferred via the proposed method is meaningful only when these inferences are made based on the same SRU model. Therefore, once we obtain $E\left(e_{shuf}\right)$ values for all injectors contributing to the same producer P_1, we normalize these values with respect to their maximum value. That is, the strongest connection to P_1 will have a normalized VI of 1.0.

$$connectivity_{VI_i} = \frac{VI_i}{VI_{max}} \tag{6}$$

This kind of producer-wise normalization of VI's is performed for all producers in the model. The resulting VI map shows the dominant connections with respect to each producer.

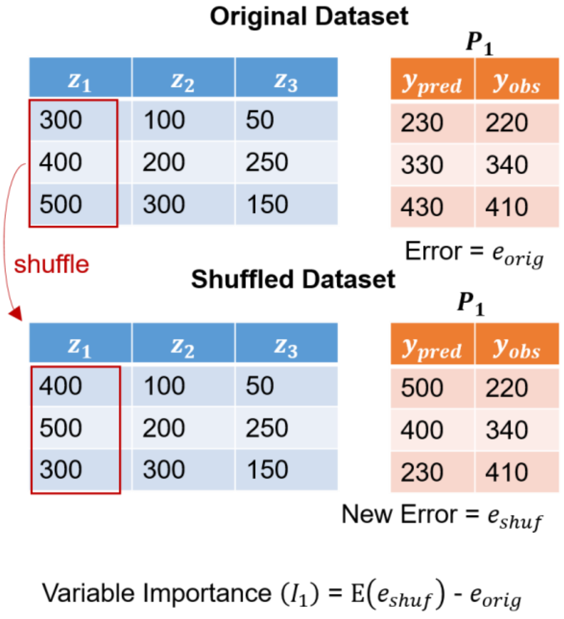


Figure 34 An illustration of permutation variable importance computation. The features z_1, z_2 and z_3 denote the gas injection rates at I_1 *and* I_2 and pressure at P_1 respectively. The predicted and observed gas production at P_1 is given by y_{pred} and y_{obs}. The variable importance of I_1 to P_1 is computed by shuffling z_1 and recording the expected increase in error in y_{pred}. (Reprinted with permission from Chen et al., 2021)

Framework Specification for CO_2 WAG Problem

The Methodology section has already established the SRU framework for general time series problems. This section covers the application designs specific to typical CO_2 WAG problems.

The features selected as input to a machine learning model can significantly affect the performance of machine learning applications. The omission of dominant features may prevent the model from capturing the true behavior of the system. Moreover, given a limited amount of data, addition of non-dominant features may cause overfitting (Hawkins 2004). To address this issue, a problem-specific tailoring of the input may reduce the chance of unreasonable model coefficients and thus improve the prediction accuracy. Another way to prevent overfitting of a machine learning model would be the technique of regularization, where a penalty term is added to the loss function to control the excessively fluctuating function so that the coefficients don’t take extreme values.

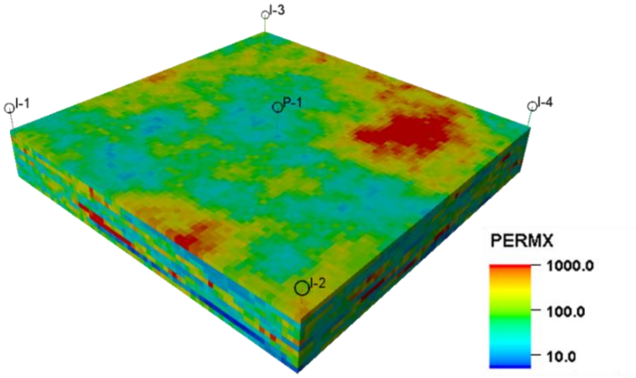
To adapt the SRU framework for successful application to the CO_2 WAG problem, a series of customizations are introduced. These include the selection of input features and regularization using the well location information. The customized application of SRU is demonstrated on synthetic model simulations under typical CO_2 WAG operations as the following. The producers are under stable pressure constraints though limited pressure variations are imposed. A typical CO_2 formation volume factor of 1.0 RB/MSCF (0.0056 cf/scf), reported by Cherdasa et al. 2018, is used in our model. The CO_2 WAG ratio of each injector is sampled between 0.25 and 0.5, referring to the work by Olalotiti-Lawal, Onishi, Datta-Gupta, et al. (2019) and Ren and Duncan (2019). The customizations are decided based on the prediction accuracy on the test dataset and the connectivity accuracy as per the streamline method.

The first subsection describes a technique to understand the effects of various input features such as gas injection rates and water injection rates. Moreover, we shall describe various derived features such as cumulative injection, which may contribute to a better predictive model. However, we also demonstrate that the decision to add more input features should be made considering the length and quality of training data that is available.

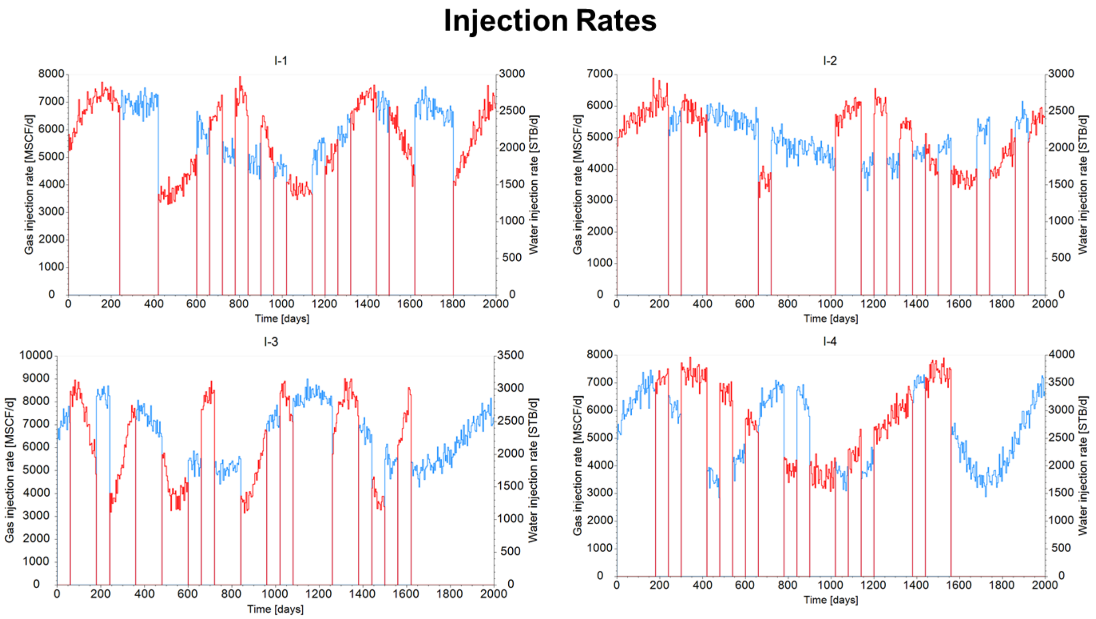
The second subsection describes the model regularization using well location information. Specifically, in cases where multiple injectors may have similar injection trends, the model should be able to pick out the more adjacent injector, rather than a distant one, in order to infer reasonable well connectivity maps. Two methods, preset radius and penalty function, are introduced to account for well locations, and the accuracy of the connectivity maps are examined using the streamline-based measurements.

Selection of Input Features

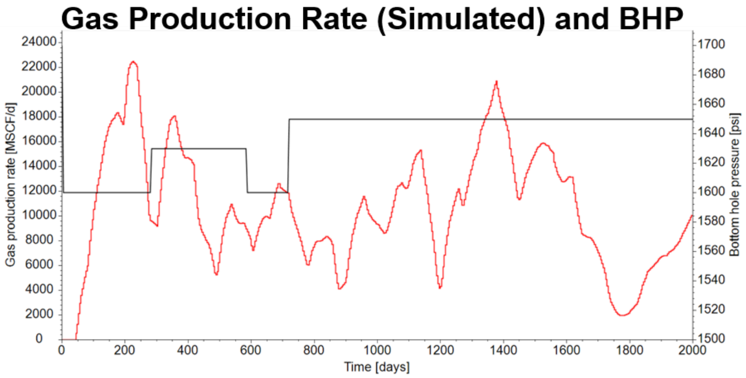
The first step in fitting the SRU is to decide on the input features going into the model. We start with a synthetic 3D heterogeneous model with four corner injectors and a single central producer, as shown in Figure 34 (a). The 3D model is of the dimension $50 \times 50 \times 10$ and the injectors are rate-controlled (with assigned rates shown in Figure 34 (b)) and the producer is operational at stable BHP, shown in Figure 34 (c). The simulated gas production at P-1 is shown in Figure 34 (c).



(a)



(b)



(c)

Figure 35 A synthetic 3D WAG case with a (a) 10-layer heterogeneous permeability field with 4 injectors and a central producer. The water and gas injection rates assigned to the four wells are shown in (b). The stable BHP constraint and simulated gas production at P-1 is given in (c). (Reprinted with permission from Chen et al., 2021)

First, we compare the effects of water injection vs gas injection for various lengths of training data to understand the relative contribution of each to the gas production. Later, we employ a similar analysis to decide the best set of inputs for training a model, given a dataset. At this stage, we shall also consider the addition of derived input features such as cumulative gas and water. By doing so, we also study the interplay between the number of features and the length of the dataset and its effect on the goodness of the model.

Water injection vs. Gas Injection

To compare the contribution of water injection vs gas injection, we train two models, one with just gas injection (red curve in Figure 36) and another with just water injection (green curve in Figure 36). The gas production observations (blue circles in Figure 36) were generated by adding noise to the simulated production from Figure 35 (c). To start with, we choose a small training dataset of length (N_{train}) 49 time steps, and predict for the rest of history. The training and test regions are shown in white and red backgrounds in the left panel of Figure 36. On the right panel, we see the cumulative production for the test case, as per the observation as well as the prediction of each model.

We see that the model that was trained on water alone, does a very poor job at prediction, as compared to the model trained on gas injection alone. As we increase the training length, even though both models do get better, only the gas injection-based model results in consistently good prediction (for $N_{train} = 147$). This shows that in this case, the gas production rate a strong function of the gas injection and water injection has a weak impact.

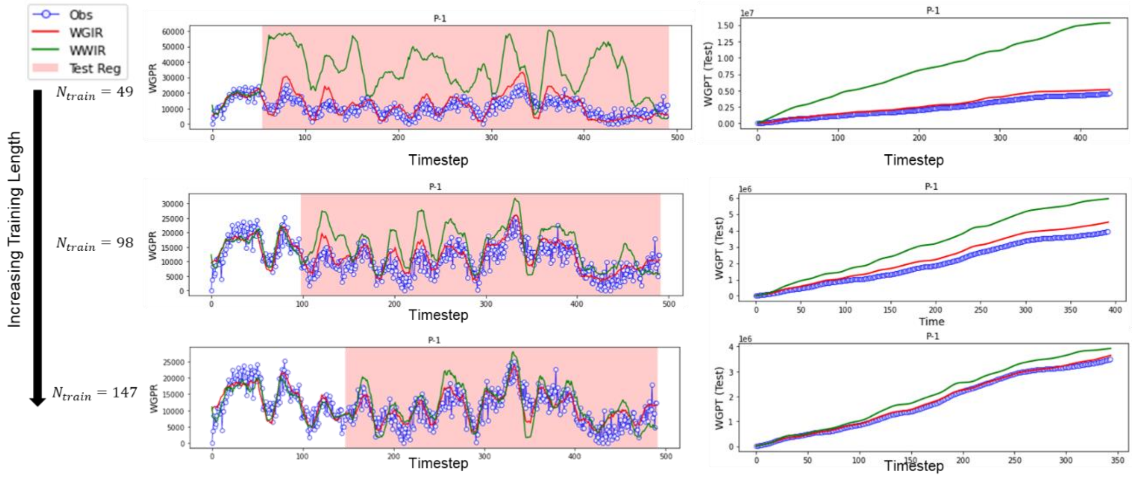


Figure 36 Comparison of SRU model prediction using water injection alone (WWIR) with that using gas injection alone (WGIR), for various training lengths. The SRU gas production rate predictions are shown on the left panel, where the training region is shown in the background and test region in the red background. The cumulative production rates for the test region alone are shown in the right panel. (Reprinted with permission from Chen et al., 2021)

Inclusion of derived features

In this section, we shall study the effects of adding several input features on the training length required for a good regression fit. To this end, we compare four sets of inputs for each injector. The resulting predictions are shown in Figure 37:

1. WGIR: Gas injection alone (Model M_1, solid red line in Figure 37)
2. WWIR+WGIR: Gas injection and water injection (Model M_2, dotted blue line)
3. WGIR+WGIT: Gas injection and cumulative gas injection (Model M_3, dotted red line)
4. WGIR+WGIT+WWIR+WWIT: Gas injection, water injection, cumulative gas injection and cumulative water injection (Model M_4, dotted green line)

For the short training dataset of $N_{train} = 49$, we see that the gas injection alone offers the best prediction. For this dataset of insufficient length, having more inputs results in a degradation of prediction performance. As we increase the dataset length to $N_{train} = 98$, both M_1 and M_3 predict well. The addition of water information (as in M_2 and M_4) hardly helps. As we increase the training dataset even further, all models predict accurately, as long as we have included the dominant input, which is the gas injection rates. However, it is clear that if our dataset is noisy and of insufficient length, it is safer to limit the input to the most dominant features.

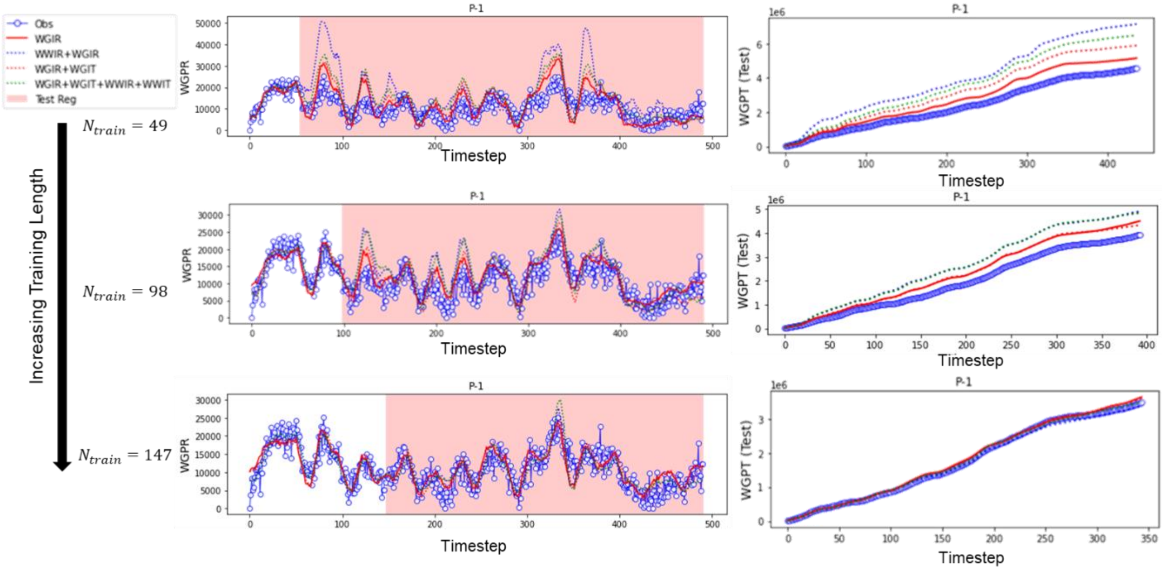


Figure 37 Comparison of SRU model prediction using four different input sets, for various training lengths. The SRU gas production rate predictions are shown on the left panel, where the training region is shown in the white background and test region in the red background. The cumulative production rates for the test region alone are shown in the right panel. The input sets being compared are gas injection along (WGIR, solid red), gas injection plus cumulative gas (WGIR+WGIT, dotted red), gas injection and water injection (WWIR+WGIR, dotted blue), gas and water injection along with their respective cumulatives (WGIR+WGIT+WWIR+WWIT, dotted green). The observed values of gas production are shown in blue circles. (Reprinted with permission from Chen et al., 2021)

Selection of Injectors

In this section we consider the choice of the injector set that is used to make the prediction at a particular producer. Especially in cases where multiple injectors have similar injection trends, it becomes necessary to impose some kind of location-based constraints on the SRU, in such a way that adjacent injectors exert more influence on a producer. Therefore, we examine ways to include location information into our model.

There are two possible approaches that we implement in this chapter. One way is to set a preset search radius where we consider only injectors which are within a specified radius around the producer. Subsequently, we do away with the pre-specified radius and perform a dynamic selection of injectors using a penalty function at the training stage. For the analysis, we use a 3D synthetic field of dimension $50 \times 50 \times 10$ with 12 rate-controlled injectors and 12 BHP-controlled producers, shown in Figure 38.

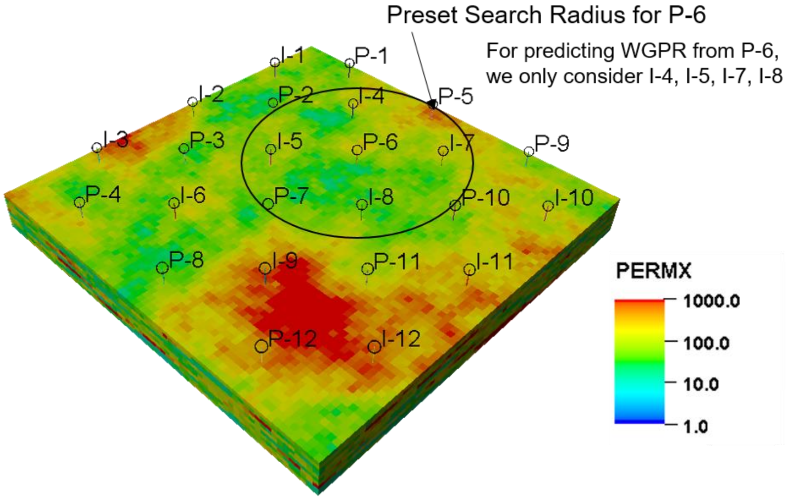


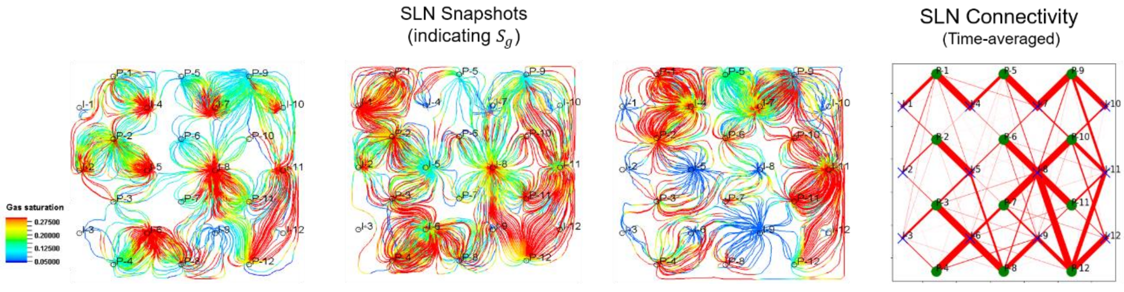
Figure 38 3D synthetic model with 12 injectors and 12 producers, used for testing the proposed methods for inclusion of location information. The search radius for producer P-6 is also plotted, within which all injectors are assumed to be capable of influencing the gas production at P-6, under the preset search radius method. (Reprinted with permission from Chen et al., 2021)

Reference Connectivity Map by Streamlines

Streamlines are the flow paths derived from the instantaneous velocity field under subsurface conditions (Datta-Gupta and King 2007). When streamlines are traced such that each streamline carries equal volumes of fluid (under reservoir conditions) to a producer, the streamline numbers from different injectors to the that producer reflect the interwell connectivity of the injector-producer pair. Accounting for the velocity field variation over time, the average streamline number over all training time steps, denoted as $N_{sln,i}$ (where i represents injector ID), is used to measure the physics-based connectivity. Similar to the metrics of normalized variable importance (VI), the connectivities represented by the streamline numbers with respect to the same producer are also normalized to their maximum value.

$$connectivity_{SLN_i} = \frac{N_{sln,i}}{N_{sln,max}} \tag{7}$$

Several streamline snapshots of the synthetic case are given in Figure 39(a)-(c) and the final streamline-based connectivity map is show in Figure 39(d).



(a) (b) (c) (d)

Figure 39 Streamlines traced at various time steps are shown in (a)-(c). The color indicates the gas concentration along the streamlines. The time-averaged streamline connectivity as computed by Eq. 7 is shown in (d). (Reprinted with permission from Chen et al., 2021)

Preset Search Radius Method

While making a prediction at a particular producer using an SRU with a preset search radius, we only consider injectors which fall within a specified radius around the producer. For example, in Figure 38, for predicting gas production at P-6, we consider only injectors I4, I5, I7 and I8 (which fall within the demarcated search radius) to have any influence. In other words, the prediction at P6 will be a function of the injection rates at I4, I5, I7 and I8. All weights that connect to other injectors are constrained to be zero.

The prediction results for all 12 producers using SRU trained with the preset search radius method are shown in Figure 40, Figure 41 and Figure 42. We can see that, while the prediction is reasonable for most of the wells, there are a few wells where the models are quite inadequate, such as P-8 and P-12 (Figure 42).

The SRU-based connectivity map was then generated by computing the permutation variable importance, as described in the previous section. This is shown in Figure 43(a). In order to validate the connectivity map, the simulator was run, followed by streamline tracing. Then we computed the average streamline-based fluxes over all training time steps, shown in Figure 43(b). Comparing these two, we see that for P12, the SRU based connectivity is missing two major fluxes between P12 and I8 and P12 and I11. These connections were not picked up by the SRU because these injectors are outside the search radius of P12. This explains the poor regression performance for this well.

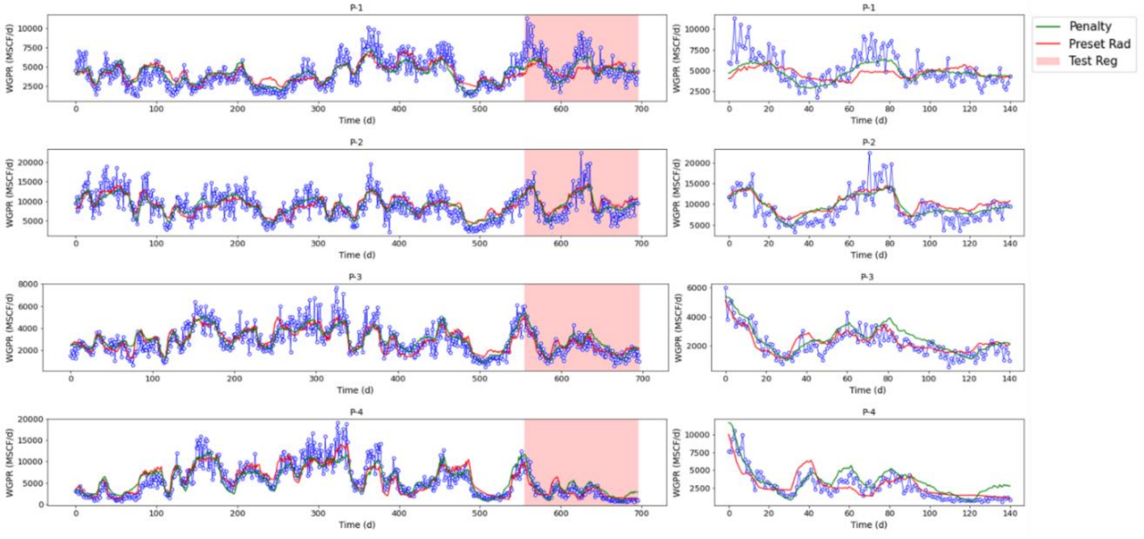


Figure 40 SRU regression results for producers P-1, P-2, P-3 and P-4 using the preset search radius method (red curve) and penalty function method (green curve). The left panel shows the SRU predictions along with the observed gas production rates. The region with white background denotes training and that with red background denotes testing. The right panel shows a close-up view of the testing performance for clarity. (Reprinted with permission from Chen et al., 2021)

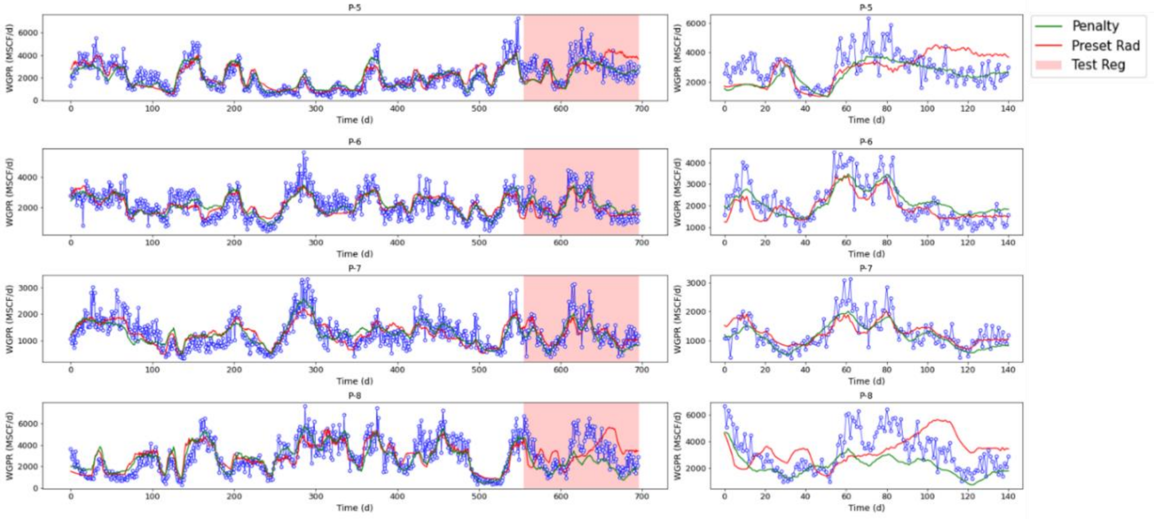


Figure 41 SRU regression results for producers P-5, P-6, P-7 and P-8 using the preset search radius method (red curve) and penalty function method (green curve). The left panel shows the SRU predictions along with the observed gas

production rates. The region with white background denotes training and that with red background denotes testing. The right panel shows a close-up view of the testing performance for clarity. (Reprinted with permission from Chen et al., 2021)

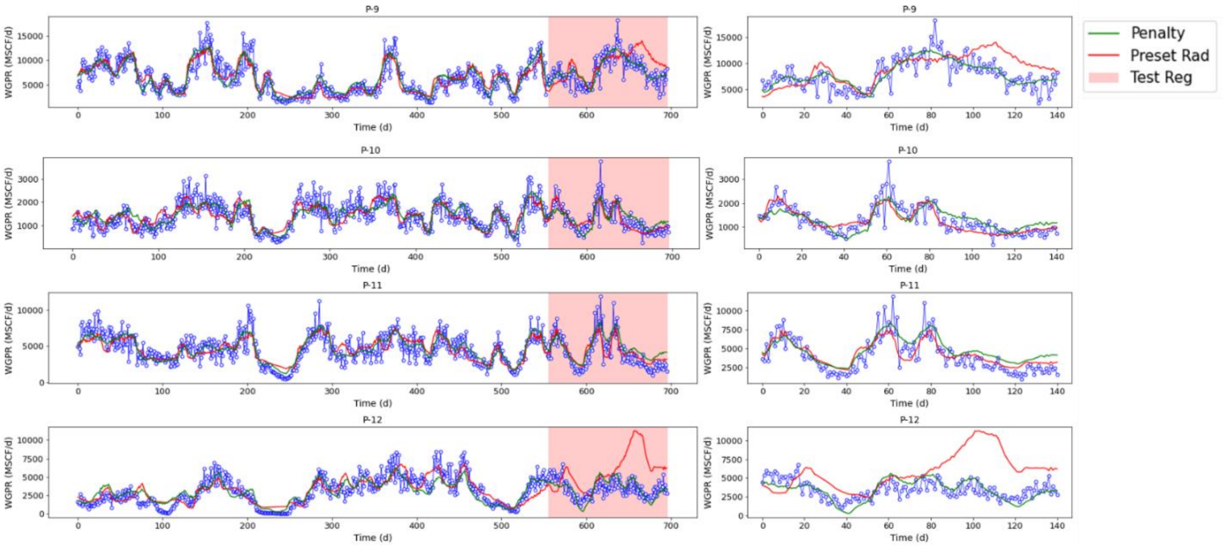


Figure 42 SRU regression results for producers P-9, P-10, P-11 and P-12 using the preset search radius method (red curve) and penalty function method (green curve). The left panel shows the SRU predictions along with the observed gas production rates. The region with white background denotes training and that with red background denotes testing. The right panel shows a close-up view of the testing performance for clarity. (Reprinted with permission from Chen et al., 2021)

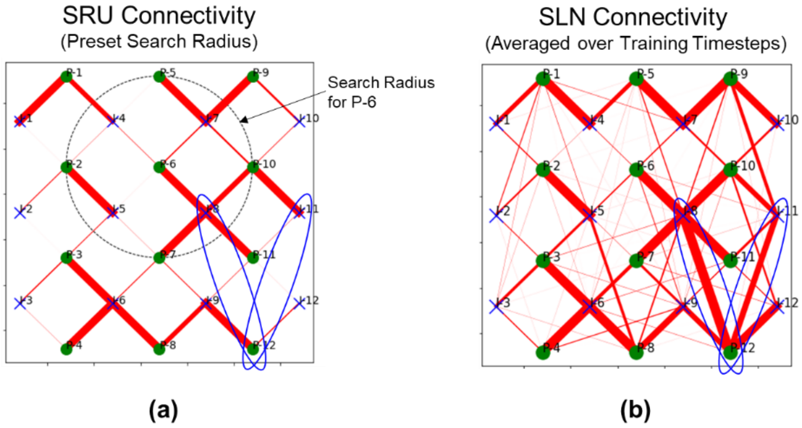


Figure 43 The SRU-based connectivity map using the preset search radius method is shown in (a). The streamline-based flux allocation averaged over all training time steps is given in (b). (Reprinted with permission from Chen et al., 2021)

Penalty Function Method

In order to mitigate the problem of missed connections due to poor choice of search radius, we introduce a penalty function for training the SRU in such a way that long injector-producer connections are automatically penalized during training. This way, we eliminate the need to pre-specify the search radius and thereby, we use all injectors for modeling the gas production at all producers. However, SRU weights for a particular producer are computed by minimizing a loss function, given in Eq. 8, which has an extra regularization term along with the usual MSE loss.

$$Loss = \frac{1}{T}\sum_{t=1}^{T}\left((y_{obs}(t) - y_{pred}(t)\right)^2 + c\sum_{i=1}^{N_i}\left\{\sum_{j=1}^{N_{w_i}}|w_{ij}|\right\}d_i \tag{8}$$

In Eq. 8, the first part of the loss expression is the MSE loss computed from the observed data and the corresponding prediction whereas the second part of the expression is the proposed penalty function. Let there be N_{w_i} trainable weights that are associated with (and only with) the i^{th} injector (out of a total of N_i injectors). The absolute value of these weights $(w_{i1}, w_{i2}, \ldots, w_{iN_{w_i}})$ are multiplied with the distance from the producer to the i^{th} injector and summed together. This sum is multiplied by a penalty coefficient c and added to the MSE to obtain the loss function that is minimized during training. This way, long connections are automatically penalized more and the strength of the regularization is set by the penalty coefficient, c in this expression.

Having defined the penalized loss in this manner, the next task is to decide on the best value of c. We trained several SRU models with different values of c and recorded the training MSE, shown in Figure 44(a). As we increase the penalty coefficient, the training MSE is flat at first (from $c = 0$ to $c = 0.0003$), and then starts to increase. The connectivity detected in this flat portion is shown from Figure 44(c) to (g), with Figure 44 (c) having zero penalty, and Figure 44 (g) having a value of 0.0003, which falls almost at the elbow of the plot in Figure 44 (a).

The streamline-based flux map, which is used as benchmark for validation, is shown in Figure 44(b). We see that for the zero penalty case, the SRU picks up several false connections in addition to the true ones. However, as we increase the penalty coefficient slowly, these spurious connections disappear, without any drop in training MSE. This way, we choose 0.0003 as our preferred penalty coefficient, whose regression quality is shown by the green curve in Figure 40 to Figure 42. We note the clear improvement in both training and test predictions by using the penalty function method over the preset search radius method. This improvement is even more significant in wells such as P-12, wherein the regression quality is improved, as the distant connections to I-8 and I-11 are captured by the latter method.

However, a caveat in using the penalty method is illustrated in Figure 45 wherein, distant yet important connections are removed if we use a higher value of penalty. Here, as we increase c from 0.0001 to 0.01, not only does the training error increase but we also lose longer connections such as those between producer P-12 and injectors I-8 and I-11. This emphasizes the importance of choosing an appropriate value c that is large enough to prevent false connections, yet not too large as to cause a high increase in training error and loss of information.

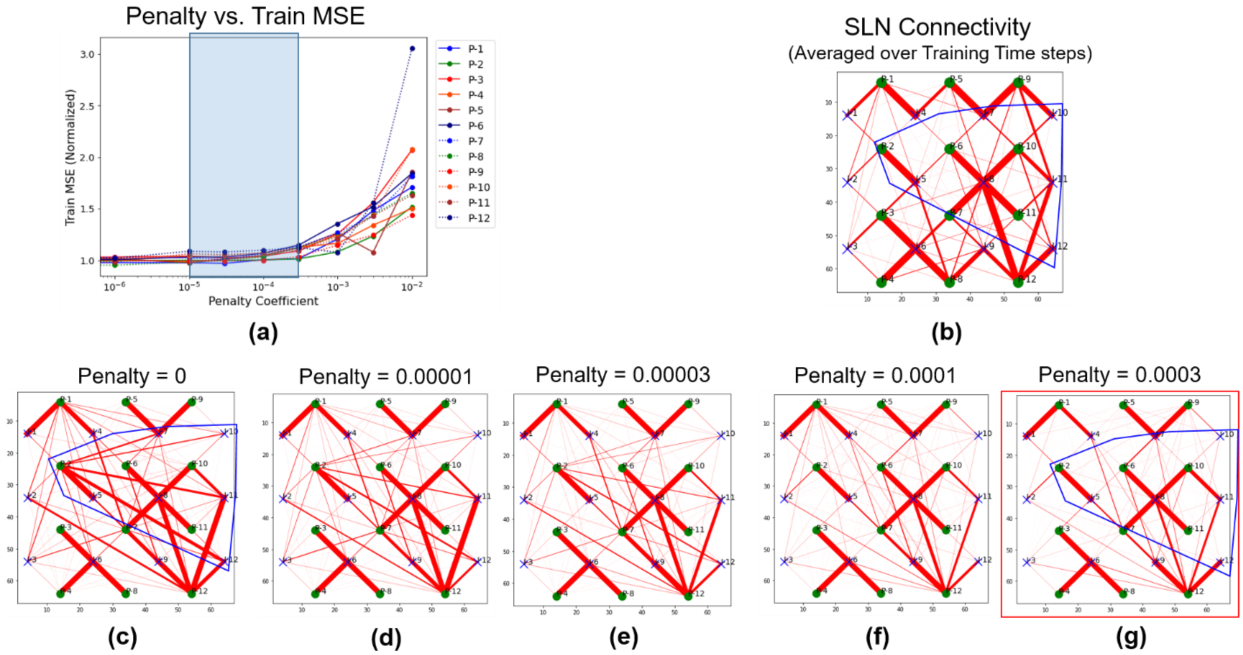
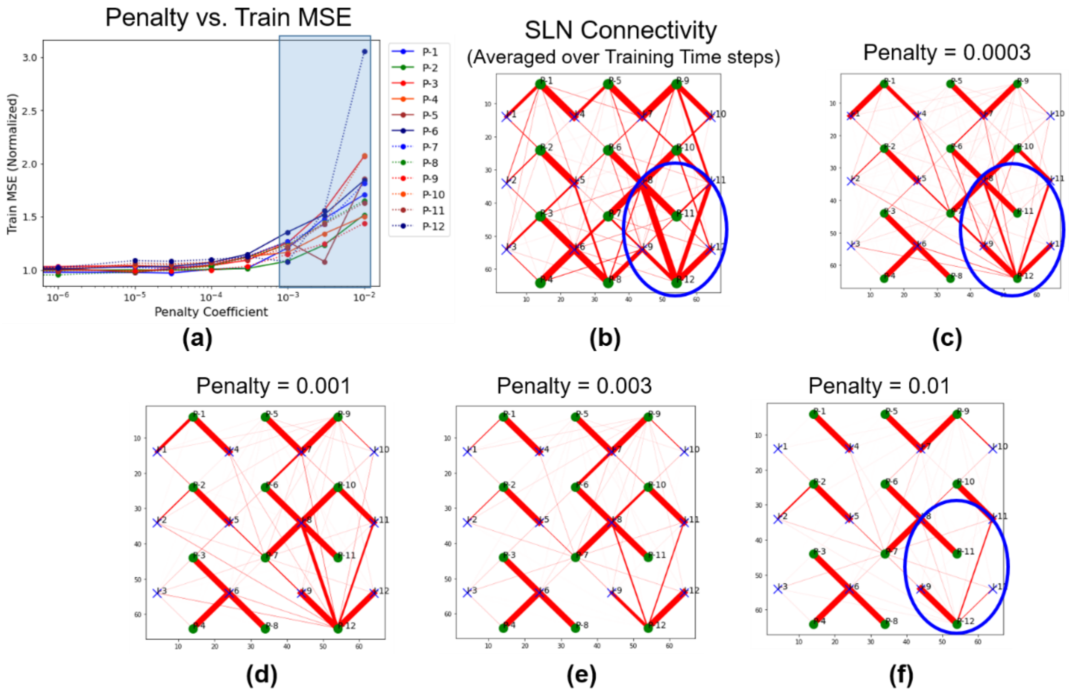


Figure 44 The results of connectivity detection using the SRU trained with the penalty function method is illustrated. The degradation in training MSE with increase in the penalty coefficient is plotted for each well. The streamline-based fluxes that are used as a benchmark connectivity is shown in (b). Maps (c) to (g) shows the SRU-based well connectivity for various values of penalty coefficient in the highlighted flat region of the plot (ranging from zero to 0.0003). It is seen from (c) to (g) that increasing penalty removes spurious connections while maintaining the training performance. The connectivity map for $c = 0.0003$ (at the elbow of (a)) is substantially close to the streamline-based map. (Reprinted with permission from Chen et al., 2021)



(a) (b) (c)

(d) (e) (f)

Figure 45 Increasing the penalty coefficient beyond the elbow in (a) corresponding to $c = 0.0003$ leads to significant loss of connectivity information, as shown in maps (d) to (f). The streamline-based flux map and the connectivity map at $c = 0.0003$ are shown for comparison. (Reprinted with permission from Chen et al., 2021)

Field-Scale Application

The proposed methodology was applied to a model cut out from a real field case introduced by Olalotiti-Lawal, Onishi, Kim, et al. (2019), which focused on a CCUS project wherein a mature oil field under CO_2 was subjected to water-alternating gas injection. A test region in the model, shown in Figure 46, with 15 injectors and 14 producers was chosen for this study. Unlike the synthetic models used in our prior applications, the model exhibits a high degree of areal and vertical heterogeneity, as evident in Figure 46(b).

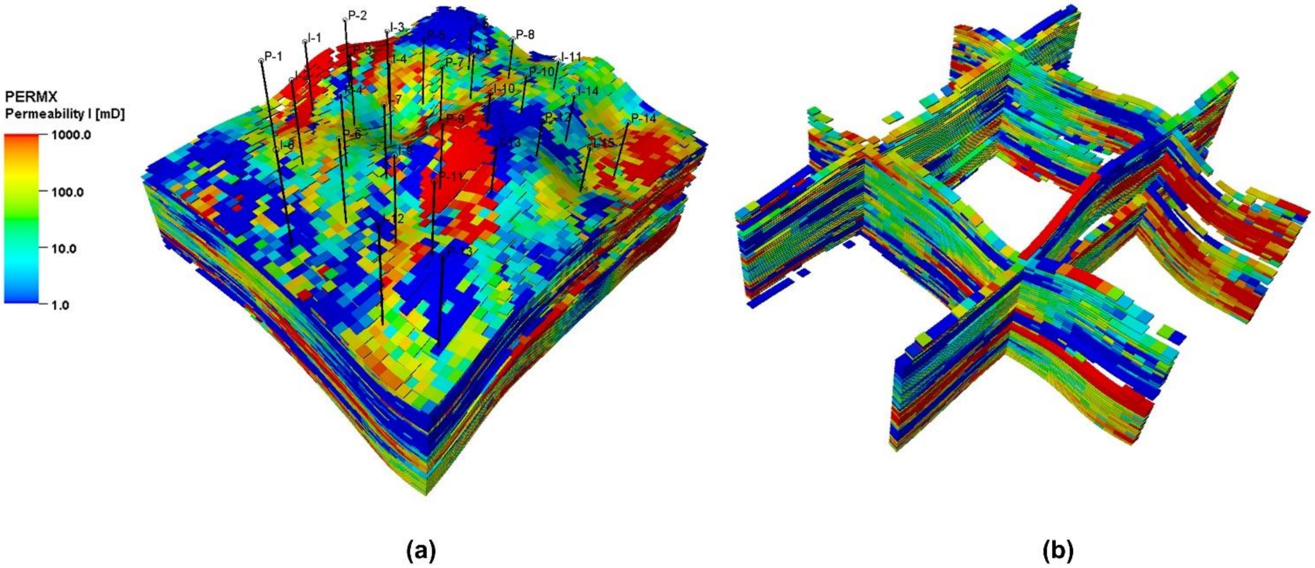


Figure 46 Field-scale case model description showing well configuration and heterogeneity (areal and vertical). (Reprinted with permission from Chen et al., 2021)

The injectors are operated under the rate constrained mode with realistic injection rate schedules, samples of data shown in Figure 47. The producers operate under stable BHP conditions and samples of the simulated data are shown in Figure 48. Additionally, unlike the previous synthetic cases studied, both injectors and producers have intermittent periods of shut-in, which are a common occurrence in a real-field setting.

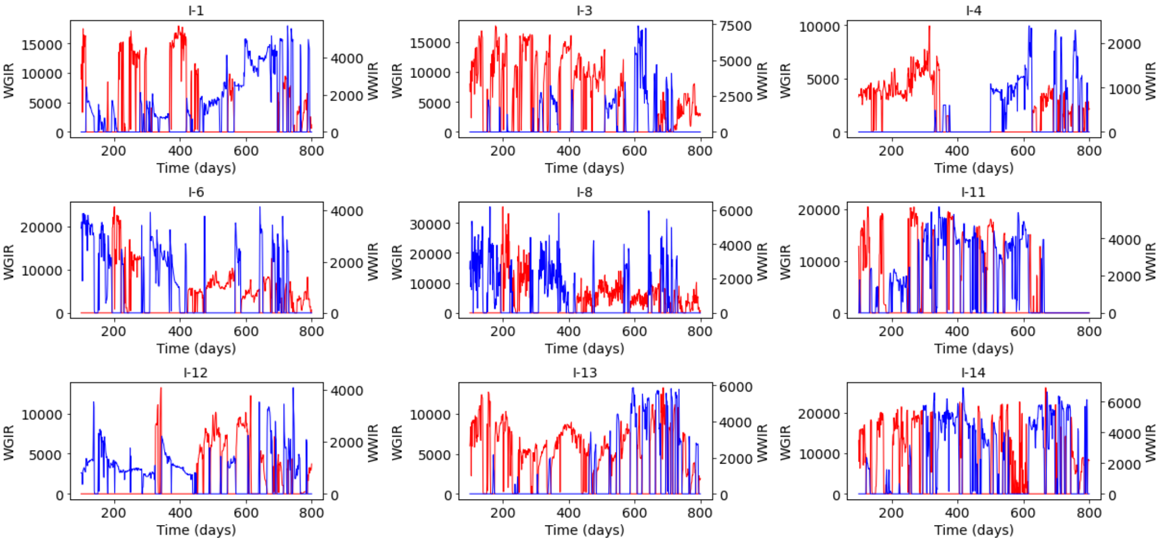


Figure 47 The gas injection rates (WGIR, in Mscf/d) in red and water injection rates (WWIR, in bbl/d) in blue for a selected number of injectors are shown here. The injection rates include intermittent shut-ins, which are common in a real-field setting. (Reprinted with permission from Chen et al., 2021)

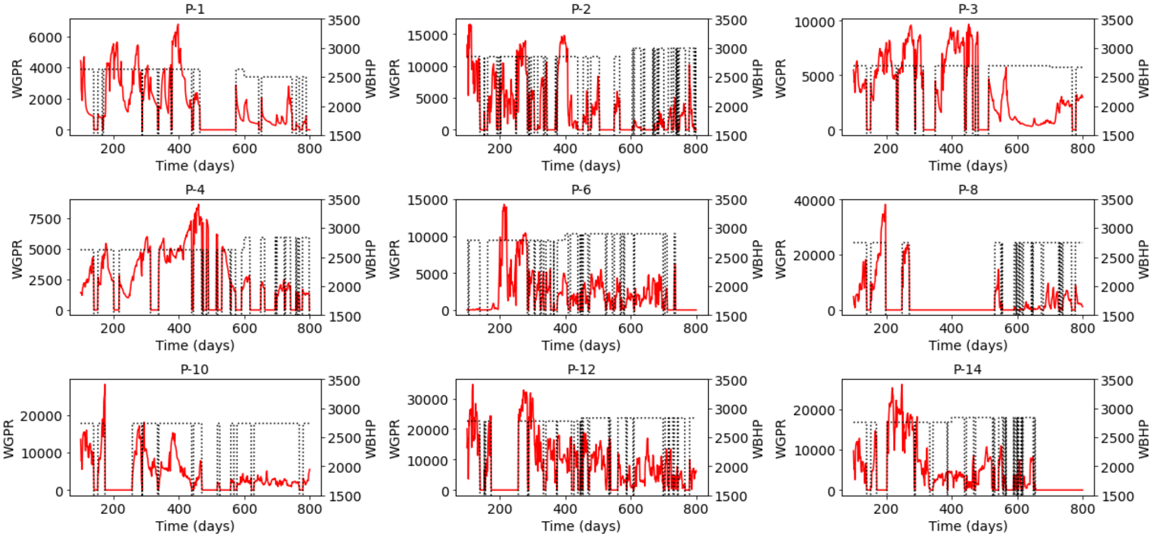


Figure 48 The gas production rates WGPR (Mscf/d) in red and the stable BHP levels (psia) in dash black for a selected number of producers are shown here. The producers are subject to intermittent shut-ins, which are common in a real-field setting. (Reprinted with permission from Chen et al., 2021)

We trained a number of SRU models with varying penalty coefficients and recorded the training MSE as shown in Figure 49.

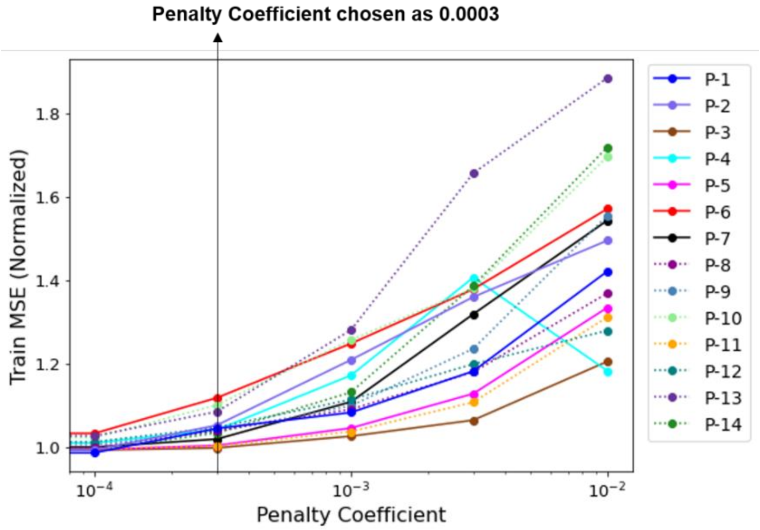


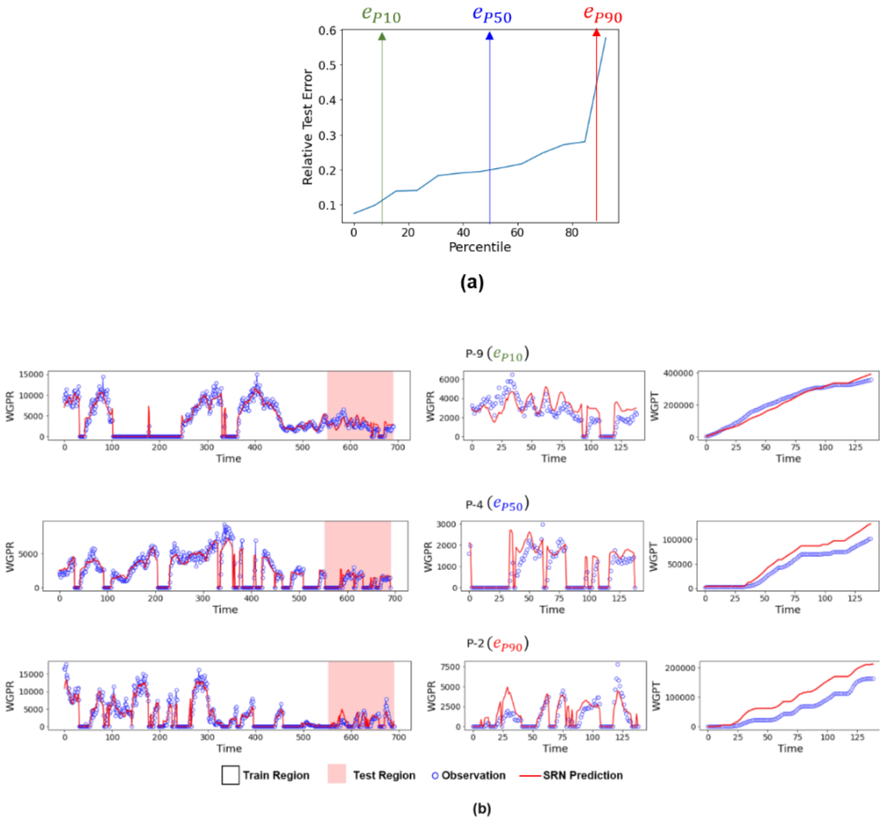
Figure 49 Variation of training loss (mean squared error) for each producer with different values of penalty coefficients. (Reprinted with permission from Chen et al., 2021)

An optimal penalty coefficient of 0.0003, at the knee of the plot (Figure 49), was chosen, and the resulting regression performance is shown in Figure 50. The relative test error for each well was computed as given in Eq. 9 and this is shown in Figure 50(a).

$$Relative\ Test\ Error = \frac{\sum_{t=1}^{T}\left(\left(y_{obs}(t) - y_{pred}(t)\right)^2\right.}{\sum_{t=1}^{T}\left(y_{obs}(t)\right)^2} \tag{9}$$

Since we have 14 producers, we show the only the wells corresponding to P10, P50 and P90 relative errors in Figure 50(b). For example, out of all the wells for which we make predictions, producer P-9 exhibits one of the best testing performance, whereas P-4 represents the median prediction performance. Likewise, producer P-2, whose relative test error corresponds to P90 in the error distribution indicates one of the worst regression performance of the SRU in this case.

Subsequently, the permutation variable importance was computed for each injector-producer pair as explained in the earlier section. In order to validate the connectivity map based on SRU variable importance, we traced streamlines at each time step and computed the average gas phase production allocation. The streamline configurations at a few time steps are shown in Figure 51.



(a)

(b)

Figure 50 The distribution of the relative test error for all 14 producers, computed as shown in Eq. 9. (b) A selected number of regression fits (corresponding to P90,

P50 and P10 errors) are shown. The left panel shows the regression fit for the training and test regions. The middle panel shows the test region zoomed in for better clarity. The right panel shows the cumulative gas production as per observed data and the SRU predictions. (Reprinted with permission from Chen et al., 2021)

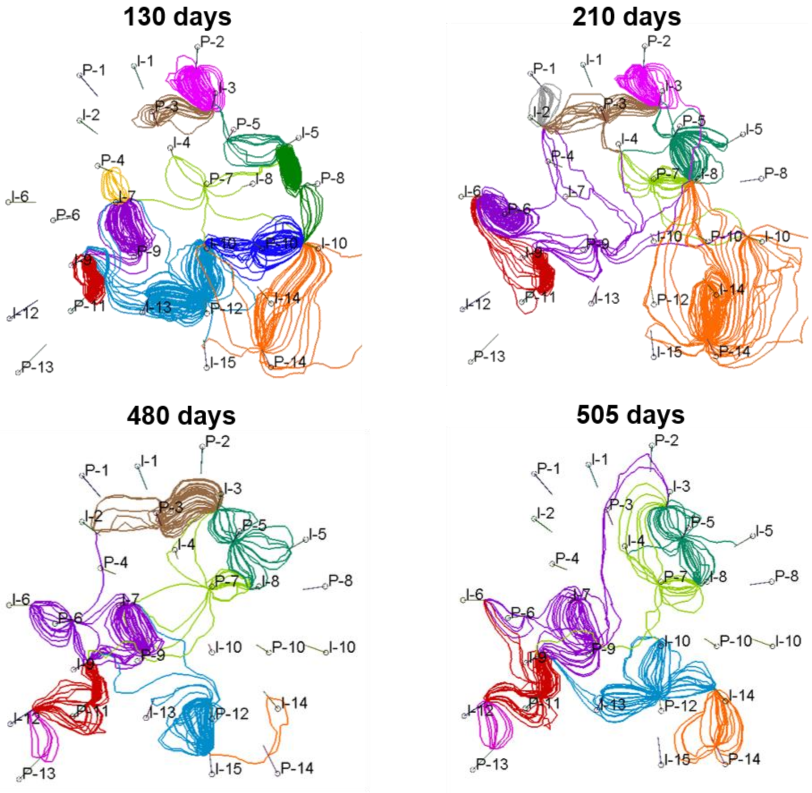


Figure 51 Streamline configurations at different time steps. The colors indicate the producer (sink) at which the streamline terminates. (Reprinted with permission from Chen et al., 2021)

The average streamline connectivity was computed by averaging these flux allocations across all time steps. A comparison between the streamline-based connectivity map and the SRU-based connectivity map is given in Figure 52. The top panel Figure 52 (a, b) shows all the connections inferred by both methods. The strong fluxes were filtered by setting a threshold on the normalized connectivities and plotted in the bottom panel Figure 52(c, d). It is seen that the strong connection inferred by the SRU is in almost perfect agreement with those based on streamlines. However, we see that the inferences of weaker fluxes are highly unreliable and indicates the need for a better model, which may be achieved with more data.

A major advantage of the SRU-VI method is its time efficiency. For training a dataset of 550 days (sampling frequency is 1 observation per day), the SRU takes around 1 minute to train each well. Moreover, the prediction time for the entire dataset (of 700 datapoints) is ~0.5 second per producer. The computation time for VI for an injector-producer pair is dependent on the number of random permutations we perform while computing the expectation in Eq. 5. In our work we used 10 permutations, bringing the computation time to ~5 seconds per injector-producer pair.

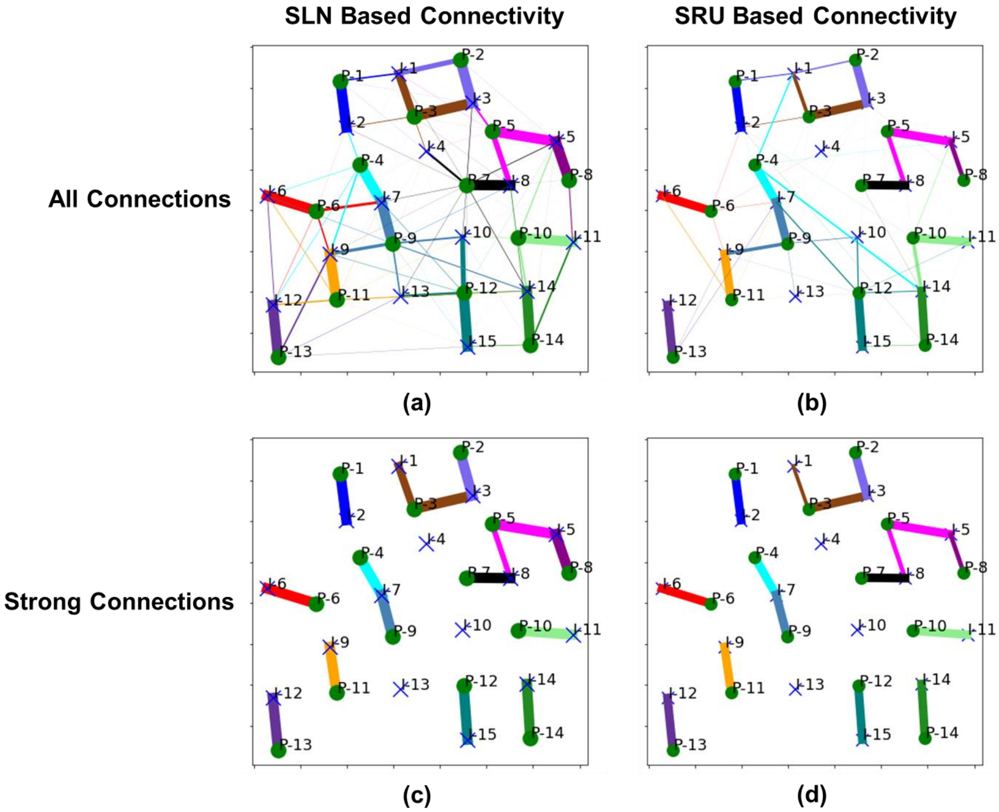


Figure 52 Connectivity maps generated from the proposed methodology compared with average streamline fluxes. The proposed method picks up all strong fluxes even though the inference of weak fluxes is highly uncertain. (Reprinted with permission from Chen et al., 2021)

Computation of Well Pair Injection Allocation and Production Allocation from SRU-Based Variable Importance

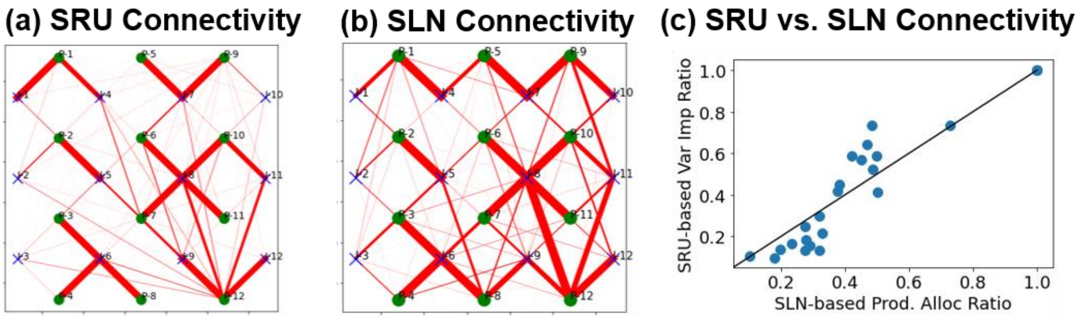
The next stage of study is directed at estimating the volumes of gas injection and production allocation from the connectivity obtained using SRU and permutation variable importance. In the previous sections, we established that the SRU-derived variable importance is in close correspondence with the average streamline based connectivity, especially for stronger fluxes. In order to demonstrate this further, we first computed the streamline based production allocation ratio for each well pair ($CumuProdAllocRatio_{I_iP_j}$) , i.e., well pair production allocation ($CumuProAlloc_{I_iP_j}$) normalized by the sum of allocation along each producer.

$$CumuProdAllocRatio_{I_iP_j} = \frac{CumuProAlloc_{I_iP_j}}{\Sigma_j CumuProAlloc_{I_iP_j}} \tag{10}$$

Next, we similarly computed the variable importance ratio ($VarImpRatio_{I_iP_j}$) , as the well pair wise variable importance ($VarImp_{I_iP_j}$) normalized by its sum along each producer.

$$VarImpRatio_{I_iP_j} = \frac{VarImp_{I_iP_j}}{\Sigma_j VarImp_{I_iP_j}} \tag{10}$$

A scatter plot of the variable importance ratio versus the production allocation ratio for the strong fluxes in the synthetic 3D WAG case (Figure 35) is shown in Figure 53 (c) and demonstrate the close correspondence between the two. Therefore, it is possible to estimate the streamline-based well pair production allocation ratio using the SRU-based variable importance.



(a) SRU Connectivity

(b) SLN Connectivity

(c) SRU vs. SLN Connectivity

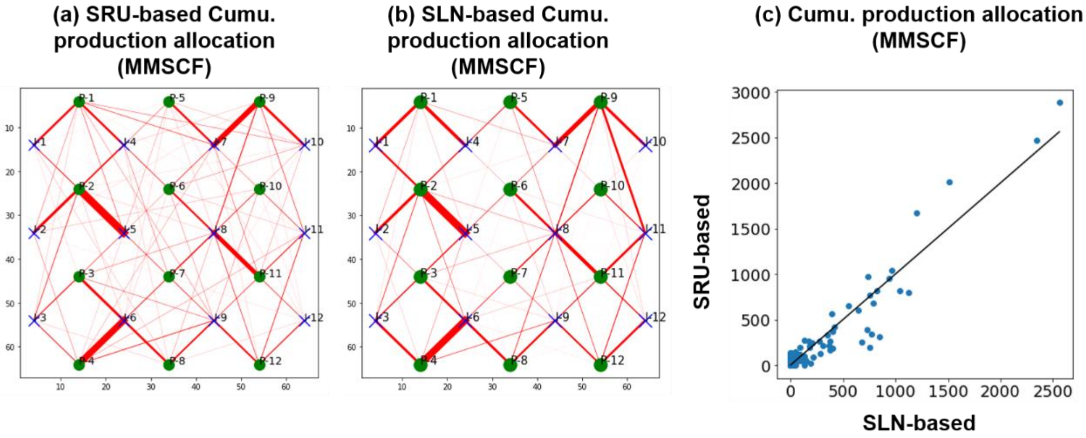
Figure 53 A comparison of (a) SRU-based and (b) streamline based connectivity can be made by (c) plotting the well pair production allocation ratio (from

streamlines) against the variable importance ratio (from SRU). The points rough fall on the x=y line, indicating a good correspondence.

Under the assumption that the variable importance ratio and the production allocation ratio are close in value, one may compute the well pair production allocation volumes by simply multiplying the variable importance ratio by the cumulative gas production from a producer (Q_{P_j}).

$$ProdAlloc_{I_iP_j} = VarImpRatio_{I_iP_j} \times Q_{P_j} \tag{10}$$

The results of doing so are given in Figure 54, wherein we are able to generate reasonable estimates of the actual well pair cumulative production allocation using the SRU. Figure 54 (a) and (b) show the cumulative production allocation of each well pair as computed using the proposed workflow and using streamlines respectively. The thickness of the red lines are indicative of the strength of the allocation in MMSCF. Furthermore, a scatterplot of the well pair cumulative production allocations obtained from SRU against those from streamlines (Figure 54 (c)) demonstrate a good correspondence.



(a) SRU-based Cumu. production allocation (MMSCF)

(b) SLN-based Cumu. production allocation (MMSCF)

(c) Cumu. production allocation (MMSCF)

Figure 54 (a) Well pair cumulative production allocation volumes were computed from the SRU-based variable importance and compared with (b) the values obtained from streamlines. (c) A scatterplot of the SRU-based volumes versus the streamline-based volumes demonstrate a good agreement between the two.

Subsequently, we estimate the well pair cumulative injection allocation from the production allocation that we computed in Figure 54. In order to do so, we need to first test the relationship between the cumulative injection allocation and the cumulative production allocations for the case under study (Figure 55). For the current model, we see that, even though the instantaneous injection allocation and instantaneous production allocation show little correspondence (Figure 55 (a)), the cumulative injection and production allocation are seen to be linearly correlated (Figure 55 (b)).

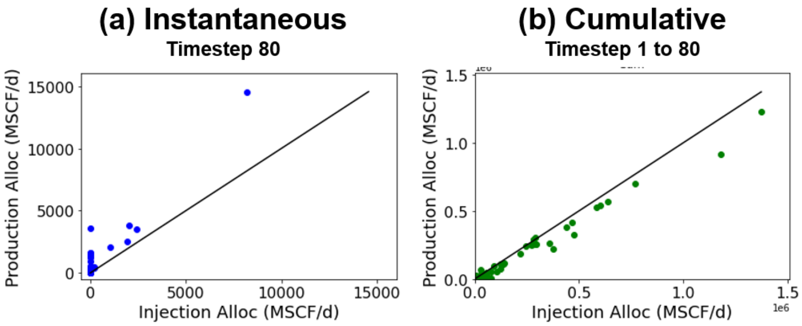
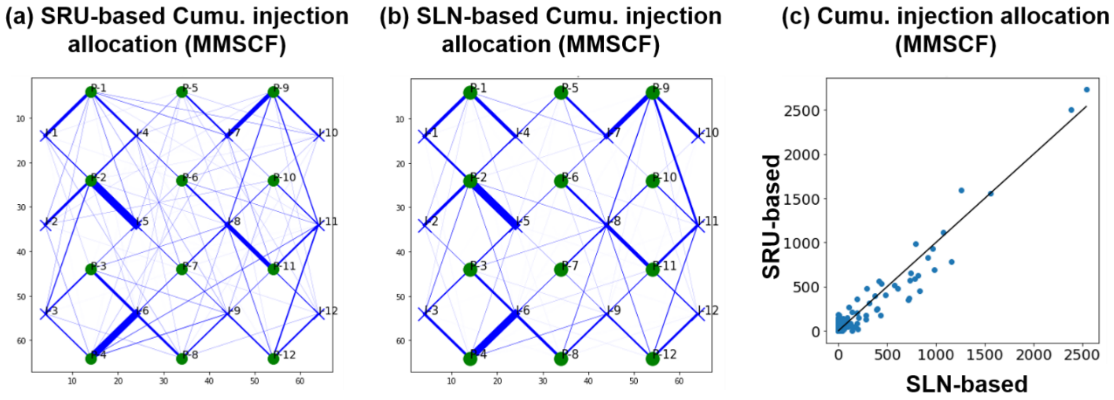


Figure 55 (a) The instantaneous streamline-based well pair production allocation plotted against the corresponding well pair injection allocation for the synthetic 3D WAG model shows no clear relationship between the two. However, (b) shows a clear linear proportionality between the cumulative production allocation and injection allocation (computed over a period of time).

This observation may be utilized to convert the production allocation volumes into injection allocation values as per the following equation:

$$CumuInjAlloc_{I_iP_j} = \frac{CumuProAlloc_{I_iP_j}}{\Sigma_j CumuProAlloc_{I_iP_j}} \times CumuInjection_i \tag{10}$$

In this equation, $CumuInjAlloc_{I_iP_j}$refers the well pair cumulative injection allocation, from injector I_i to producer P_j. $CumuProAlloc_{I_iP_j}$ refers to the well pair cumulative production allocation from I_i to P_j that was computed in step 1 using the variable importance connectivity and the total production at P_j. $CumuInjection_i$ represents the cumulative injection at I_i. The resulting values of well pair cumulative injection allocation from SRU are compared with the actual streamline-derived values in Fig… The comparison shows that the SRU-derived and streamline-derived volumes align closely along the $y = x$ line in Figure 56.



(a) SRU-based Cumu. injection allocation (MMSCF)

(b) SLN-based Cumu. injection allocation (MMSCF)

(c) Cumu. injection allocation (MMSCF)

Figure 56 (a) Well pair cumulative injection allocation volumes were computed from the SRU-based variable importance and compared with (b) the values obtained from streamlines. (c) A scatterplot of the SRU-based volumes versus the streamline-based volumes demonstrate a good agreement between the two.

Figure 56 (a) and (b) show the cumulative injection allocation of each well pair as computed using the proposed workflow and using streamlines respectively. The thickness of the blue lines are indicative of the strength of the allocation in MMSCF. Furthermore, a scatterplot of the well pair cumulative injection allocations obtained from SRU against those from streamlines (Figure 56 (c)) demonstrate a good correspondence.

Summary and Conclusion

In this work we establish a machine-learning framework using the statistical recurrent unit for general time series problems. Using the proposed SRU as a proxy model for predicting production rates, we perform injector-producer connectivity detection by computing the variable importance of injection rates on the production rates.

- The inputs, outputs and loss function of the SRU model are customized to the CO_2 WAG problem. We perform input feature selection to capture underlying patterns with limited data. A penalty function is introduced to include well location information for regularizing the neural net weights.
- Streamlines are used for quantifying physics-based inter-well connectivity for comparison with the SRU model.
- Synthetic and field-scale tests show that the dominant connections identified by the data-driven SRU method and streamline method are in close agreement. Moreover, time-cost for the data-driven method is trivial. This makes the proposed method highly efficient and advantageous for practical field applications.
- The approach can be easily extended to estimate injection and production allocation volumes between well pairs

CHAPTER IV

ACCELERATING WATER FLOOD OPTIMIZATION UNDER GEOLOGIC UNCERTAINTY BY FLOW FEATURE CLUSTERING

Introduction

Waterflooding involves the injection of water into the depleting reservoir through injectors to increase/maintain reservoir pressure while oil is being extracted at the producers. This is one of the most extensively used improved oil recovery methods employed post primary depletion. The presence of high-permeability streaks in a reservoir may lead to early water breakthrough at the producers, resulting in premature rate decline and reduced oil recovery per barrel of water injected. Hence, optimizing well rates for a flow field in favor of recovering remaining oil and reducing water cycling is critical to the success of waterflood projects. Given a geologic model, there exists several simulation-based approaches for waterflood optimization. However, the optimal schedule based on one individual geologic model may not necessarily result in favorable outcomes for the real field due to the geologic inconsistencies between the real and the model (Sen et al. 2020). Geologic uncertainty is typically represented by an ensemble of history matched realizations of the reservoir. Subsequently, optimal schedules for each well shall also be expressed as a distribution rather than single values. The corresponding uncertainty assessment is crucial to decision making regarding field implementation. In this study, we focus on formulating and accelerating the generation of the distribution of optimal rates of each well in waterflood projects, while considering geologic uncertainty.

An extensive literature survey on waterflood optimization was provided by Sen et al. (2020). Sudaryanto and Yortsos (2001) developed an injection optimization workflow for single-phase incompressible flow based on optimal control theory, which was tested on a rectangular bounded system with two injectors and one producer. Nævdal et al. (2006) developed a closed-loop control approach using ensemble Kalman filters (EnKF) in order to maximize the NPV subject to constraints. This approach requires a simulator to be run for each member in the ensemble, so that a control vector that maximizes the objective function can be computed by solving the adjoint equation using steepest descent. Alhuthali et al. (2007) proposed a streamline-based methodology that involves equalizing the arrival time of the waterfront at producers in separate subregions of the reservoirs. Chen et al. (2020b) proposed a streamline-based gradient-free method for maximizing the cumulative oil production under constraints such as field injection and liquid production rates, allowable ranges on well rates and bottomhole pressures, and operational events like infill well drillings and well conversions.

Since all the methods described are simulation-based, a geologic model of the reservoir becomes a key requirement for their applicability. A geologic model is typically initialized with structural and geophysical measurements and in turn calibrated (via history-matching) by dynamic measurements from the field, such as well and field-level pressures and fluid rates. The geologic uncertainty may be represented by multiple realizations generated by history matching techniques, which can be carried out by evolutionary methods like differential evolution (Hajizadeh et al. 2018) and multi-objective genetic algorithm (Ferraro and Verga 2009), or statistical methods like ensemble-smoother with multiple data assimilation (ESMDA) (Emerick and Reynolds 2013) and ensemble-based nonlinear orthogonal matching pursuit algorithm (Elsheikh et al. 2013). These methods generate an ensemble of realizations that can reproduce the observations to a sufficient degree, but usually require large numbers of simulations and are typically performed using coarse-scale models. The streamline-based history matching methods (He et al., Cheng et al., Chen et al. 2020b and 2021, Liu et al. 2020) can efficiently tune the fine-scale geologic models mapped from the coarsen-scale realizations, so that misfit between simulation curves and observed data can be further reduced in well-level for fine-scale models.

In order to capture the full range of geologic uncertainty, a large number of history-matched realizations are typically required. Subsequent applications like rate optimization will hence require simulation runs over all realizations and can be computationally prohibitive for models with a large number of cells. Therefore, sampling methods that reduce the realization pool size while preserving the geological uncertainty are needed to mitigate the workload engaged in such ensemble-based studies. Clustering refers to any unsupervised method that groups a dataset into subsets such that similar datapoints fall in the same subset (Rokach and Maimon 2005). K-means clustering is one of the most used and easily implementable clustering algorithms. This involves the grouping of datapoints to say, K clusters, in such a way that the within-cluster sum of squares is minimized (Hartigan and Wong 1979). Other clustering algorithms include spectral clustering (Von

Luxburg 2007), DBSCAN (Birant and Alp 2007), and affinity propagation (Dueck 2009). Cluster analysis has long been used to perform model selection in order to identify model groups that show similar recovery responses, and thereby capture the uncertainty (Baker 2015). Sharma et al. (2008) conducted a systematic uncertainty analysis of oil-in-place in carbonate reservoirs, whereas K-means clustering was used to sample from the component distributions. More recently, clustering was used in the context of unconventional wells, for quantifying uncertainty in production forecasting (Ravikumar and Lee, 2020).

In this study, we describe a comprehensive workflow for the uncertainty analysis of optimal rates for each well engaged in waterflood optimization, where the history matching process generates geologic realizations, the streamline method performs rate optimization for each realization, and the uncertainties of optimal well rates are summarized and demonstrated using boxplots and spatial bubble plots. On top of that, we also propose a clustering-method, based on the flow features extracted using streamlines, to scale down the workload to a smaller set of geologic realizations, while preserving the uncertainty from the full-ensemble. The rest of this chapter is organized as follows: The section on ‘Methodology’ provides a detailed description of the workflow and the various algorithms that have been used in this work. Next, the workflow is applied to a 2D synthetic case, starting with generation of the history-matched ensemble to the well-wise uncertainty analysis of optimal rates. Subsequently, we apply the workflow to the 3D SAIGUP model (Matthews et al. 2008) to demonstrate the feasibility of the workflow for application to a field-scale scenario. Finally, the ‘Conclusions’ section summarizes the study.

Methodology

In this work, we propose a clustering-based approach to reduce the number of optimization runs required for obtaining reliable optimized schedules, while accounting for geologic uncertainty. A brief summary of the workflow is shown in Figure 57.

1. **History-matching.** Given a series of observed historical data on the production watercut and well bottom-hole pressures for time periods $t = 1,2, \dots, T_{hist}$, we start with generating multiple realizations of history matched models. For this purpose, we may use an evolutionary method or any statistical algorithm such as Ensemble Smoother with Multiple Data Assimilations (ESMDA).
2. **Streamline Tracing and Clustering**. We compute the gridblock-wise streamline-based time of flight (TOF) for each realization from the resulting flux field at the end of history (i.e. at $t = T_{hist}$) and apply k-means clustering algorithm to generate N_{clust} subsets of the realizations.
3. **Rate Optimization and Uncertainty Analysis**. The rate optimization algorithm is performed on the representative realizations for each cluster (cluster centroids) for a number of time periods (say t_{opt}). Doing so, we obtain the optimized schedules for $t = T_{hist} + 1, \dots, T_{hist} + t_{opt}$ for each centroid realization. In order to quantify and visualize the uncertainty in the optimal schedule, we compute the average optimal rate during the optimization window as:

$$q_x^* = \frac{\sum_{t=1}^{t_{opt}} q_{x_{T_{hist}+t}} \Delta T_t}{\sum_{t=1}^{t_{opt}} t \Delta T_t} \qquad (1)$$

Where $q_{x_{T_{hist}+t}}$ is the optimal rate assigned to well x at the t^{th} optimization interval, which is of length ΔT_t.

A boxplot of q_x^* for the set of centroid realizations may be used to represent the uncertainty in the optimal schedule. Moreover, the average rate change may be computed as

$$\Delta q_x^* = \left(q_x^* - q_x^{ref}\right) \qquad (2)$$

A bubble plot of this quantity is used in this work to demonstrate the spatial uncertainty in optimal rates.

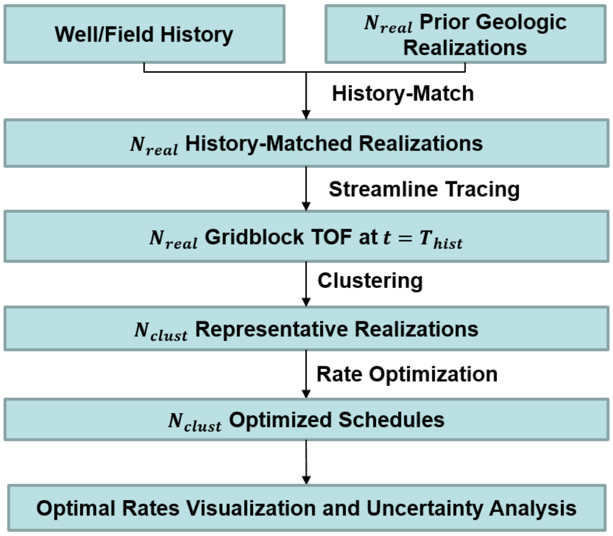


Figure 57 Proposed workflow for waterflood rate optimization under geologic uncertainty

History Matching

History matching refers to a model calibration technique that involves simulating the historical reservoir response and comparing these with field observations. This is essentially an inverse problem wherein we try to update a geologic model by seeking to match its responses with observed data. Typically, the discrepancies between the various responses and observations are evaluated as an objective function, which is to be minimized. Minimization of this objective function proceeds by iterative updates to the model, which may be computed via a wide variety of algorithms. It is to be noted that history-matching is typically an extremely underdetermined problem, which may yield a large number of equally satisfactory reservoir descriptions (which fit observed data). Typically used algorithms for history-matching include variants of evolutionary algorithms such as multi-objective genetic algorithm, differential evolution (Hajizadeh et al. 2009) and particle swarm optimization (Lee and Karl 2019). Other history matching techniques include gradient-based algorithms such as multi-scale history-matching using streamlines (Chen et al. 2020a; Chen et al. 2020b; Liu et al. 2019), and adjoint method (Chen et al.1974; Li et al. 2003). A number of ensemble-based history-matching technique have gained popularity in recent years that include ESMDA (Emerick and Reynolds 2013) and ensemble-based non-linear orthogonal matching pursuit algorithm (Elsheikh et al. 2013). In this work, we use ESMDA for generating the history-matched realizations, considering its ease of implementation and significantly faster convergence than most evolutionary algorithms.

Clustering Realizations

In this work, we use clustering to reduce the number of realizations over which optimization has to be carried out, while accounting for the full range of geologic uncertainty. In order to do so, we have to form feature vectors that parameterize the geologic uncertainty in a way that retains all relevant information which may affect the results of optimization. One way to do this is to parameterize each geologic field (such as permeability, porosity etc.) separately and use all of these features as input into the clustering algorithm. However, such an approach would require a large number of feature vectors in case of geologic uncertainty in multiple parameters. We propose to perform clustering on the time-of-flight field, which is calculated from the flux field at the end of history, which, in turn encompasses the effects of all model properties – both certain and uncertain. The computation of the gridblock-wise TOF was computed using the Pollock algorithm implementation in the in-house software 'Destiny'. Subsequently, the gridblock TOF field was converted into a gridblock TOF rank field. In this way, the gridblock with the highest value of TOF will be given a rank of 1 whereas the one with the lowest TOF will be ranked N_{grid} where N_{grid} represents the total number of gridblocks in the model. Ranking was done since raw gridblock TOF may assume very high magnitudes in cells which are close to stagnation points, and this may degrade the performance of the clustering algorithm.

Clustering was performed using k-means algorithm which proceeds by computing the squared distance between each realization and a randomly chosen set of N_{clust}

‘centroids’, which are assumed to representative of all realizations belonging to each cluster. Each realization is assigned the cluster with the closest centroid. The ‘centroids’ are then updated by recalculating the mean of the realizations within the cluster. In effect, the k-means algorithm partitions the dataset into N_{clust} clusters such that the within-cluster variance (referred to as ‘inertia’) is minimized. Mathematically, the problem is expressed as follows. Given a set of data points $x_1, x_2, \ldots, x_{N_{real}}$ we are to find the set of clusters $\mathbb{C}^* = \{C_1, C_2, \ldots, C_k\}$ such that:

$$\mathbb{C}^* = \arg\min_{\mathbb{C}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{3}$$

Where μ_i is the centroid of cluster C_i.

A major shortcoming of k-means is the requirement that the user specifies N_{clust}. Typically, this is done by repeatedly running k-means with varying $N_{clust}$and plotting the resulting inertia. Subsequently we choose a value for N_{clust} such that any increase from this value results in minimal decrease in inertia.

Rate Optimization

Rate optimization refers to the computation of well rates that maximizes a specified reward function, which may be the cumulative oil production, NPV or waterflood sweep efficiency. To summarize these, Asheim (1988) developed an optimization that was coupled with a reservoir simulator, which was used to maximize net present value (NPV) subject to constraints on reservoir flow dynamics, total flow capacity and well constraints. Brouwer et al. (2001) performed rate optimization by reducing the distribution in the arrival times of the flood front for a time-independent flow field. Alhuthali et al. (2007) proposed a methodology that involves equalizing the arrival time of the waterfront at producers in separate subregions of the reservoirs. The arrival time was calculated by running a reservoir simulator and tracing streamlines explicitly. A machine-learning based workflow that builds upon Alhuthali's methodology was introduced by Sen et al (2020). Chen et al. (2020b) proposed a streamline-based gradient-free method for maximizing oil production at the end of production time under constraints such as fieldwide injection and production rates, minimum well producing pressure and allowable ranges on well rates. This algorithm was implemented on a real field (Mangala, India) under polymer flooding, demonstrating the efficacy of the algorithm (Chen et al. 2020).

Owing to its computational efficiency and proven results on real-field settings, Chen's algorithm was used to perform the rate optimization for the realization subset generated by clustering the flow field. The objective of this algorithm is to minimize the overall water cut within the optimization period under constraints such as field injection rate target, field production rate target, pressure constraints, rate and rate change limits. This is done by computing the well-pair efficiency, which is the ratio of oil volumes produced per water injected.

$$e_{ip} = \frac{OR_{ip}}{q_{ip}\Delta t} \tag{4}$$

Here OR_{ip} refers to the volumes of mobile oil within the streamtube bundle starting from injector i and ending at producer p, and q_{ip} is the streamline-based injection allocation from i to p. The duration of the optimization window is denoted by Δt.

Subsequently the field-level efficiency is also calculated in a similar manner, by taking the ratio of the mobile oil in all streamtube bundles to the total injection allocation over all bundles (throughout the optimization interval).

$$e_{ip} = \frac{\Sigma\, OR_{ip}}{\Sigma\, q_{ip}\Delta t} \tag{5}$$

The injection and production rates are reallocated in such a way that well pairs that perform worse than the field average, are allocated lesser volumes, as per:

$$\hat{q}_i = \Sigma_p \lambda_{ip} q_{ip} \tag{6}$$

$$\hat{q}_p = \Sigma_i \lambda_{ip} q_{ip} \tag{7}$$

Where $\hat{q}_i$ is the updated injection rate of injector i and $\hat{q}_p$ is the updated production rate of producer p.

Finally, the updated rates are rescaled to fulfill field-level rate constraints.

$$q_i^{update} = \frac{\Sigma_i q_i}{\Sigma_i \hat{q}_i} \hat{q}_i \tag{8}$$

$$q_p^{update} = \frac{\Sigma_p q_p}{\Sigma_p \hat{q}_p} \hat{q}_p \tag{9}$$

This process is carried out iteratively over each optimization window. For a detailed description of the algorithm, the reader is referred to Chen et al. (2020b).

Synthetic Field Application: SAIGUP model

The SAIGUP model stands for Sensitivity Analysis of the Impact of Geological Uncertainties on Production forecasting in clastic hydrocarbon reservoirs. In this application, we use the upscaled model of dimensions $40 \times 120 \times 20$, with each gridblock of areal dimensions $75ft \times 75ft$. The model is highly faulted with close to 20,000 non-neighbor connections. We consider a pattern flooding with 6 injectors and 6 producers. The initial setup is shown in Figure 58.

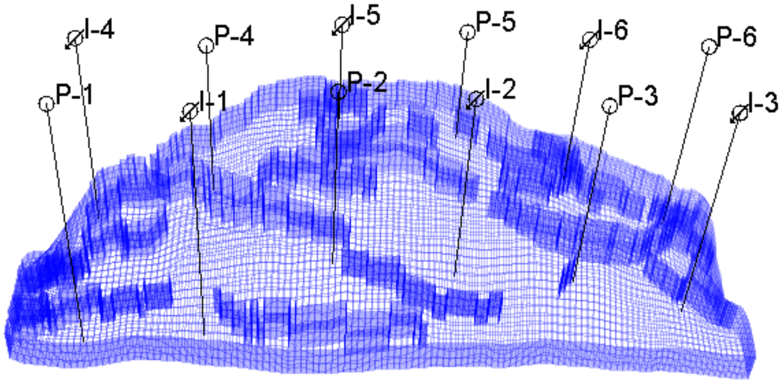


Figure 58 SAIGUP faulted grid (Matthews et al. 2008) showing well configuration for pattern flooding

Ensemble Generation using ESMDA

The observed data, is given in Figure 59 (red background) and was generated by simulating production from the true realization (Figure 60 (a)). The observation spans 210 days and as with the 2D case, it consists of the following measurements, recorded every 30 days:

a. Field-wide liquid production rates in STB/d

b. Field-wide water injection rates in BBL/d

c. Well BHPs for all injectors and producers, in psia.

d. Oil production rates in STB/d for all producers

e. Water production rates in BBL/d for all producers

f. Water injection rates in BBL/d for all injectors

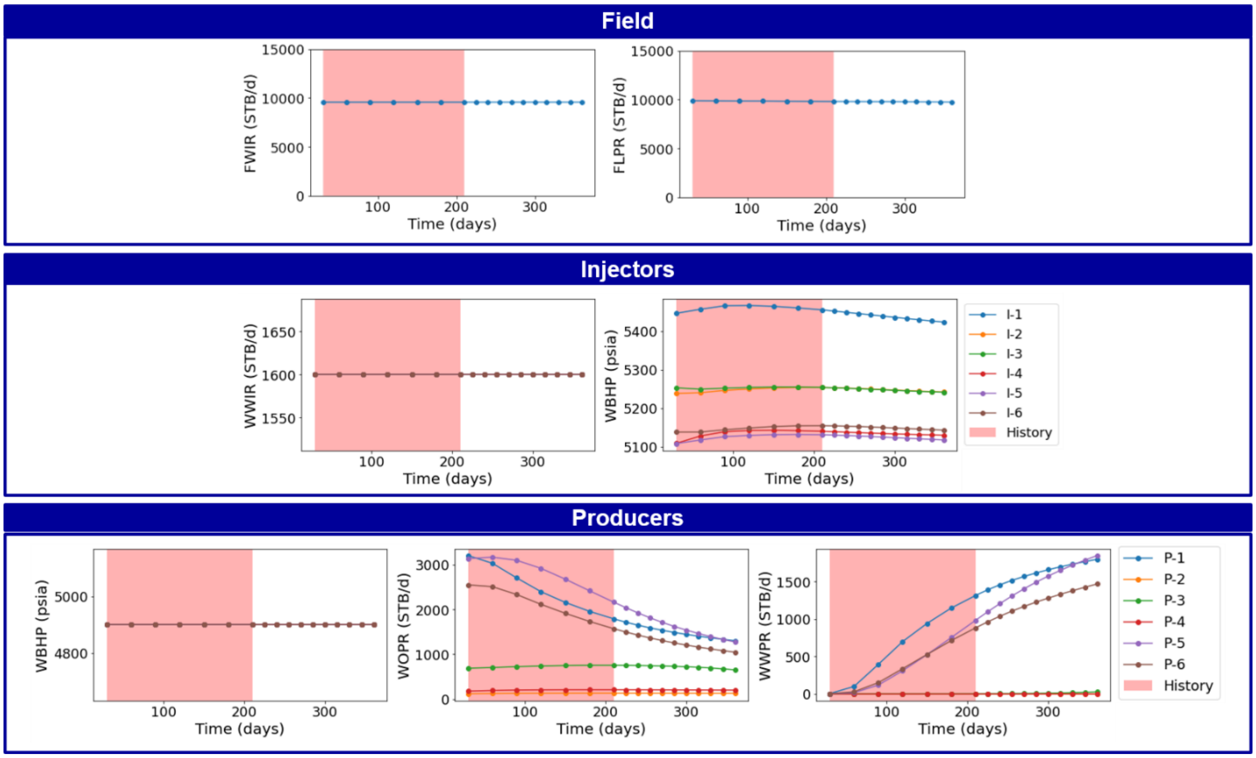


Figure 59 Observed data for SAIGUP case

In order to perform the ensemble-based history match, we start with a prior ensemble of 200 permeability fields, a few of which are shown in Figure 60(c-d). The ensemble mean of the prior is shown in Figure 60(b). Unlike the 2D case, we have some degree of prior knowledge on the permeability field in the regions close to the wells, as is typically the case in a real setting. For comparison purposes, the true permeability field, wherefrom the observed data was generated in given in Figure 60 (a). ESMDA was performed with 5 assimilation steps, wherein the data misfits where calculated as the difference between the observed data and the corresponding responses of each ensemble member. The well and field responses of the prior models and the resulting posteriors (after 5 assimilations) are shown in Figure 61 as grey and blue plots respectively. The posterior responses are seen to converge towards the observed data with reduced variance than the prior responses. A few of the resulting posterior permeability fields (obtained from the priors in Figure 60(c-e)) are shown in Figure 60(g-i). The ensemble mean of the posterior, shown in Figure 60(f), exhibits significant permeability trends which are also present in the true permeability field (Figure 60 (a)).

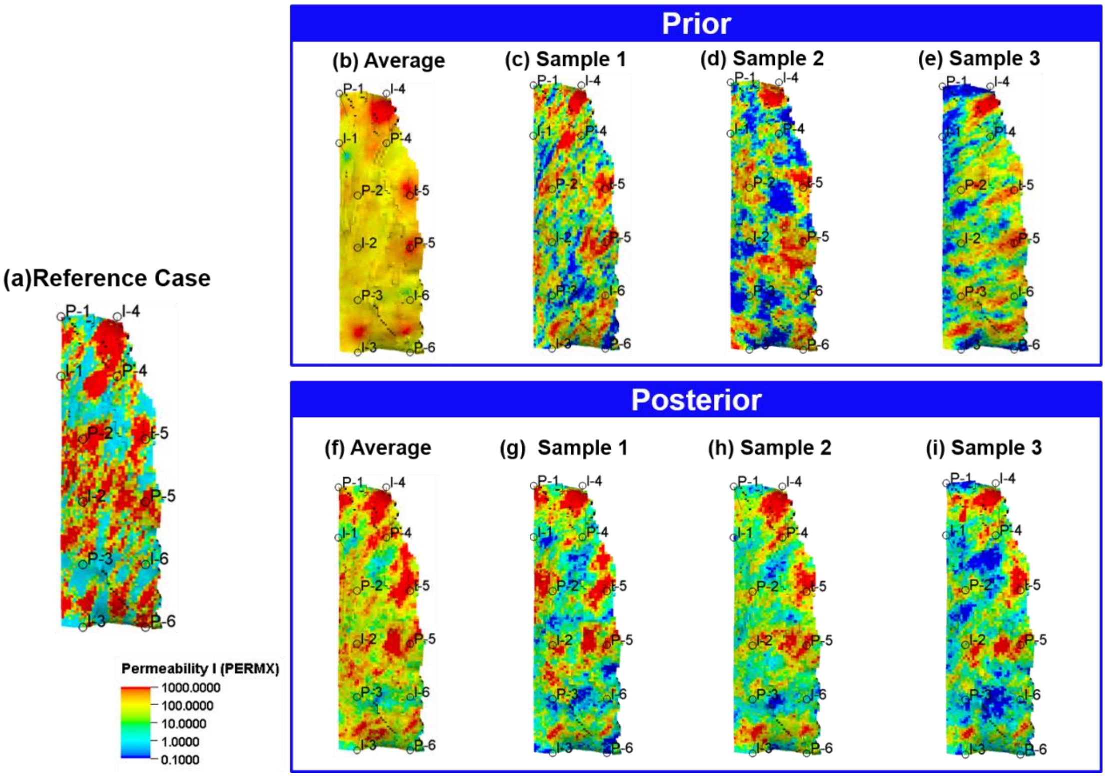


Figure 60 A comparison of prior and posterior permeability field, the latter of which is obtained using ESMDA with 5 assimilations.

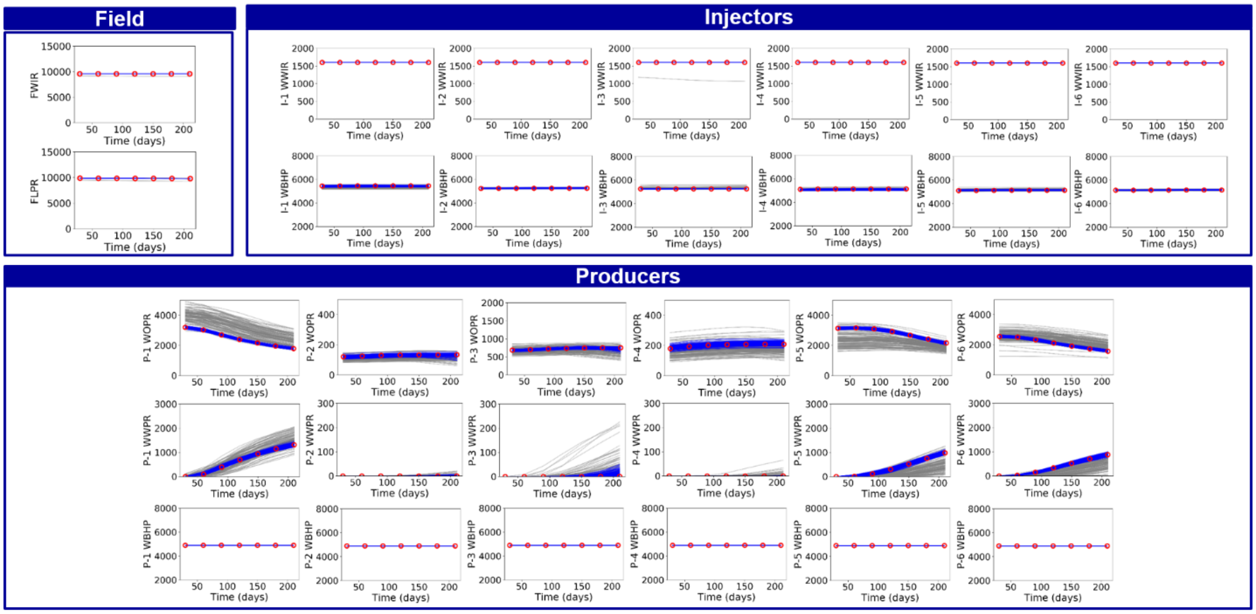


Figure 61 Field and well responses of prior and posterior SAIGUP models illustrating the history-match process via ESMDA

Clustering

Once we have a set of 200 history-matched realizations, the next step is to group these into a number of clusters (N_{clust}) based on an appropriate feature, prior to rate optimization. In the 2D case, the gridwise TOF field was ranked and parameterized to generate feature vectors, based on which, we performed clustering using k-means clustering. For the SAIGUP model, with 96000 gridblocks, the direct application of this approach was found to yield an unwieldly number of clusters. This would later on require us to perform optimization on too many representative realizations.

In order to avoid this problem, we averaged the gridblock-wise TOF field (at the end of history) along the z direction, so that we have a 2D map of averaged TOF (with

4800 gridblocks). Subsequently, these were ranked and parameterized, as in the 2D case, and feature vectors were generated for clustering.

K-means was used for clustering, wherein N_{clust} was chosen to be 15, based on the inertia vs. N_{clust} plot shown in Figure 62. Having a 2D map also renders the visualization of clustering results feasible, as apparent from a few selected results from clustering, shown in Figure 63.

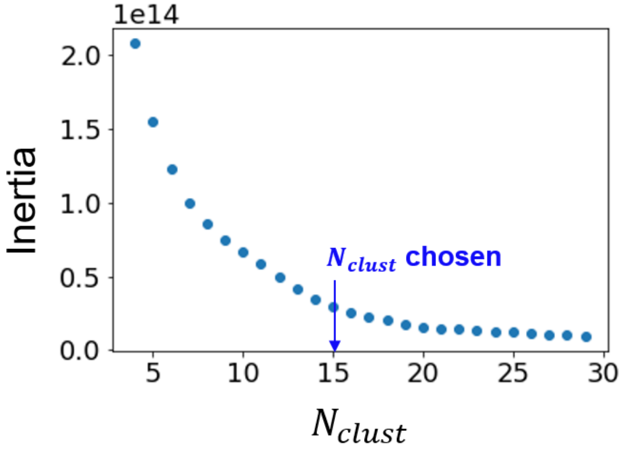


Figure 62 While using the k-means algorithm for clustering, the choice of an appropriate N_{clust} has to be made by plotting the reduction in inertia against increasing N_{clust}. For the SAIGUP case, we choose $N_{clust} = 15$ since the inertia almost levels off around this value.

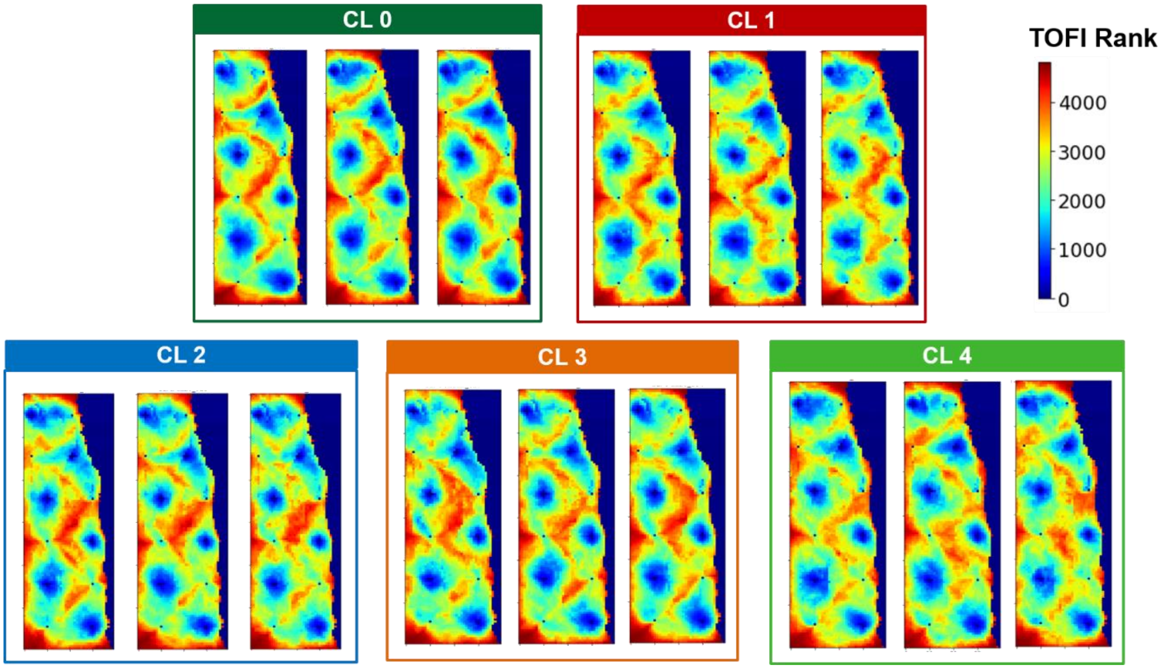


Figure 63 A few selected results from clustering the vertically averaged ranked grid-wise TOF. Each box corresponds to a particular cluster, and a few of its member ranked TOF fields are displayed in each box.

Rate Optimization

As with the 2D case, the reference (unoptimized) case is the 'do-nothing' case wherein all the wells (in all realizations) are rate-constraint at their respective values at the end of history. The resulting field-wide water injection, oil production and water production for the ensemble at reference case are given in Figure 64. The region in pink background represents the duration of history, wherein the uncertainty in responses have been minimized using ESMDA. The white region represents the optimization period, where there is considerable uncertainty in oil and water production, under constant field injection and liquid production rates.

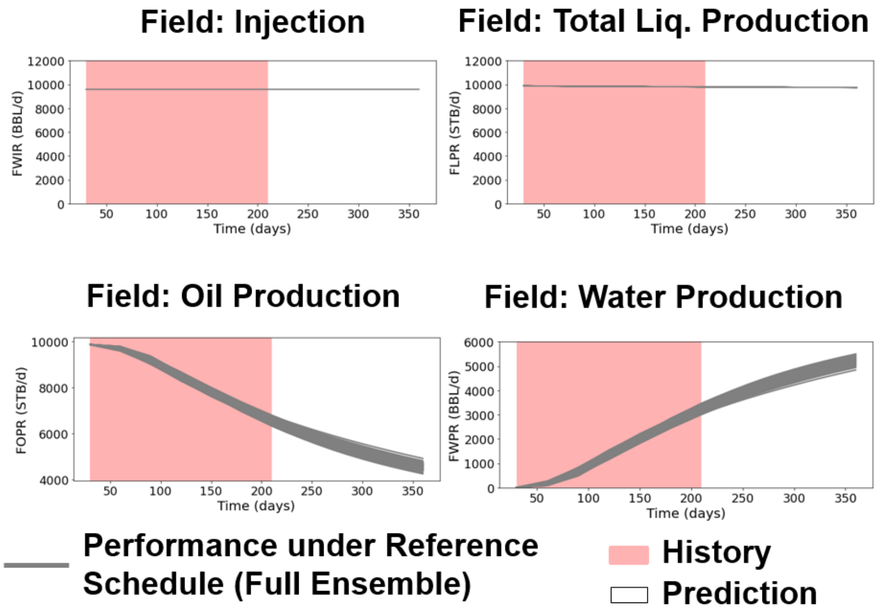
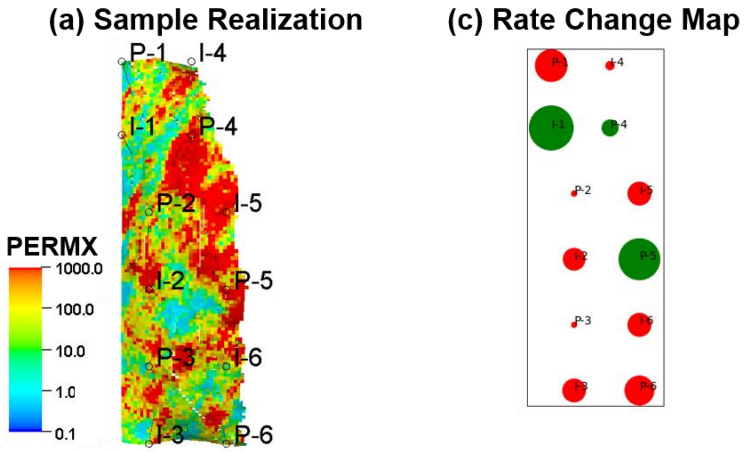


Figure 64 Reference case for optimization for field-scale SAIGUP, wherein all wells are constrained at the end-of-history rates

The subsequent streamline-based optimization is performed for a total of 150 days at 15 day intervals under constrained field liquid production and field water injection rates. In order to demonstrate the improvement in oil recovery brought about by optimization, the values of these constraints are set to the average field liquid production and average field water injection of the reference case in the optimization period (white region in Figure 64). Furthermore, secondary BHP constraints on both injectors (maximum BHP at 7000 psia) and producers (minimum BHP at 5000 psia) are carried over from history.

Optimization on a Single Realization

Streamline-based optimization is demonstrated on a single realization shown in Figure 65(a). The historical field oil production is plotted in Figure 65 (d) (red region). The pore-volume weighted summation of the end-of-history oil saturation is shown in Figure 65(b). On performing rate optimization, the prescribed rate changes for each well (averaged across the optimization period) is indicated by the green and red circles in Figure 65(c), where the green (and red) circles denotes an increased (and decreased) rate allocation. The size of the circles is indicative of the magnitude of rate change.



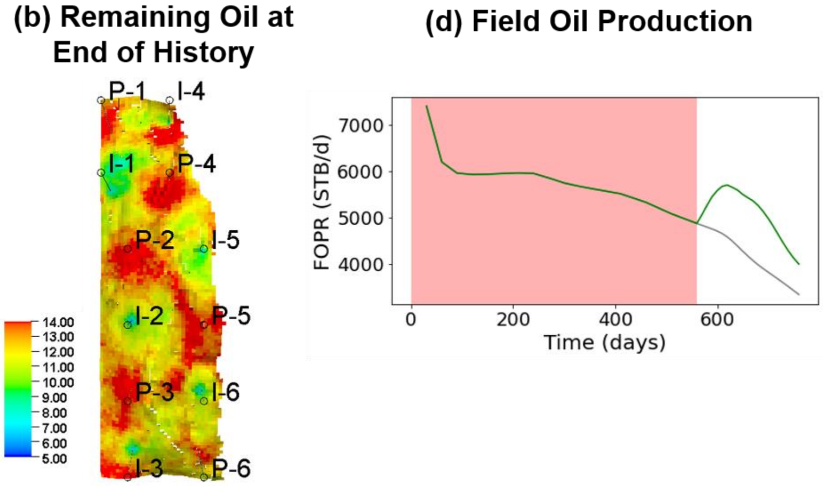


Figure 65 Application of streamline-based rate optimization on a sample realization shown in (a) The weighted sum of oil saturation map (in the vertical direction) at the end of history is shown in (b) and the rate changes prescribed by the optimization algorithm, averaged across the entire optimization period is shown in (c) wherein green and red circles denote a positive and negative rate change respectively. The size of the circle is proportional to the magnitude of the prescribed rate change. The resulting improvement in field-wide oil production is illustrated in (d) by the green curve (optimized case) as compared to the grey curve (reference case).

The re-distribution of streamlines (and hence the sweep improvement) that results from the rate changes in Figure 65(c), can be seen in Figure 66. Furthermore, the PV-weighted summation of remaining oil at the end of the optimization period (for both the reference and the optimized cases) is also given. The reduction in remaining oil after optimization, especially in the circles regions, demonstrates the increased sweep efficiency brought about by optimization.

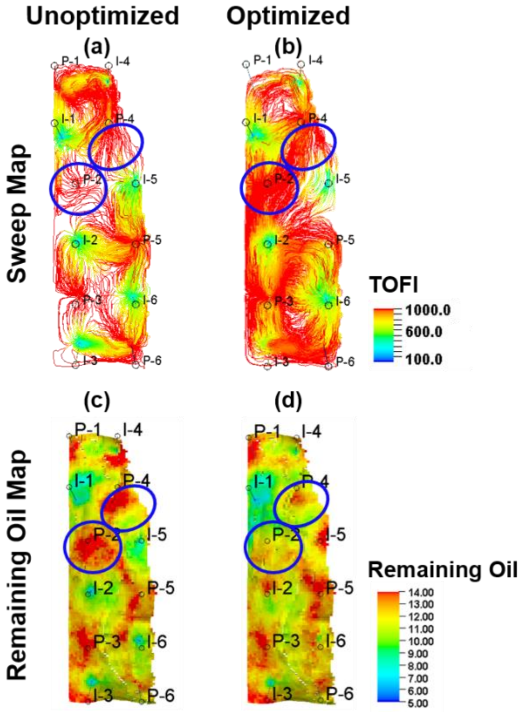


Figure 66 The sweep of the waterflood (a) before and (b) after optimization may be visualized by plotting the streamlines, along with their TOF from injector. Optimization facilitates a redistribution of streamlines, resulting in a more efficient sweep pattern that in turn leads to more recovered oil. The increase in oil recovery can be seen in the remaining oil map at the end of the optimization period in the reference (c) and optimized (d) cases. The blue circles indicate the regions where increased sweep (due to optimization) leads to less remaining oil.

Rate Optimization over Multiple Realizations

The rate optimization was performed on the $N_{clust} = 15$ centroid realizations, obtained by performing k-means clustering on the TOF field at the end of history. The resulting optimized schedules for each centroid were weighted in proportional to its cluster membership. Subsequently, the q_x^* and Δq_x^* were computed for each optimal schedule and their distributions were generated by weighting each q_x^* (and Δq_x^*) by the corresponding number of realizations in the cluster (cluster membership). The resulting spatial rate change map and distribution boxplots are given in Figure 67(a) and (b).

In Figure 67 (a), the size of the bubbles indicate the magnitude of the median Δq_x^* (over the weighted centroid set) and the red/green colors correspond represent negative/positive rate changes. Moreover, the transparency of the bubbles are indicative of the relative uncertainty of the rate change of each schedule, expressed as the ratio between the standard deviation and the median of rate change. For instance, a well with a small value of median rate change and a relatively large value of standard deviation, such as I-1, is marked by increased transparency in Figure 67 (a).

Figure 67 (b) shows the distribution of q_x^* for each well in the form of boxplots. The rates assigned to each well is also marked by a red dot on the boxplot. One can see from the boxplots that rate change may be implemented on most producers with a high degree of certainty, such as P-1, P-3, P-4, P-5 and P-6. Broadly speaking, production must be re-allocated from P-1, P-5 and P-6 to primarily P-3 and to a lesser extent, P-4 and P-2.

On the contrary, there is a high degree of uncertainty surrounding rate changes to all the injectors, with the exception of I-5, as per Figure 67 (a).

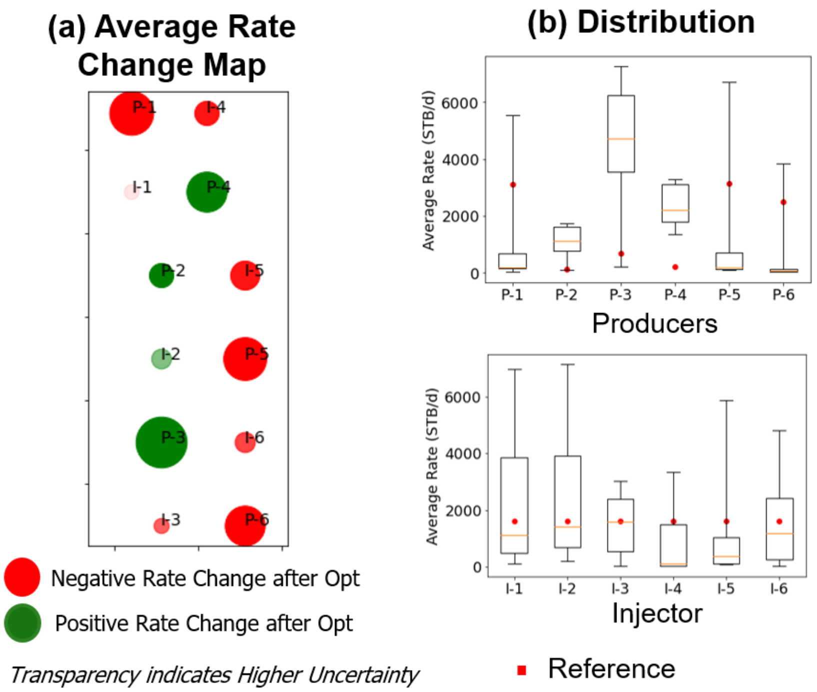


Figure 67 (a) The bubble plot showing spatial distribution of the average rate change (Δq_x^*) for each well on running the optimization on all centroid realizations individually. (b) The boxplots shows the distributions of average rate, q_x^*, for each well (obtained on optimizing each of the centroids). The rate assigned to the respective wells in the reference case (q_x^{ref}) is shown by the red dot.

The resulting field responses obtained by optimizing each of the centroid realization are shown in Figure 68. The reference responses are shown in grey whereas the optimized field-level injection/production rates are given in blue/green. For similar field injection and total liquid production rates (Figure 68 (a, b)), there is significant improvement in oil production (Figure 68 (c)) for all the centroid realizations, accompanied by reduction in water production (Figure 68 (d)).

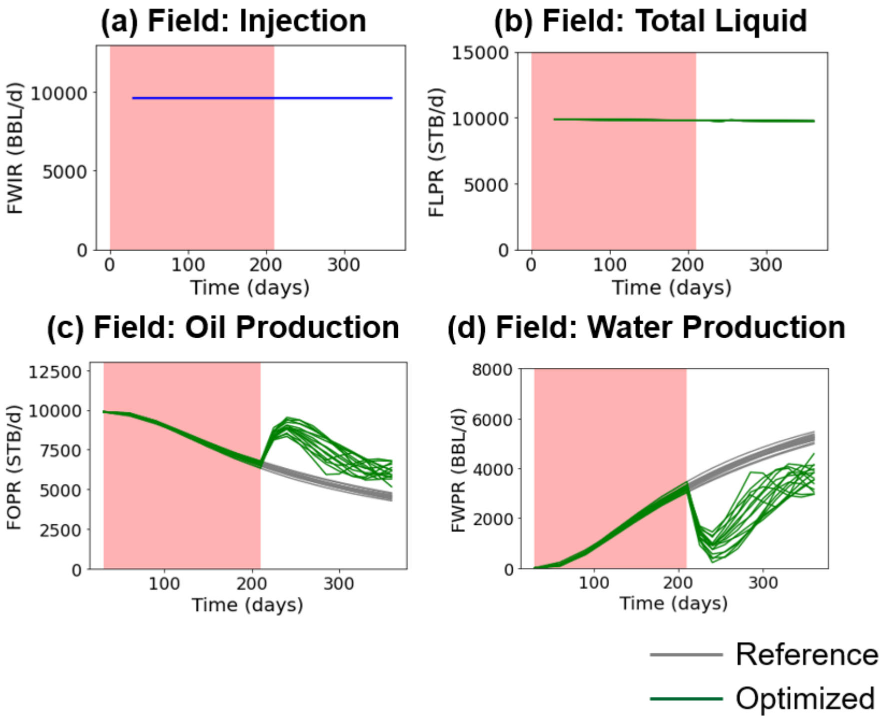


Figure 68 The field response plots demonstrates that, under similar (a) field injection and (b) field total liquid production rates, there is significant improvement in (c) field oil production and (d) reduced water production post optimization (shown in green), as compared to the reference case (shown in grey).

In order to validate this approach, we need to compare the q_x^* distributions obtained using the centroids, with those obtained using the full ensemble. This is illustrated in Figure 69 and Figure 70. Figure 69 (a) shows the rate change (Δq_x^*) obtained using only the 15 centroids whereas Figure 69 (b) shows the same obtained using the full ensemble of 200 realizations. A comparison between these demonstrate the high accordance between the centroid-based and full ensemble-based Δq_x^*.

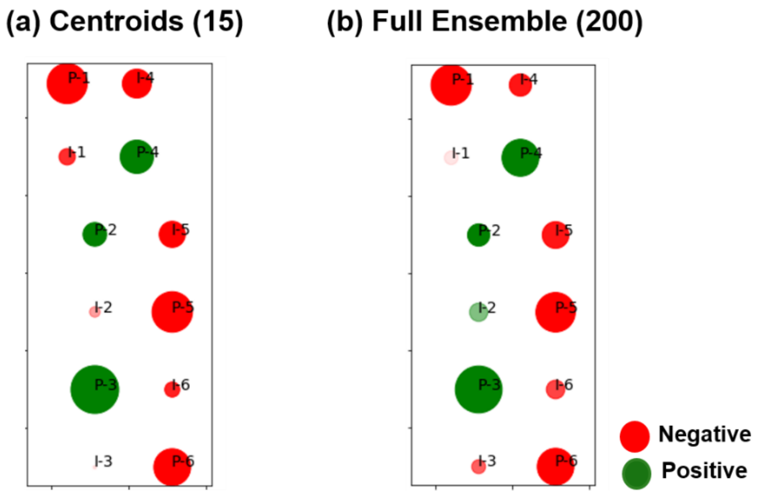


Figure 69 The spatial distribution of the rate changes (optimized rate minus reference rate) obtained on performing optimization on (a) solely the centroids set and (b) the full ensemble of the SAIGUP realizations. The size of the bubbles corresponds to the median of Δq_x^* over the realizations and the transparency corresponds to the standard deviation.

Similarly, the boxplots in Figure 70 (a) and (c) represent the distributions of average rate change q_x^* for each well in the model, as obtained by rate-optimizing the 15 centroid realizations (taking into account their respective cluster membership). Figure 70 (b) and (d) show the distributions obtained by rate optimizing the full ensemble of 200 realizations. These boxplots also demonstrate a high accordance between the centroid-based and full ensemble-based analyses, thus validating our proposed workflow.

Moreover, we were able to perform the analysis by running the optimization on solely the cluster centroids as opposed to the full ensemble, thereby reduce the time-cost by a factor of $\left(1 - \frac{N_{clust}}{N_{real}}\right)$. For this particular case, where $N_{real} = 200$ and $N_{clust} = 15$, the reduction in optimization time was ~93%, while preserving the uncertainty information.

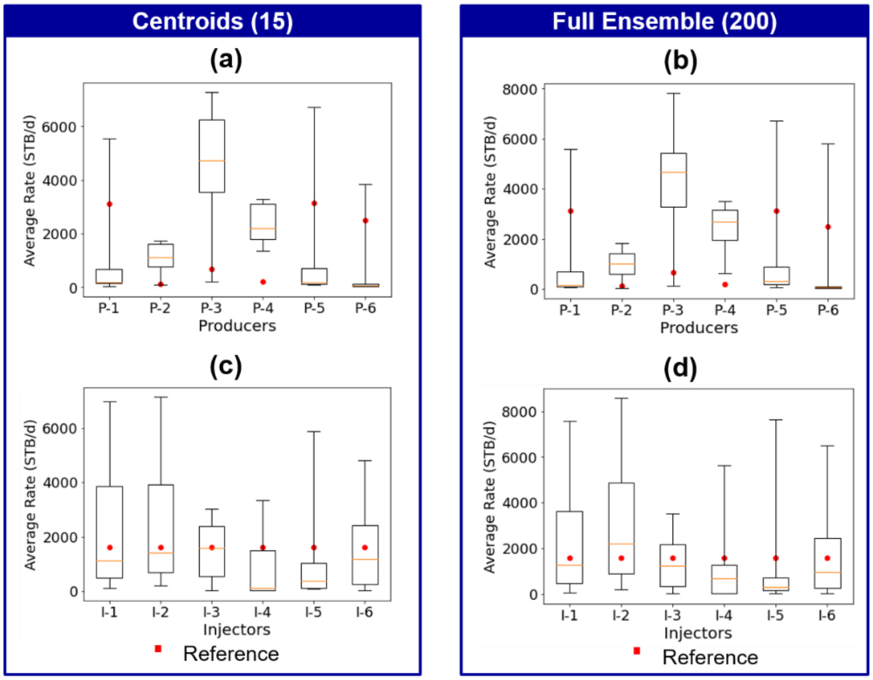


Figure 70 A comparison of the distributions of average optimized rates for each well obtained by using solely the centroid sets (a,c) and those obtained by using the full ensemble (b,d) for the SAIGUP case. The red dots represent the reference rates (q_x^{ref}).

Conclusions

This chapter describes a comprehensive workflow starting with the generation of history-matched geologic realizations, given a set of observed data. Subsequently the k-means algorithm was used to cluster the time-of-flight field at the end of history, to group the realizations into N_{clust} sets, each with its representative realization (centroid). In this manner, the geologic uncertainty described by N_{real} ($N_{real} = 240$ for 2D case and $N_{real} = 200$ for SAIGUP case) realizations were represented using merely $N_{clust} = 15$ centroid realizations. Rate optimization was performed solely on the centroid realizations and the distribution of rate changes assigned to each well was shown to be similar to the values obtained by optimizing the entire ensemble. The uncertainty analysis generated using the TOF-based centroid realizations may be used to inform the selection of key wells for implementation of rate change.

To summarize, the key contributions of this chapter are listed below:

- TOF field is used to cluster an ensemble of history-matched realizations, of a 2D heterogeneous field as well as the field-scale 3D SAIGUP model.
- The set of centroids (15 realizations) is used to represent the geologic uncertainty of the full ensemble (200 realizations in case of SAIGUP model).
- Optimization on the centroid set and that on the full ensemble show similar uncertainty in each well's optimal schedule.
- The resulting uncertainty analysis for individual wells can be used to advise the selection of wells for implementing optimal rates.

CHAPTER V

CONCLUSIONS

This study explored various applications of machine learning in inferring the inter-well connectivity in a field under water flood and CO_2 WAG injection via proxy models such as neural networks and statistical recurrent units. The workflow was applied in the context of rate optimization under geologic uncertainty. A chapter-wise summary follows:

In Chapter II, a neural network based proxy model was introduced that can infer the reservoir connectivity in terms of the average time-of-flight of the fastest 20% streamlines, using parameterized forms of reservoir properties (such as permeability and porosity) and well rates. The proxy model was seen to be faster than a conventional simulator by around 3 orders of magnitude. This computational efficiency made the model suitable to be used in a rate optimization framework considering geologic uncertainty. The data-driven rate optimization workflow was applied successfully to a 2D synthetic case as well as the field-scale Brugge model.

In Chapter III, a purely data-driven workflow for inferring interwell connectivity was developed using statistical recurrent units. An SRU-based model was fitted to field observations such as gas injection rates, gas production rates and well pressure data. After obtaining a sufficiently predictive model, the contribution of each injector to the prediction of gas production at each producer was quantified using permutation variable importance. The connectivity map from SRU was compared with the average streamline-based connectivity and the former was demonstrated to be highly similar to the latter. The workflow was applied to field-scale model under CO_2 WAG injection with frequent shut-ins, which are typical of real field operations. The connectivity map obtained from the proposed methodology was seen to be of high accordance with the time-averaged streamline-based connectivity.

Chapter IV described a comprehensive workflow for identifying key wells for implementing waterflood optimization, considering geologic uncertainty. The workflow started with the generation of history-matched geologic realizations, given a set of observed data. Subsequently the k-means algorithm was used to cluster the time-of-flight field at the end of history, to group the realizations into N_{clust} sets, each with its representative realization (centroid). In this manner, the geologic uncertainty described by N_{real} ($N_{real} = 200$ for SAIGUP case) realizations were represented using merely $N_{clust} = 15$ centroid realizations. Rate optimization was performed solely on the centroid realizations and the distribution of rate changes assigned to each well was shown to be similar to the values obtained by optimizing the entire ensemble. The uncertainty analysis generated using the TOF-based centroid realizations was used to inform the selection of key wells for implementation of rate change.

Recommendations for Future Work

The work presented in this dissertation may be continued in a number of directions. A few of these are listed below:

- The proxy model based approach to rate optimization presented in Chapter II may be extended to perform rate optimization over time intervals. In order to do so, one way would be to use a recurrent neural network based model that can deal with time series data. Another possible direction may be to parameterize the temporal changes in connectivity and use these are output variables.
- A logical extension to the SRU-based study is the inference of production and injection allocation volumes, using the variable importance. Such an extension was briefly introduced in this dissertation, and its applicability to various scenarios may be investigated in future.
- Since the SRU-based variable importance directly provides the connectivity map, one may use the SRU model for performing rate optimization during the WAG process. Alternatively, the SRU gas predictions may be used under a rate optimization framework, with the objective of minimizing gas production, thereby maximizing CO_2 storage within the reservoir.
- Further study may be conducted to compare various types of clustering algorithms in the rate optimization workflow introduced in Chapter IV. In this dissertation, k-means algorithm was used to perform flow feature clustering. The performance of the workflow may be greatly increased by using better types of clustering algorithms such as affinity propagation. For instance, in contrast to k-means, affinity propagation automatically selects the optimal number of clusters, eliminating the need to use the elbow method for choosing the best N_{clust}.

REFERENCES

Abadpour, Anahita , Adejare, Moyosore , Chugunova, Tatiana , Mathieu, Helene , and Norman Haller. "Integrated Geo-Modeling and Ensemble History Matching of Complex Fractured Carbonate and Deep Offshore Turbidite Fields, Generation of Several Geologically Coherent Solutions Using Ensemble Methods." Paper presented at the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, November 2018. doi: https://doi.org/10.2118/193028-MS

Achen, Christopher H. 1982. Interpreting and using regression, Vol. 29: Sage.

Alakeely, Abdullah and Horne, Roland N. 2020. Simulating the Behavior of Reservoirs with Convolutional and Recurrent Neural Networks. SPE Reservoir Evaluation & Engineering 23 (03): 0992-1005. https://doi.org/10.2118/201193-PA.

Albertoni, A., & Lake, L. W. (2003). Inferring interwell connectivity only from well-rate fluctuations in waterfloods. SPE reservoir evaluation & engineering, 6(01), 6-16.

Alhuthali, Ahmed Humaid H., Datta-Gupta, Akhil, Yuen, Bevan Bun Wo et al. 2008. Optimal Rate Control Under Geologic Uncertainty. Presented at the SPE Symposium on Improved Oil Recovery, Tulsa, Oklahoma, USA. 2008/1/1/. https://doi.org/10.2118/113628-MS.

Alhuthali, Ahmed, Oyerinde, Adedayo, and Datta-Gupta, Akhil. 2007. Optimal Waterflood Management Using Rate Control. SPE Reservoir Evaluation & Engineering 10 (05): 539-551. https://doi.org/10.2118/102478-PA.

Asheim, H. 1988. Maximization of Water Sweep Efficiency by Controlling Production and Injection Rates. Presented at the European Petroleum Conference, London, United Kingdom. 1988/1/1/. https://doi.org/10.2118/18365-MS.

Balaji, Karthik, Suhag, Anuj, Ranjith, Rahul et al. 2017. Optimization of Recovery in Waterfloods with Bang-Bang Control in Reservoirs with Subsidence and Uplift. Presented at the SPE Western Regional Meeting, Bakersfield, California. 2017/4/23/. https://doi.org/10.2118/185727-MS.

Bao, A., Gildin, E., Huang, J., Coutinho, E. J. R. et al. (2020). Data-Driven End-To-End Production Prediction of Oil Reservoirs by EnKF-Enhanced Recurrent Neural Networks. Proc., SPE Latin American and Caribbean Petroleum Engineering Conference. https://doi.org/10.2118/199005-MS.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In Noise reduction in speech processing (pp. 1-4). Springer, Berlin, Heidelberg.

Bhark, Eric W, Jafarpour, Behnam, and Datta-Gupta, Akhil. 2011. A generalized grid connectivity–based parameterization for subsurface flow model calibration. Water Resources Research 47 (6).

Breiman, L. (2001). Random forests. Machine learning. 45 (1): 5-32.
Bring, J. (1996). A geometric approach to compare variables in a regression model. The American Statistician. 50 (1): 57-62.

Brouwer, D. R. and Jansen, J. D. 2002. Dynamic Optimization of Water Flooding with Smart Wells Using Optimal Control Theory. Presented at the European Petroleum Conference, Aberdeen, United Kingdom. 2002/1/1/. https://doi.org/10.2118/78278-MS.

Brouwer, D. R., Jansen, J. D., van der Starre, S. et al. 2001. Recovery Increase through Water Flooding with Smart Well Technology. Presented at the SPE European Formation Damage Conference, The Hague, Netherlands. 2001/1/1/. https://doi.org/10.2118/68979-MS.

Budescu, D. V. (1993). Dominance analysis: a new approach to the problem of relative importance of predictors in multiple regression. Psychological bulletin 114 (3): 542.

Cao, Fei, Luo, Haishan, and Lake, Larry W. 2014. Development of a Fully Coupled Two-phase Flow Based Capacitance Resistance Model (CRM). Presented at the SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, USA. 2014/4/12/. https://doi.org/10.2118/169485-MS.

Chen, Chaohui, Wang, Yudou, Li, Gaoming et al. 2010. Closed-loop reservoir management on the Brugge test case. Computational Geosciences 14 (4): 691-703.

Chen, H., Yang, C., Datta-Gupta, A., Zhang, J., Chen, L., Liu, L., … Bahar, A. (2020a). Fracture Inference and Optimal Well Placement Using a Multiscale History Matching in a HPHT Tight Gas Reservoir, Tarim Basin, China. Upstream Oil and Gas Technology, Volume 2, 2020. doi：10.1016/j.upstre.2020.100002.

Chen, H., Park, J., Datta-Gupta, A., Shekhar, S., Grover, K., Das, J., ... & Chitale, A. (2020b, August). Improving Polymerflood Performance Via Streamline-Based Rate Optimization: Mangala Field, India. In SPE Improved Oil Recovery Conference. Society of Petroleum Engineers.

Chen, H., Yao, C., Datta-Gupta, A., et al. (2020c, October). Identification of Fractures and Preferential Flow Paths Using Streamlines and Dynamic Data in Dual Porosity Dual Permeability Reservoir Models. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Chen, W. H., Gavalas, G. R., Seinfeld, J. H., & Wasserman, M. L. (1974, December 1). A New Algorithm for Automatic History Matching. Society of Petroleum Engineers. doi:10.2118/4545-PA

Chollet, Francois. 2015. Keras. GitHub. https://github.com/fchollet/keras.

Dorn, Oliver, and Rossmary Villegas. "History matching of petroleum reservoirs using a level set technique." Inverse problems 24.3 (2008): 035015.

Chung, J., Gulcehre, C., Cho, K. et al. (2015). Gated feedback recurrent neural networks. In International conference on machine learning (pp. 2067-2075). PMLR.

Cordes, C., & Kinzelbach, W. (1992). Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. Water resources research, 28(11), 2903-2911.

Coulibaly, P. and Baldwin, C.K. (2005). Nonstationary hydrological time series forecasting using nonlinear dynamic methods. Journal of Hydrology 307 (1-4): 164-174.

de Holanda, R.W., Gildin, E., & Jensen, J.L. (2018). A generalized framework for Capacitance Resistance Models and a comparison with streamline allocation factors. Journal of Petroleum Science and Engineering. 162: 260-282.

Datta-Gupta, A. & King, M.J. (2007). Streamline simulation: theory and practice, Vol. 11: Society of Petroleum Engineers Richardson.

Dunis, C.L. and Huang, X. (2002). Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. Journal of Forecasting. 21 (5): 317-354.

Dutta, K. & Sarma, K.K. (2012). Multiple feature extraction for RNN-based Assamese speech recognition for speech to text conversion application. In 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS) (pp. 600-603). IEEE.

Elsheikh, Ahmed H., Wheeler, Mary F., and Ibrahim Hoteit. "An Ensemble Based Nonlinear Orthogonal Matching Pursuit Algorithm for Sparse History Matching of Reservoir Models." Paper presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, February 2013. doi: https://doi.org/10.2118/163582-MS

Emerick, Alexandre A., and Albert C. Reynolds. "History-Matching Production and Seismic Data in a Real Field Case Using the Ensemble Smoother With Multiple Data Assimilation." Paper presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, February 2013. doi: https://doi.org/10.2118/163675-MS

Ferraro, P., and F. Verga. "Use Of Evolutionary Algorithms In Single And Multi-Objective Optimization Techniques For Assisted History Matching." Paper presented at the Offshore Mediterranean Conference and Exhibition, Ravenna, Italy, March 2009.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence, 31(5), 855-868.

Grinestaff, GH. 1999. Waterflood pattern allocations: Quantifying the injector to producer relationship with streamline simulation. Proc., SPE Western Regional Meeting.

Grinestaff, GH and Caffrey, Daniel J. 2000. Waterflood management: A case study of the northwest fault block area of Prudhoe Bay, Alaska, using streamline simulation and traditional waterflood analysis. Proc., SPE Annual Technical Conference and Exhibition.

Grömping, Ulrike. (2015). Variable importance in regression models. Wiley Interdisciplinary Reviews: Computational Statistics, 7 (2): 137-152.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., & Nordlund, P. J. (2002). Particle filters for positioning, navigation, and tracking. IEEE Transactions on signal processing, 50(2), 425-437.

Gundersen, Kristian, Hosseini, Seyyed A, Oleynik, Anna et al. 2020. A Variational Auto-Encoder for Reservoir Monitoring.

Hajizadeh, Yasin, Christie, Michael A., and Vasily Demyanov. "Application of Differential Evolution as a New Method for Automatic History Matching." Paper presented at the Kuwait International Petroleum Conference and Exhibition, Kuwait City, Kuwait, December 2009. doi: https://doi.org/10.2118/127251-MS

Hastie, T., Tibshirani, R., and Friedman, J.H. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Springer.

Hawkins, D. M. (2004). The problem of overfitting. Journal of chemical information and computer sciences, 44(1), 1-12.

Heffer, K.J., Fox, R.J., McGill, C.A. et al. (1997). Novel Techniques Show Links between Reservoir Flow Directionality, Earth Stress, Fault Structure and

Geomechanical Changes in Mature Waterfloods. SPE Journal 2 (02): 91-98. https://doi.org/10.2118/30711-PA.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation 9 (8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hoffimann, Júlio. Brugge Realizations. Github Repository (Reprint). https://github.com/juliohm/HUM/tree/master/case_studies/brugge.

Holanda, R.W., Gildin, E., Jensen, J.L. et al. (2018). A state-of-the-art literature review on capacitance resistance models for reservoir characterization and performance forecasting. Energies 11 (12): 3368.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities (in eng). Proceedings of the National Academy of Sciences of the United States of America 79 (8): 2554-2558.

Jahani, N., Ambía, J., Fossum, K., Alyaev, S., Suter, E., & Torres-Verdín, C. (2021, May). Real-Time Ensemble-Based Well-Log Interpretation for Geosteering. In SPWLA 62nd Annual Logging Symposium. https://doi.org/10.30632/SPWLA-2021-0105. OnePetro

Jafarpour, Behnam and McLaughlin, Dennis B %J SPE Journal. 2009. Reservoir characterization with the discrete cosine transform. 14 (01): 182-201.

Jimenez, E., Datta-Gupta, A., & King, M. J. (2010). Full-field streamline tracing in complex faulted systems with non-neighbor connections. SPE Journal, 15(01), 7-17.

Kansao, Rami, Yrigoyen, Andres, Haris, Zameel et al. 2017. Waterflood Performance Diagnosis and Optimization Using Data-Driven Predictive Analytical Techniques from Capacitance Resistance Models CRM. Presented at the SPE Europec featured at 79th EAGE Conference and Exhibition, Paris, France. 2017/6/12/. https://doi.org/10.2118/185813-MS.

Khaninezhad, Mohammad-Reza M., Jafarpour, Behnam, and Li, Lianlin. 2010. History Matching With Learned Sparse Dictionaries. Proc., SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/133654-MS.

Kim, Y.D. & Durlofsky, L.J. (2021). A Recurrent Neural Network–Based Proxy Model for Well-Control Optimization with Nonlinear Output Constraints. SPE Journal: 1-21. https://doi.org/10.2118/203980-PA.

Kruskal, W. (1987). Relative importance by averaging over orderings. The American Statistician 41 (1): 6-10.

Lee, Sanghyun , and Karl, Dunbar Stephen. "Field Application Study on Automatic History Matching Using Particle Swarm Optimization." Paper presented at the SPE Reservoir Characterisation and Simulation Conference and Exhibition, Abu Dhabi, UAE, September 2019. doi: https://doi.org/10.2118/196678-MS

Lee, Hyokyeong, Yao, Ke-thia, Okpani, Olu Ogbonnaya et al. 2010. Identifying Injector-Producer Relationship in Waterflood Using Hybrid Constrained Nonlinear Optimization. Presented at the SPE Western Regional Meeting, Anaheim, California, USA. 2010/1/1/. https://doi.org/10.2118/132359-MS.

Li, R., Reynolds, A. C., & Oliver, D. S. (2003, December 1). History Matching of Three-Phase Flow Production Data. Society of Petroleum Engineers. doi:10.2118/87336-PA

Lin, Yen-Ting, Ortega, Antonio, Nejad, Amir Mohammad et al. 2010. Waterflood Tomography: Mapping High Contrast Permeability Structures Using Injection/Production Data. Presented at the SPE Western Regional Meeting, Anaheim, California, USA. 2010/1/1/. https://doi.org/10.2118/132699-MS.

Liu, Feilong, Guthrie, Charles F., and Shipley, David. 2012. Optimizing Water Injection Rates for a Water-flooding Field. Presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA. 2012/1/1/. https://doi.org/10.2118/157996-MS.

Liu, Tian , Hetz, Gill , Chen, Hongquan , and Akhil Datta-gupta. "Integration of Time-Lapse Seismic Data Using the Onset Time Approach: The Impact of Seismic Survey Frequency." Paper presented at the SPE Annual Technical Conference and Exhibition, Calgary, Alberta, Canada, September 2019. doi: https://doi.org/10.2118/196001-MS

Liu, T., Chen, H., Hetz, G., & Datta-Gupta, A. (2020). Integration of time-lapse seismic data using the onset time approach: The impact of seismic survey frequency. Journal of Petroleum Science and Engineering, 189, 106989.

Liu, W., Liu, W., & Gu, J. (2019). Reservoir Inter-Well Connectivity Analysis Based on a Data Driven Method. In Abu Dhabi International Petroleum Exhibition & Conference. https://doi.org/10.2118/197654-MS.

Ma, Xiang, and Nicholas Zabaras. "Kernel principal component analysis for stochastic input model generation." Journal of Computational Physics 230.19 (2011): 7311-7331.

Matthews, J. D., Carter, J. N., Stephen, K. D., Zimmerman, R. W., Skorstad, A., Manzocchi, T., & Howell, J. A. (2008). Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the SAIGUP project. Petroleum Geoscience, 14(1), 35-44.

Mishra, Srikanta and Datta-Gupta, Akhil. 2017. Applied Statistical Modeling and Data Analytics : A Practical Guide for the Petroleum Geosciences. Saint Louis, UNITED STATES: Elsevier.

Nævdal, Geir, Brouwer, D Roald, and Jansen, Jan-Dirk. 2006. Waterflooding using closed-loop control. Computational Geosciences 10 (1): 37-60.

Narasingam, Abhinav and Kwon, Joseph Sang-Il. 2017. Development of local dynamic mode decomposition with control: Application to model predictive control of hydraulic fracturing. Computers & Chemical Engineering 106: 501-511. http://www.sciencedirect.com/science/article/pii/S0098135417302739.

Nwachukwu, Azor, Jeong, Hoonyoung, Sun, Alexander et al. 2018. Machine Learning-Based Optimization of Well Locations and WAG Parameters under Geologic Uncertainty. Presented at the SPE Improved Oil Recovery Conference, Tulsa, Oklahoma, USA. 2018/4/14/. https://doi.org/10.2118/190239-MS.

Olalotiti-Lawal, F., Onishi, T., Datta-Gupta, A., Fujita, Y., Watanabe, D., & Hagiwara, K. (2019). Model calibration and optimization of a post-combustion CO2 WAG pilot in a mature oil field. Fuel, 255, 115810.

Olalotiti-Lawal, F., Onishi, Tsubasa, Kim, Hyunmin et al. 2019. Post-Combustion Carbon Dioxide Enhanced-Oil-Recovery Development in a Mature Oil Field: Model Calibration Using a Hierarchical Approach. SPE Reservoir Evaluation & Engineering 22 (03): 998-1014. https://doi.org/10.2118/187116-PA.

Oliva, J.B., Póczos, B., & Schneider, J. (2017). The Statistical Recurrent Unit. In International Conference on Machine Learning, (pp. 2671-2680). PMLR.

Pan, Y., Deng, L., Zhou, P., & Lee, W. J. (2021). Laplacian Echo-State Networks for production analysis and forecasting in unconventional reservoirs. Journal of Petroleum Science and Engineering, 109068.

Park, Jihoon, and Jef Caers. "Direct forecasting of global and spatial model parameters from dynamic data." Computers & Geosciences 143 (2020): 104567.

Peters, Lies, Arts, Rob, Brouwer, Geert et al. 2010. Results of the Brugge Benchmark Study for Flooding Optimization and History Matching. SPE Reservoir Evaluation & Engineering 13 (03): 391-405. https://doi.org/10.2118/119094-PA.

Pollock, David W. 1988. Semianalytical computation of path lines for finite-difference models. Groundwater 26 (6): 743-750.

Prakasa, Bona, Shi, Xiang, Muradov, Khafiz et al. 2017. Novel Application of Capacitance-Resistance Model for Reservoir Characterisation and Zonal, Intelligent Well Control. Presented at the SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition, Jakarta, Indonesia. 2017/10/17/. https://doi.org/10.2118/186277-MS.

Prevost, M., Edwards, M. G., & Blunt, M. J. (2002). Streamline tracing on curvilinear structured and unstructured grids. SPE Journal, 7(02), 139-148.

Raihan, D., & Chakravorty, S. (2018a). Particle Gaussian mixture filters-I. Automatica, 98, 331-340.

Raihan, D., & Chakravorty, S. (2018b). Particle Gaussian mixture filters-II. Automatica, 98, 341-349.

Rasmussen, A. F. (2010, September). Streamline tracing on irregular geometries. In ECMOR XII-12th European Conference on the Mathematics of Oil Recovery. (pp. cp-163). European Association of Geoscientists & Engineers.

Ren, B. & Duncan, I.J. (2019). Reservoir simulation of carbon storage associated with CO2 EOR in residual oil zones, San Andres formation of West Texas, Permian Basin, USA. Energy 167: 391-401.

Reynolds, Albert C, He, Nanqun, Chu, Lifu et al. 1996. Reparameterization techniques for generating reservoir descriptions conditioned to variograms and well-test pressure data. 1 (04): 413-426.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. Nature 323 (6088): 533-536.

Sarma, Pallav, Durlofsky, Louis J, Aziz, Khalid et al. 2007. A new approach to automatic history matching using kernel PCA. Proc., SPE Reservoir Simulation Symposium.

Sayarpour, Morteza. (2008). Development and application of capacitance-resistive models to water/CO_2 floods.

Sayarpour, Morteza, Kabir, C. Shah, and Lake, Larry Wayne. 2008. Field Applications of Capacitance Resistive Models in Waterfloods. Presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA. 2008/1/1/. https://doi.org/10.2118/114983-MS.

Sayyafzadeh, Mohammad , Haghighi, Manouchehr , and Jonathan N. Carter.
"Regularization in History Matching Using Multi-Objective Genetic Algorithm and Bayesian Framework." Paper presented at the SPE Europec/EAGE Annual Conference, Copenhagen, Denmark, June 2012. doi: https://doi.org/10.2118/154544-MS

Schlumberger. 2018. Eclipse 2018.1 User Guide. Eclipse Manuals.

Sen, D., Chen, H., Datta-Gupta, A., Kwon, J., & Mishra, S. (2020a, October). Data-Driven Rate Optimization Under Geologic Uncertainty. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers. https://doi.org/10.2118/201325-MS.

Sen, D., Chen, H., Datta-Gupta, A., Kwon, J., & Mishra, S. (2021). Machine learning based rate optimization under geologic uncertainty. Journal of Petroleum Science and Engineering, 109116.

Sen, D., Ong, C., Kainkaryam, S., & Sharma, A. (2020b). Automatic Detection of Anomalous Density Measurements due to Wellbore Cave-in. Petrophysics, 61(05), 434-449.

Sen, D., Raihan, D., & Chidambaram, M. (2014). Multiway continuous hidden Markov model-based approach for fault detection and diagnosis. AIChE Journal, 60(6), 2035-2047.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404: 132306.

Suarsana, I. P., & Badril, A. (2011, May). Comparison of Tracer Test Result and Analysis of Connectivity Injector and Producer during Pilot Waterflood Kenali Asam Zone P/1050. In SPE EUROPEC/EAGE Annual Conference and Exhibition. OnePetro.

Sudaryanto, Bagus and Yortsos, Yannis C. 2001. Optimization of Displacements in Porous Media Using Rate Control. Presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana. 2001/1/1/. https://doi.org/10.2118/71509-MS.

Taware, Satyajit, Alhuthali, Ahmed H., Sharma, Mohan et al. 2017. Optimal rate control under geologic uncertainty: water flood and EOR processes. Optimization and Engineering 18 (1): 63-86. https://doi.org/10.1007/s11081-016-9345-y.

Tian, Chuan and Horne, Roland N. 2016. Inferring Interwell Connectivity Using Production Data. Presented at the SPE Annual Technical Conference and Exhibition, Dubai, UAE. 2016/9/26/. https://doi.org/10.2118/181556-MS.

Tian, C., & Horne, R. N. (2015, September). Machine learning applied to multiwell test analysis and flow rate reconstruction. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Tian, C., & Horne, R. N. (2016, September). Inferring interwell connectivity using production data. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Tian, C., & Horne, R. N. (2017, October). Recurrent neural networks for permanent downhole gauge data analysis. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Tian, C., & Horne, R. N. (2019). Applying machine-learning techniques to interpret flow-rate, pressure, and temperature data from permanent downhole gauges. SPE Reservoir Evaluation & Engineering, 22(02), 386-401.

Van Essen, G. M., Zandvliet, M. J., Van den Hof, P. M. J. et al. Robust optimization of oil reservoir flooding. 2006, 699-704.

Veettil, D. R. A., & Clark, K. (2020). Bayesian geosteering using sequential monte carlo methods. Petrophysics, 61(01), 99-111.

Von Luxburg, Ulrike 2007. A tutorial on spectral clustering. %J Statistics and computing 17 (4): 395-416.

Wan, E. A., & Van Der Merwe, R. (2000, October). The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373) (pp. 153-158). IEEE.

Wang, Chunhong, Li, Gaoming, and Reynolds, Albert C. 2009. Production Optimization in Closed-Loop Reservoir Management. SPE Journal 14 (03): 506-523. https://doi.org/10.2118/109805-PA.

Xiong, Hao, Devegowda, Deepak, and Huang, Liangliang. 2019. EOR solvent-oil interaction in clay-hosted pores: Insights from molecular dynamics simulations. Fuel 249: 233-251. http://www.sciencedirect.com/science/article/pii/S0016236119304752.

Yousef, Ali A., Gentil, Pablo H., Jensen, Jerry L. et al. 2006. A Capacitance Model To Infer Interwell Connectivity From Production and Injection Rate Fluctuations. SPE Reservoir Evaluation & Engineering 9 (06): 630-646. https://doi.org/10.2118/95322-PA.

Zhai, Xiang, Wen, Tailai, and Matringe, Sebastien. 2016. Production Optimization in Waterfloods with a New Approach of Inter-Well Connectivity Modeling. Presented at the SPE Asia Pacific Oil & Gas Conference and Exhibition, Perth, Australia. 2016/10/25/. https://doi.org/10.2118/182450-MS.

Zhang, K., Tang, M., Yong, S. S., & Shan, Y. (2016, August). Evaluation of Stage Contribution and Interwell Connectivity During Initial Flowback and Oil Production in a Tight Oil Horizontal Stimulation Using Tracer Technology. In SPE Asia Pacific Hydraulic Fracturing Conference. Society of Petroleum Engineers.

Zhang, Kai, Zhang, Jinding, Ma, Xiaopeng et al. 2021. History Matching of Naturally Fractured Reservoirs Using a Deep Sparse Autoencoder. SPE Journal: 1-22. https://doi.org/10.2118/205340-PA.

Zhang, Y., King, M. J., & Datta-Gupta, A. (2012). Robust streamline tracing using inter-cell fluxes in locally refined and unstructured grids. Water Resources Research, 48(6).

Zhou, P., Pan, Y., Sang, H., & Lee, W. J. (2018, September). Criteria for proper production decline models and algorithm for decline curve parameter inference. In Unconventional Resources Technology Conference, Houston, Texas, 23-25 July 2018 (pp. 3535-3551). Society of Exploration Geophysicists, American Association of Petroleum Geologists, Society of Petroleum Engineers.

Zuber, V., & Strimmer, K. (2011). High-dimensional regression and variable selection using CAR scores. Statistical Applications in Genetics and Molecular Biology, 10(1).

Zuo, L., Lim, J., Chen, R., Chen, H., Shamekhi, A., & King, M. J. (2021). Continuous streamline trajectories on complex grids. Computational Geosciences, 1-25.