

# Oil Field Production using Machine Learning

## CS 229 Project Report

Sumeet Trehan, Energy Resources Engineering, Stanford University

### 1 Introduction

Effective management of reservoirs motivates oil and gas companies to do uncertainty analysis, optimize production and/or field development etc. Any such analysis requires a large number of flow simulations, of the order of hundreds in case of gradient-based methods or even thousands in case of direct search or stochastic procedures such as genetic algorithms. Since flow simulations are computationally very expensive, performing a large number of simulations (say, hundreds) is difficult.

While recent advances in parallel computing have reduced the computing time [3], the simulation of realistic models with  $O(10^4 - 10^6)$  grid cells and multiple components still remains challenging. As a result, reservoir simulation has found limited application as a tool in the development of smart fields.

In this study, combination of supervised and unsupervised machine learning (ML) techniques have been explored to enable large speedups in reservoir simulation, thereby lowering its computational cost and enabling wider-scale applications. These techniques enable fast computation of approximate solutions to various scenarios that arise in optimization or uncertainty quantification.

### 2 Model setup and training data

The reservoir model, shown in figure 2.1, is a two-dimensional horizontal synthetic permeability field and corresponds to a fluvial depositional system described by [1]. The field has three producers designated as  $P_1, P_2$  and  $P_3$ , and three injectors designated  $I_1, I_2$  and  $I_3$ .

Training data is obtained by solving the governing equation (conservation of mass and momentum) for reservoir flow. As described in the introduction, solving governing equations is very expensive. Thus, we perform only one simulation. Since reservoir flow is a dynamic process, we get time series data for amount of oil and water produced.

The input variable which we would like to vary between training and test is referred to as bottom-hole pressure (BHP). This time varying variable (represented by  $x$ ) plays a critical role in optimizing production or field development.

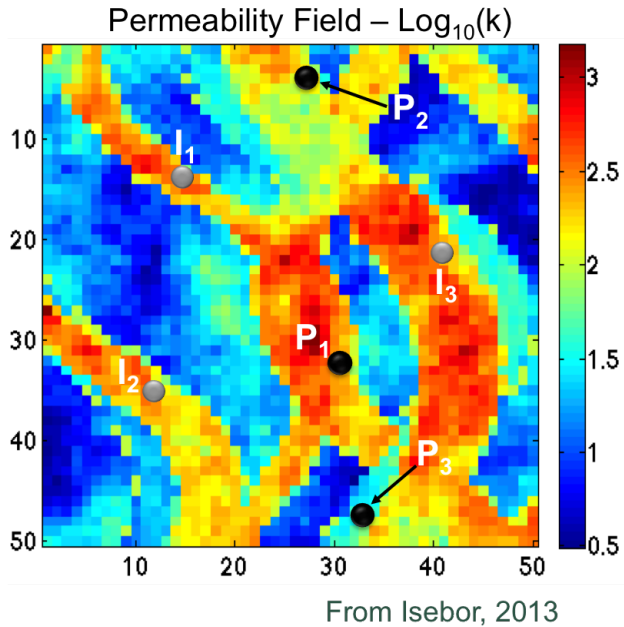


Figure 2.1: Permeability  $\log_{10}K$  for reservoir model

### 3 Model based on supervised and unsupervised learning techniques

In the current study, we develop a model which uses three different techniques, viz. 1) modified version of linear regression, 2) finding closest neighbor, and 3) principal component analysis (PCA).

#### 3.1 Modified linear regression

As described above, training data is obtained by solving the governing equations which obey the laws of physics. The use of simple linear or weighted linear regression with multiple features to build a model was observed to lead to large errors.

Therefore, a separate mathematical procedure was adopted using Taylor series [2, 4]. Since the Taylor series was truncated at the linear term, it is equivalent to performing a linear regression where we directly obtain the product  $(\theta^T x)$  without explicitly constructing all the features or coefficients. If at any instant of time  $n + 1$ , we want to estimate the output function  $y^{n+1}$  for some new query point  $x^{n+1}$  (i.e., a different BHP value), then the procedure is demonstrated below.

An underlying assumption in Taylor series is the availability of a point in the neighborhood where the function value is known and where linearization of the function will hold true. In this problem, we pick the point (around which we expand) by exploiting the concept of pairwise distance

and determining the closest point in the neighborhood. This is done by comparing the function value  $y^n$ , a known quantity, against the time series data available from training.

Note that  $y^{n+1}$  represents the unknown, while  $y^n$  is known, since we assume that at  $T = 0$ , the function value is known, and this is always true.

$$\mathbf{y}^{n+1} = \mathbf{y}^{i+1} - (\mathbf{J}^{i+1})^{-1} (\mathbf{B}^{i+1}(\mathbf{y}^n - \mathbf{y}^i) + \mathbf{C}^{i+1}(\mathbf{x}^{n+1} - \mathbf{x}^{i+1})) \quad (3.1)$$

The above equation is in the form of  $y = \theta^T x + \theta_o$  where  $\theta^T x = -(\mathbf{J}^{i+1})^{-1} \mathbf{C}^{i+1} \mathbf{x}^{n+1}$  and  $\theta_o = \mathbf{y}^{i+1} - (\mathbf{J}^{i+1})^{-1} \mathbf{B}^{i+1}(\mathbf{y}^n - \mathbf{y}^i) + (\mathbf{J}^{i+1})^{-1} \mathbf{C}^{i+1} \mathbf{x}^{i+1}$ .

Notation:

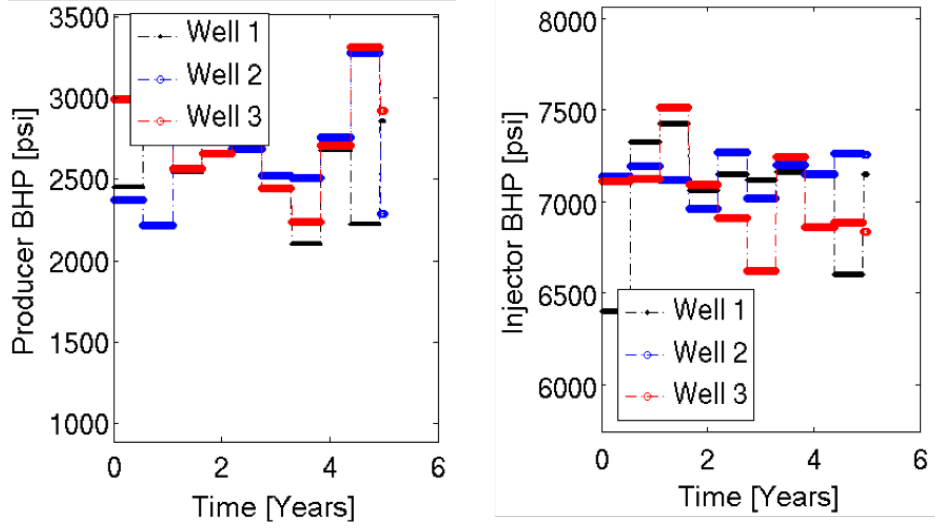
1. Superscript indicates time level, e.g.,  $i$  means time level  $i$
2. Coefficients with superscript  $i$ ,  $n$  and  $i + 1$  are known. Thus, J, B, C are known coefficient matrices.
3. BHP is the variable that is varied between the training and test and is represented by  $x^{n+1}$ . As described earlier, it is a time varying variable whose variation for training and test case is shown in Figure 3.1. The relationship between  $x$  and  $y$  is very non-linear.
4.  $y^i$  is a vector which represents pressure and saturation at different spatial locations in the reservoir (Figure 2.1) at time level  $i$ .
5. Amount of oil production at time level  $i$  is computed using pressure and saturation information contained in vector  $y^i$ .

Since at any instant of time, the output  $y^i$  is a high-dimensional vector we project  $y$  in lower dimensional space by construction PCA basis ( $\Phi$ ) using a different value of  $y$  available from training. We represent the lower dimensional vector with subscript  $r$ . The above equation in lower dimensional space is shown below:

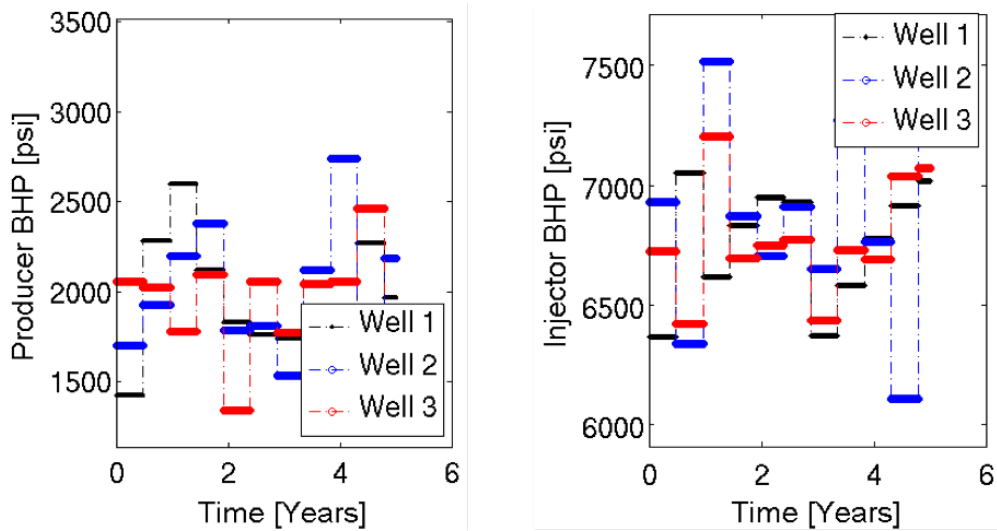
$$\mathbf{y}_r^{n+1} = \mathbf{y}_r^{i+1} - (\Phi^T \mathbf{J}^{i+1} \Phi)^{-1} (\Phi^T \mathbf{B} \Phi^{i+1} (\mathbf{y}_r^n - \mathbf{y}_r^i) + \Phi^T \mathbf{C}^{i+1} (\mathbf{x}^{n+1} - \mathbf{x}^{i+1})) \quad (3.2)$$

## 4 Results

Time series data for training and test cases is shown in Figure 4.1. As observed from Figure 3.1, the input variable, BHP, is significantly different between training and test. Since we have only one training, the training error is zero.



(a) Training BHP represented by  $x^i$  for producer and injector wells

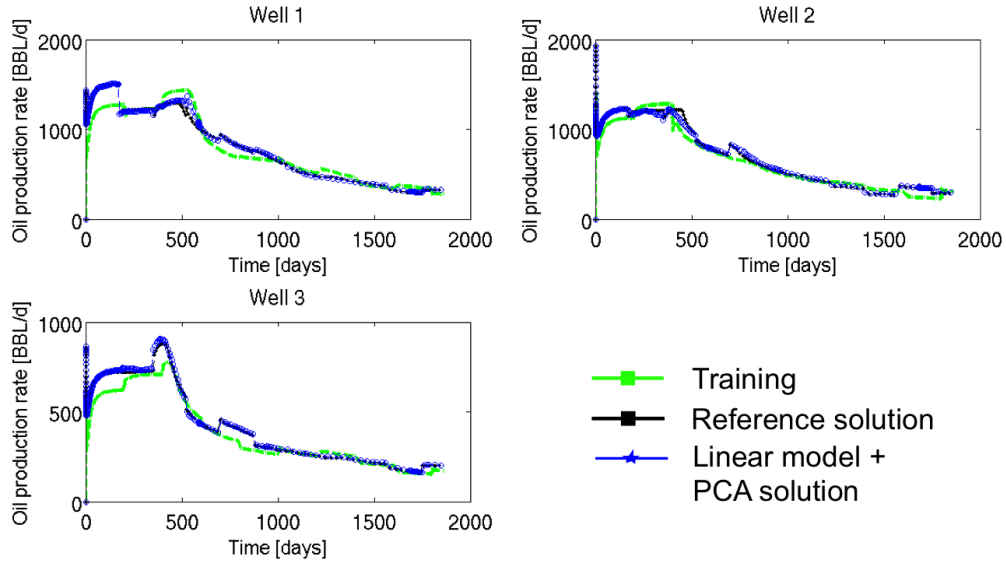


(b) Test BHP represented by  $x^n$  for producer and injector wells

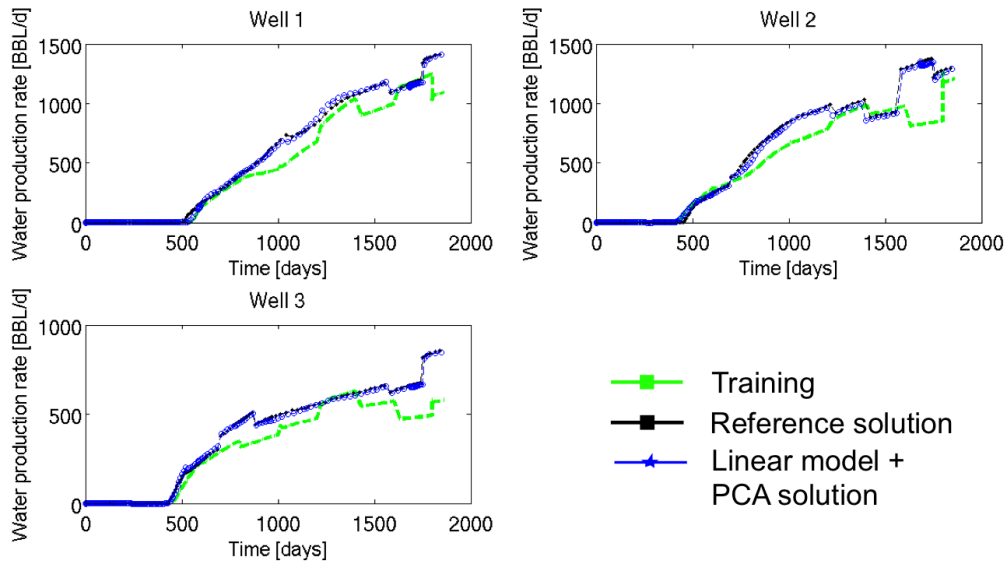
Figure 3.1: BHP profile for training and one sample test case

We observe from Figure 4.1 that we have an excellent match between the solution obtained from ML technique described in § 3.1 (shown in blue color) and reference solution to the test problem (shown in black). We observe a consistently good match for both oil and water rates. In order to evaluate the test error, we adopted a three step procedure, viz.,

1. Create multiple test cases where each test case has a unique BHP profile. One such profile is shown in Figure 3.1b.
2. For each test case, solve the governing equations and obtain a reference solution.



(a) Oil rates: Training, Reference solution for test ML solution



(b) Water rates: Training, Reference solution for test ML solution

Figure 4.1: Time series data for oil and water production rates for training and test

- Find the error in oil and water production rates by comparing solution obtained using ML techniques versus the reference (obtained as part of post-processing step and described above). The error for this time-varying process is defined in equation 4.1 and error behavior is shown in Figure 4.2.

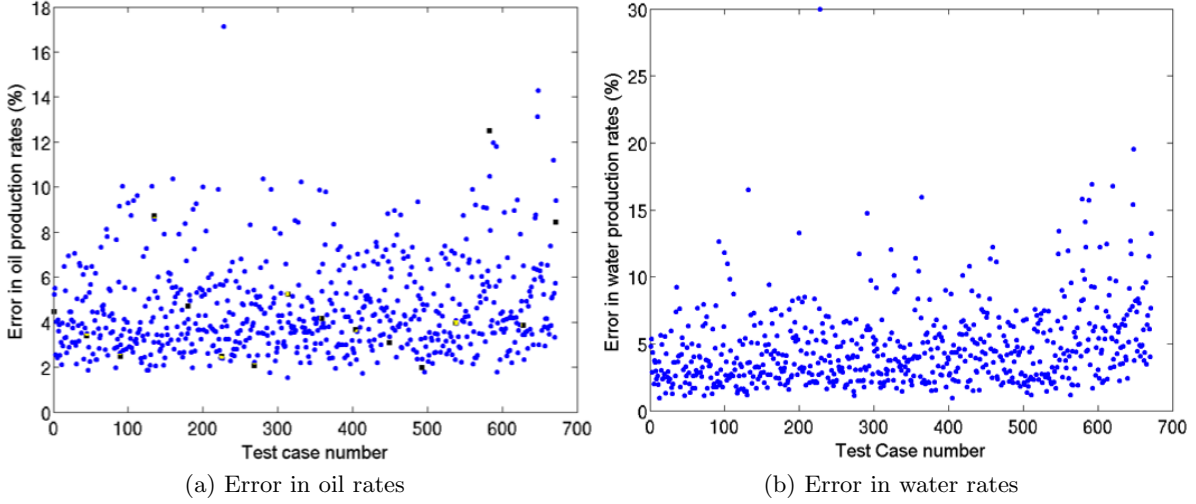


Figure 4.2: Test error

$$E_{o,w}^j = \frac{\int_0^T |Q_{o,w}^{j, ML} - Q_{o,w}^{j, ref}| dt}{\int_0^T Q_{o,w}^{j, ref} dt} \quad (4.1)$$

$$Error_{o,w} = \frac{1}{n_{pw}} \sum_{j=1}^{n_{pw}} E_{o,w}^j$$

The error plots indicate that for majority of the test cases, the test error is around 5-8%, which is an acceptable value, given other uncertainties present in the reservoir (outside the scope of this study). However, the large speedups obtained using ML techniques indicate that this is promising approach.

## 5 Conclusion and Future Work

We have implemented linear regression models without explicitly constructing parameters. Further, we found that traditional linear or weighted linear regression does not work well in this problem. In order to make the algorithm work, we successfully applied the concepts of closest neighbors and dimensionality reduction using PCA. Future work includes exploration of neural networks and the EM algorithm for such problems.

## 6 References

1. O. J. Isebor. *Derivative-free optimization for generalized oil field development*. PhD thesis, Department of Petroleum Engineering, Stanford University, 2013.
2. J. He. *Reduced-order modeling for oil-water and compositional systems, with application*

*to data assimilation and production optimization.* PhD Thesis, Department of Petroleum Engineering, Stanford University, 2013.

3. S. Maliassov, B. Beckner, and V. Dyadechko. Parallel reservoir simulation using a specific software framework (SPE 163653). In *Reservoir Simulation Symposium*, Woodlands, TX, February 2013.
4. M. Rewiński and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155170, February 2003.