PARAMETRIC MODEL ORDER REDUCTION FOR OPTIMIZATION IN CLOSED LOOP FIELD DEVELOPMENT USING MACHINE LEARNING TECHNIQUES

A Dissertation

by

HARDIKKUMAR ASHOKBHAI ZALAVADIA

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Eduardo Gildin
Committee Members,	Michael King
	Akhil Datta-Gupta
	Yalchin Efendiev
Head of Department,	Jeff Spath

August 2019

Major Subject: Petroleum Engineering

ABSTRACT

Field development workflows consist of production optimization and data assimilation procedures that require running large number of reservoir simulations for fine scale models. Recent advancements in parallel computing and accelerated solvers have reduced simulation times for such high-fidelity models, however, repeated simulations and underlying complex non-linearities involved in multiphase and multicomponent models still remain a bottleneck. This computational challenge has motivated the development of Model Order Reduction (MOR) techniques which provide low dimensional representation of high-fidelity models and thus provide significant computational savings with the efforts to preserve the accuracy of simulation outputs. The aim of my research is to develop projection based MOR workflows for optimization problems in closed loop field development procedure, which include well control optimization and well placement optimization. We pose the problem formulation as Parametric Model Order Reduction (PMOR) that allows for taking into consideration a system parameter for each optimization problem considered. For developing Reduced Order Models (ROMs) for such problems, we use projection based Proper Orthogonal Decomposition (POD) which enables representation of reservoir state variables in terms of highly reduced set of variables.

First part of the research is based on developing ROMs for well control optimization problem, where we look for the optimal strategy to control the wells settings. Here we use DEIM in addition to POD for quick evaluation of non-linear functions. We introduce a novel training procedure for global ROM during control optimization, which proved to give accurate results when compared to optimization using fine scale simulations. We test the performance of POD-DEIM for different optimization parameterization methods like polynomial and piecewise polynomial approximations on a waterflooding scenario. Polynomial approximation of BHP control served as good training sets for POD-DEIM with the training

strategy proposed leading to accurate and fast reduced model.

The second part of my research, which is a major contribution of my work, is based on developing ROMs for changing well locations during well placement optimization problem. Here, we do not employ proposed MOR on well location optimization problem, rather develop MOR strategies as a precursor to be used for well location optimization in future. Projection based reduced order modeling methodologies for well control optimization have reached a good level of maturity, however, MOR development for changing well configurations, is unexplored. We first propose error based local PMOR for new well location using a Machine Learning (ML) framework with POD. ML algorithms like Neural Networks and Random Forests help us predict the ROM error that eventually will choose appropriate basis at a new well location from previously computed reduced models. We introduce geometry based features and physics based flow diagnostics features to train ML models.

In efforts to tackle the issues with local PMOR technique proposed, we introduce a novel global non-intrusive PMOR technique based on machine learning. The idea here is to represent the entire parameter space of well location by a single global ROB and then using ML model to establish a relation between the input well location information and the POD basis coefficients of each state. We then also formulate the error correction model based on the reduced model solution, to account for solution discrepancies. The proposed method, that can make use of parallel resources efficiently, shows promising results on waterflooding case studies in predicting various quantities of interest (QoI) at new well locations such as oil production rates and water cut, and showed significant speedups of one to two orders of magnitude for the test cases.

DEDICATION

To my beloved grandparents, who I always look up to.

To my parents, for their endless support and believing in me.

To my brother, who is my best friend.

To my uncle and aunt, without whose support this day would not have been possible.

To my fiancee, who I find my soulmate in.

And finally to God, for giving us all we have, the strength and a healthy life.

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to my advisor, Dr. Eduardo Gildin. He provided me with a very interesting research topic and constant guidance throughout the course of my PhD. Dr. Gildin has been a constant source of encouragement and positivity, and provided me with a freedom to explore ideas and direct my research. His continuous availability for discussions and belief in his students have helped me a lot to excel beyond my expectations. It has been an immense pleasure to work with and learn from him.

Special thanks to Dr. Yalchin Efendiev for giving me the opportunity to work with him during my research, that helped me explore many directions and provided me with useful insights on the research topic. Special thanks is also due to Dr. Michael King for his very useful comments and suggestions that helped me immensely during the second part of my research and to Dr. Datta-Gupta for serving in my committee and his insights and guidance in my research through the courses I took with him. A big thank to late Dr. Fluodas with whom we collaborated for first part of the research. Special thanks to Nadav Sorek and Xiaosi Tan, colleagues in our group, and Nilabja Guha, Dr. Efendiev's student, with whom I collaborated during the first part of my research. I am grateful to Sathish Sankaran and Rabi Chakraborty, who served as my mentors during various internships and got a lot to learn from them.

My wonderful time at Texas A&M would have been incomplete without my colleagues and friends with whom I spent good times and had interesting discussions including Alex Lichtig, Jaime Leal, Marcello Dall'Aqua, Alexander Tarakanov, Jae Wook Lee, Rafael Holanda, Anqi Bao, Enrique Losoya, Kildare George, Emilio Coutinho, Rupen Sakariya, Prakhar Sarkar, Nitin Chaudhary and Shyam Sajeev. Heartful thanks to my friends, Krishna Nunna, Ankit Bansal and Hongquan Chen from Dr. King and Dr. Datta-Gupta's group for useful discussions during my research. Finally, I would like to thank Skoltech Institute and Energi Simulation (formerly Foundation CMG) for financial support of this research.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by my dissertation committee members Professor Eduardo Gildin, Professor Michael King, Professor Yalchin Efendiev and Professor Akhil Datta-Gupta. Research in Section 3 in this dissertation was carried out in collaboration with Dr. Nadav Sorek, previous colleague from our research group.

Funding Sources

This work was supported in part by Skoltech Institute and Energi Simulation (Former Foundation CMG) Research Chair Grant at Texas A&M.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGMENTS v

CONTRIBUTORS AND FUNDING SOURCES vi

TABLE OF CONTENTS vii

LIST OF FIGURES x

LIST OF TABLES xviii

1. INTRODUCTION AND LITERATURE REVIEW 1

    1.1 Background 1
    1.2 Literature Review 5
        1.2.1 Model Order Reduction 5
        1.2.2 Model Order Reduction Background for Well Control Optimization 9
        1.2.3 Model Order Reduction Background for Well Placement Optimization 11
    1.3 Scope of Work 14
    1.4 Dissertation Outline 16

2. RESERVOIR SIMULATION AND PROJECTION BASED MODEL ORDER REDUCTION FOR PARAMETRIC SYSTEM 18

    2.1 Reservoir Simulation 18
        2.1.1 Governing Equations 18
        2.1.2 Incompressible and slightly compressible flow 19
        2.1.3 Discretized parametric equations for well control and well location 20
    2.2 Controllability of reservoir flow 22
        2.2.1 Case study to analyze controllability properties for changing well controls and well locations 24
    2.3 Projection based Model Order Reduction 29
        2.3.1 Galerkin projection 30
        2.3.2 Proper Orthogonal Decomposition 32
        2.3.3 Parametric Model Order Reduction concept 34

3. MODEL ORDER REDUCTION FOR WELL CONTROL OPTIMIZATION 36

3.1 Well Control Optimization Formulation 37
3.1.1 Optimal Control of Reservoir Flooding 37
3.1.2 Optimization Dimensionality Reduction by Polynomial Approximation 38
3.1.3 Infinite Dimensional Problem Statement 39
3.1.4 Objective function 40
3.1.5 The Infinite Dimension Optimization Problem 41
3.2 Parameterization Strategies 41
3.2.1 Piecewise Constant Parametrization 42
3.2.2 Polynomial Approximation using Chebyshev Polynomials 43
3.2.3 Piecewise Polynomial Approximation using Cubic Spline Interpolation . 47
3.3 Motivation for Polynomial Controls 48
3.3.1 Case Study 49
3.3.2 Parametrization Sensitivity Computational Results 50
3.4 Global Parametric Model Order Reduction 54
3.4.1 Discrete Empirical Interpolation Method 54
3.4.2 Global PMOR training strategy 56
3.5 Case Study 58
3.5.1 Problem Statement 59
3.5.2 PMOR training 59
3.5.3 POD-DEIM Performance for a Representative Case 60
3.5.4 Comparison: PSO Optimal Control Solutions 62
3.6 Summary 68
3.7 Bayesian framework for probabilistic sampling of basis functions 70
3.7.1 Bayesian Formulation 71
3.7.2 Case Study 73
3.7.3 Summary 76

4. LOCAL MODEL ORDER REDUCTION FOR CHANGING WELL LOCATIONS DURING WELL PLACEMENT OPTIMIZATION 78

4.1 MOR challenges for changing well locations 79
4.2 Observations 80
4.2.1 Global PMOR results 80
4.2.2 Local PMOR results 82
4.3 POD-based local PMOR problem formulation 86
4.4 Machine Learning Overview 88
4.4.1 Artificial Neural Network (ANN) 88
4.4.2 Random Forests (RF) 90
4.5 Well Location Feature Determination 91
4.5.1 Geometric features 92
4.5.2 Physics based features 94
4.5.3 Feature Selection 99
4.6 Case Study 101
4.6.1 Case 1: Error maps of basis obtained by single well location 103

4.6.2 Case 2: Error maps of basis obtained by snapshot concatenation of two well locations ... 118
4.7 Computational Complexity ... 122
4.8 Conclusion and Future Work ... 123

5. NON-INTRUSIVE GLOBAL MODEL ORDER REDUCTION FOR CHANGING WELL LOCATIONS DURING WELL PLACEMENT OPTIMIZATION ... 125

5.1 Motivation for non-intrusive global PMOR using machine learning ... 126
5.1.1 Parameterizing states of reservoir using POD ... 127
5.1.2 Addressing physical constraints in POD ... 130
5.2 Global PMOR problem formulation ... 130
5.2.1 Machine Learning and feature selection ... 132
5.2.2 Remarks ... 134
5.3 Case Study ... 134
5.3.1 Energy of eigenvalues - Global PMOR ... 136
5.3.2 Results ... 138
5.4 Modeling error correction ... 148
5.5 Case study ... 149
5.5.1 Homogeneous reservoir model ... 149
5.5.2 Heterogeneous reservoir model - single well case ... 159
5.5.3 Heterogeneous reservoir models - multiple wells case ... 174
5.5.4 Advantages and Limitations ... 178
5.6 Summary ... 179

6. CONCLUSIONS AND FUTURE WORK ... 184

6.1 PMOR for well control optimization ... 184
6.2 PMOR for changing well location during well location optimization ... 186
6.3 Suggestions for future work ... 187

REFERENCES ... 190

LIST OF FIGURES

FIGURE Page

1.1 The U.S EIA prediction of energy consumption by fuel (reprinted from [1]) 2

1.2 Key elements of the closed-loop reservoir management process 3

1.3 Hybrid Modeling Methods 5

1.4 Model Order Reduction concept for reservoir simulation models (reprinted from [20]) 7

1.5 PMOR methodology where the offline stage computes fine scale simulations at sample points in database and ROMs or ROBs are constructed, and in the online stage, solutions are predicted at a new parameter ζ^* based on the information from offline stage (adapted from [21]) 8

1.6 Scope of Work. Green boxes are the area of focus..... 15

2.1 Homogeneous reservoir permeability field with 1 producer and 1 injector..... 25

2.2 (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation 26

2.3 (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for BHP case 1 26

2.4 (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation 27

2.5 (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for BHP case 2 27

2.6 (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation 28

2.7 (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for new well location 28

3.1 Chebyshev polynomials of the first kind. Left: degrees zero up to four. Right: degrees five, six and seven..... 45

3.2 Examples of polynomial trajectories obtained by weighted sum of Chebyshev polynomials. Intervals outside of the box are treated as (nearest) boundary control. 46

3.3 Synthetic two-dimensional reservoir, with its relative permeability curves 49

3.4 (Left) Gradient free (PSO) and (Right) gradient based (IPOPT) results for different numbers of basis functions, normalized to the best solution found ($17.73 million) 51

3.5 (Left) Gradient free (PSO) and (Right) gradient based (IPOPT) number of iterations 51

3.6 (Optimal controls of level 10 obtained by each parametrization method. Top: Gradient-free (PSO). Bottom: Gradient-based (IPOPT) 53

3.7 Workflow for optimization-online training of POD-DEIM 57

3.8 Reservoir permeability field with 4 producers and 1 injector (after [90]) 60

3.9 (a) Pressure and (b) Saturation basis selection for POD and (c) cells selected for DEIM (in strong colors) 61

3.10 Three training samples from the initial eight-particles swarm of Chebyshev profiles 61

3.11 (Top) Saturation profiles from POD-DEIM Chebyshev optimal control using fine scale simulation and (Below) using the reduced model at different times and their relative errors 62

3.12 Oil production rate from the high fidelity simulation and reduced model simulation for 4 production wells using the Chebyshev optimal control solution from POD-DEIM..... 63

3.13 Water production rate from the high fidelity simulation and reduced model simulation for 4 production wells using the Chebyshev optimal control solution from POD-DEIM 64

3.14 PSO optimal control solution comparison from POD-DEIM and fine-scale using all three parameterization methods (a) Chebyshev polynomial control (b) Spline control (c) Piecewise constant control 66

3.15 Control trajectories relative error for a PSO algorithm 67

3.16 Control trajectories relative error for a PSO algorithm 67

3.17 Heterogeneous permeability field with one producer (red) and one injector (white) 74

3.18 (a) Training BHP profile for producer and injector (b) Test BHP profiles of producer and injector 74

3.19 Pressure solution comparison at the last time step for which basis functions are sampled - $N_{basis} = 8$ 76

3.20 Saturation solution comparison at the last time step for which basis functions are sampled - $N_{basis} = 8$ 76

4.1 Normal scale k_{norm} of permeability field. The values of permeability are chosen as $5^{k_{norm}} mD$ to exhibit low contrast. Red indicates the producer well and white is the injector 82

4.2 Global basis construction using greedy search. At each stage from (a) – (c), a producer well is set to the location of maximum error to concatenate the snapshot with that from previous configuration 83

4.3 (a) is the permeability field with well configuration shown. (b) shows the error map for this well configuration and the given basis 84

4.4 Oil production rate comparison at well location (7,16) between the true solution and that predicted by POD using a basis from well location (2,19) 84

4.5 (a) and (b) shows the error map for different basis dimensions for the given well configuration 86

4.6 Schematic of a feedforward Artificial Neural Network 90

4.7 Description of Random Forests model where, each of the branches from sample dataset correspond to a bootstrap sample and a random feature subset is chosen for each branch to construct a decision tree 91

4.8 (a) shows the error map for a homogeneous reservoir. (b) shows the feature parameters chosen on the basis of results from (a) 92

4.9 (a) Heterogeneous channelized permeability field with one producer in red and one injector in white (b) Streamlines imposed on the permeability field 99

4.10 (a) The total travel time of the particle injected at injector to any point in the reservoir (b) $F - \Phi$ plot to compute Lorenz coefficient as the area under the solid $F - \Phi$ curve and the dashed line $F = \Phi$ 100

4.11 Wrapper approach 101

4.12 Workflow of the ML based local PMOR procedure 102

4.13 Normal scale k_{norm} of permeability field. The values of permeability are chosen as $5^{k_{norm}} mD$ to exhibit low contrast. Red indicates the producer well and white is the injector. The producer well changes location at any location in the reservoir 104

4.14 Samples collected from 4 error maps as shown by red $\times$ for training the regression ML models 105

4.15 Samples collected from 2 of the 4 error maps as shown by red $\times$ for training the classification ML models 106

4.16 ANN regression prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) and (b) $L_p = 2, L_s = 8$ at (2,19) gridblocks 107

4.17 RF regression prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) and (b) $L_p = 2, L_s = 8$ at (2,19) gridblocks 108

4.18 ANN classification prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) gridblock and (b) $L_p = 2, L_s = 8$ for basis well at (2,19) gridblock 109

4.19 RF classification prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) gridblock and (b) $L_p = 2, L_s = 8$ for basis well at (2,19) gridblock 110

4.20 Oil production rate comparison at well location (17,14) between the true solution and that predicted by POD using a basis from well location (4,19) 111

4.21 Box plot of accuracy for regression results using RF and NN on 10 sample datasets 112

4.22 Box plot of accuracy for classification results using RF and NN on 10 sample datasets 113

4.23 Heterogeneous permeability field in log scale 114

4.24 Samples collected from 4 error maps as shown by red $\times$ for training the classification ML models 115

4.25 ANN classification prediction for an error map corresponding to (a) $L_p = 16, L_s = 23$ for basis well at (44,11) gridblock and (b) $L_p = 15, L_s = 22$ for basis well at (32,19) gridblock 116

4.26 RF classification prediction for an error map corresponding to (a) $L_p = 16, L_s = 23$ for basis well at (44,11) gridblock and (b) $L_p = 15, L_s = 22$ for basis well at (32,19) gridblock 117

4.27 Normal scale k_{norm} of permeability field. Two producer well location used for snapshot concatenation 119

4.28 Error map prediction (regression) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 2, L_s = 3$ for basis wells at (4,19) and (9,15) gridblock 120

4.29 Error map prediction (regression) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock 120

4.30 Error map prediction (classification) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock 121

4.31 Error map prediction (classification) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock 121

5.1 (a) Homogeneous permeability reservoir with 1 injector (white) and 1 producer (red). Producer is the parameter and can change location anywhere in the reservoir (b) BHP profiles of injector and producer that is set constant for all well configurations considered 135

5.2 Change in cumulative energy of the basis as calculated by the magnitude of eigenvalues for increasing number of parameters (well configurations) in the state snapshot matrix 137

5.3 (a) Training samples of producer well locations shown by red crosses (b) Test case 1 well location at (3,13) in homogeneous reservoir 138

5.4 Pressure solution comparison at (a) Time = 70 days and (b) Time = 360 days 140

5.5 Saturation solution comparison at (a) Time = 70 days and (b) Time = 360 days 141

5.6 Relative error in pressure at time = (a) 1 day , (b) 70 days and (c) 360 days .. 141

5.7 Relative error in saturation at time = (a) 1 day , (b) 70 days and (c) 360 days 142

5.8 Well block state solution comparison (a) Pressure and (b) Saturation 142

5.9 Quantities of Interest comparison (a) Oil production rate and (b) Water cut .. 143

5.10 Test case 2 with producer at (8,4) grid block and injector at (20,1) gridblock in homogeneous permeability reservoir 143

5.11 Pressure solution comparison at (a) Time = 70 days and (b) Time = 360 days 144

5.12 Saturation solution comparison at (a) Time = 70 days and (b) Time = 360 days 145

5.13 Well block state solution comparison (a) Pressure and (b) Saturation 145

5.14 Quantities of Interest comparison (a) Oil production rate and (b) Water cut .. 146

5.15 True and ML predicted pressure basis coefficient comparison at time = (a) 70 days and (b) 360 days 146

5.16 True and ML predicted saturation basis coefficient comparison at time = (a) 70 days and (b) 360 days 147

5.17 Comparison of predicted orthogonal and true orthogonal solutions with true solution at time = 360 days for (a) Pressure and (b) Saturation 147

5.18 (a) Test case with producer well location at (2,6) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 100 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) ... 151

5.19 (a) Test case with producer well location at (3,16) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 152

5.20 (a) Test case with producer well location at (16,19) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 100 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) ... 153

5.21 Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions 154

5.22 (a) Test case with producer well location at (3,16) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 50 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) ... 156

5.23 (a) Test case with producer well location at (11,10) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 50 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) ... 157

5.24 Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions 158

5.25 Heterogeneous log permeability field (section of SPE10 model layer 50) with one producer and one injector 160

5.26 Samples considered as potential well locations shown by red crosses. Injector location is fixed at location (10,1) (not shown here) and hence not sampled. ... 161

5.27 (a) Training samples randomly chosen for basis coefficient prediction (b) Cumulative energy of eigenvalues for pressure (red) and saturation basis (blue) ... 162

5.28 Test case with producer well at gridblock (28,50) 162

5.29 Pressure solution comparison at (a) Time = 506 days and (b) Time = 1026 days 163

5.30 Saturation solution comparison at (a) Time = 506 days and (b) Time = 1026 days 164

5.31 Relative error in pressure at time = (a) 506 days (b) 1026 days 165

5.32 True and ML predicted pressure basis coefficient comparison at time = (a) 506 days and (b) 1026 days 165

5.33 True and ML predicted pressure basis coefficient comparison at time = (a) 506 days and (b) 1026 days 166

5.34 Comparison of relative error in pressure due to orthogonal projection (left) and ML predicted solution (right) 166

5.35 (a) Test case 1 with producer well location at (36,26) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 167

5.36 (a) Test case 2 with producer well location at (28,50) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 168

5.37 (a) Test case 3 with producer well location at (14,12) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 169

5.38 Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions 170

5.39 Box plot of accuracies in pressure and saturation using predicted POD coefficients for increasing basis dimensions 171

5.40 (a) Test case 1 with producer well location at (9,29) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 172

5.41 (a) Test case 2 with producer well location at (26,48) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 173

5.42 Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions 174

5.43 Reservoir model with on injector at location (24,26) and red regions showing the regions from where producer 1 (top of the reservoir) and producer 2 (bottom of the reservoir) locations are sampled 177

5.44 (a) Test case 1 with producer 1 at (9,2) and producer 2 at (31,50) (b) Comparison of oil production rate for producer 1, (c) Comparison of water cut for producer 1, (d) Comparison of oil production rate for producer 2, (e) Comparison of water cut for producer 2, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 181

5.45 (a) Test case 2 with producer 1 at (20,3) and producer 2 at (32,49) (b) Comparison of oil production rate for producer 1, (c) Comparison of water cut for producer 1, (d) Comparison of oil production rate for producer 2, (e) Comparison of water cut for producer 2, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line) 182

5.46 Error in prediction of (a) Oil Production Rate and (b) Water Cut for producer 1, (c) Oil Production Rate and (d) Water Cut for producer 2, for all the test cases before and after the error correction of the solutions 183

LIST OF TABLES

TABLE Page

3.1 DEIM procedure 56

3.2 NPV and number of iterations for fine scale and reduced scale optimization using the 3 methods with IPOPT. NPV values are in million USD 65

3.3 NPV and number of iterations for fine scale and reduced scale optimization using the 3 methods with PSO. NPV values are in million USD 65

3.4 Relative error (%) in predicted solutions by posterior samples of basis functions 75

3.5 Indices of the pressure (38 basis) and saturation (58 basis) tail basis functions selected in one of the 20 samples 77

4.1 Geometric features corresponding to the basis well (a single well) and new wells for error map prediction. Features corresponding to basis wells correspond to $\zeta_{\mathbb{D}}$ and those for new wells correspond to ζ^* in equations (4.2) and (4.3) 93

4.2 Geometric features corresponding to the basis wells (concatenate solutions from 2 wells) and new wells for error map prediction. Add these to features in Table 4.1, for basis with snapshot concatenation 94

4.3 Physics based features corresponding to the basis well (a single well) and new wells for error map prediction. Features corresponding to basis wells correspond to $\zeta_{\mathbb{D}}$ and those for new wells correspond to ζ^* in equations (4.2) and (4.3) 100

4.4 Dictionary of basis used for prediction 103

4.5 Regularization parameters and hyperparameters chosen by 5-fold Cross Validation 106

4.6 Error map prediction accuracy for ANN and RF regression problem 111

4.7 Error map prediction accuracy for ANN and RF classification problem 112

4.8 Dictionary of basis used for prediction 114

4.9 Regularization parameters and hyperparameters chosen by 5-fold Cross Validation 115

4.10 Error map prediction accuracy for ANN and RF classification problem 116

4.11 Dictionary of ROBs used for predicting basis (obtained by snapshot concatenation) error maps 118

4.12 Snapshot concatenation based basis - error map prediction accuracy for ANN and RF regression problem and corresponding optimal hyperparameters 119

4.13 Snapshot concatenation based basis - error map prediction accuracy for ANN and RF classification problem and corresponding optimal hyperparameters ... 122

5.1 Geometric and physics based features for ML model construction corresponding to the well configuration 133

5.2 Hyperparameters chosen by 5-fold Cross Validation for Random Forest Regressor using 100 training samples 139

5.3 Homogeneous reservoir case 1: Average accuracy of oil production rate and water cut for all test samples 151

5.4 Hyperparameters chosen by 5-fold Cross Validation for Random Forest Regressor using 50 training samples 155

5.5 Homogeneous reservoir case 2: Average accuracy of oil production rate and water cut for all test samples 156

5.6 Heterogeneous reservoir case 1: Average accuracy of oil production rate and water cut for all test samples 164

5.7 Time (seconds) comparison for the two test cases between fine scale simulation and reduced order model with error correction 167

5.8 Heterogeneous reservoir case 2: Average accuracy of oil production rate and water cut for all test samples 172

5.9 Time (seconds) comparison for the two test cases (lower dimensional basis) between fine scale simulation and reduced order model with error correction ... 173

5.10 Time (seconds) comparison for the two test cases between fine scale simulation and reduced order model with error correction 177

1. INTRODUCTION AND LITERATURE REVIEW

My research is an amalgamation of petroleum engineering, applied mathematics and machine learning. The main idea of this work is to reduce the time complexity involved in the oil and gas field development decision making. A majority of time is invested in the simulation of complex fine scale reservoir models during the field development process. Thus, I propose to develop lower complexity models using the concepts from linear algebra and machine learning. This will eventually help reduce the cycle times for project evaluations without losing accuracy at reduced costs. In the future, it will also facilitate in applying these methods to explore problems that were otherwise unexplored by the industry due to the time constraint.

1.1 Background

Oil and gas industry is considered to be the biggest industry in terms of dollar value, catering to the needs of a large number of sectors such as manufacturing, transportation, medical, leisure etc. The demand for fossil fuels continues to rise, as it supports the majority of global energy needs. According to the U.S. Energy Information Administration's Annual Energy Outlook (AEO) [1], hydrocarbons (oil and gas) will still dominate energy supplies in 2050, shown in Figure 1.1.

The U.S. crude oil production has doubled since 2010 and by 2025 it is expected to increase 160% above the 2010 level. AEO2019 projects that the U.S. will export more energy than it imports in 2020. The primary reason for this is the production from unconventional reservoir (e.g., shale formations). However, the success of meeting the production targets and the world demand depends on the efficiency of reservoir management and the quality of subsurface information. Thus, it is the role of petroleum engineers to understand the physics of subsurface flow and make better field development decisions.

One of the areas heavily invested by the industry, that focuses on predicting flow of fluids

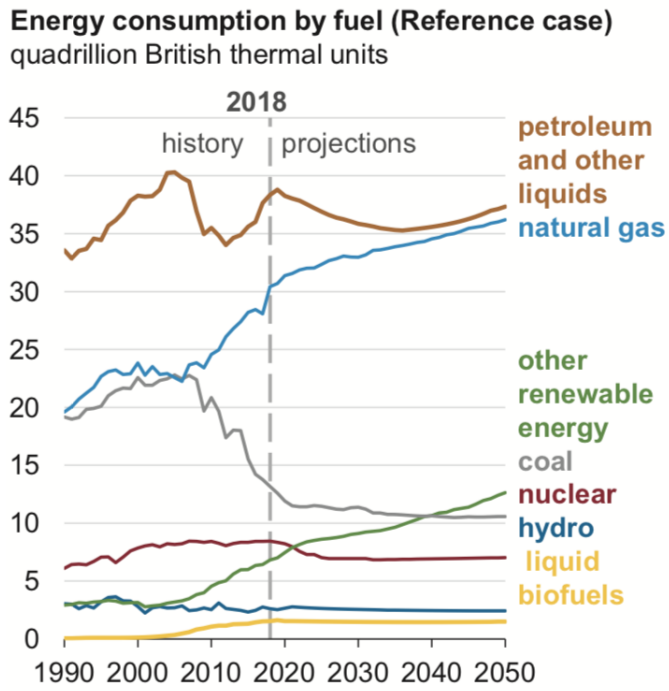


Figure 1.1: The U.S EIA prediction of energy consumption by fuel (reprinted from [1])

(typically oil, water and gas) in the subsurface porous media, is reservoir simulation. They rely on solving numerical partial differential equations that are deemed representative of flow physics in the reservoirs. These equations are solved for large geological reservoir models that represent the physical space and properties of the reservoir. Reservoir simulation assists in getting important information such as oil and gas production forecast, optimization of parameters to increase the value of the projects like number of wells required to drill, optimal completion of wells, updating reservoir model given the past data etc.

In efforts to improve the field management decisions a workflow was proposed in [2], called Closed Loop Reservoir Management (CLRM). CLRM is a combination of model-based optimization and data assimilation that aims at maximizing the performance of the reservoir. The improvement metric can either be percentage oil recovery or financial measures like Net Present Value (NPV), over the life of the reservoir by changing reservoir management from periodic to a near-continuous process. The workflow of CLRM involves well control

optimization based on the current geological knowledge, using these settings to collect data for some time period and finally performing history matching to integrate this data to update the geological models for consistency with observed data.

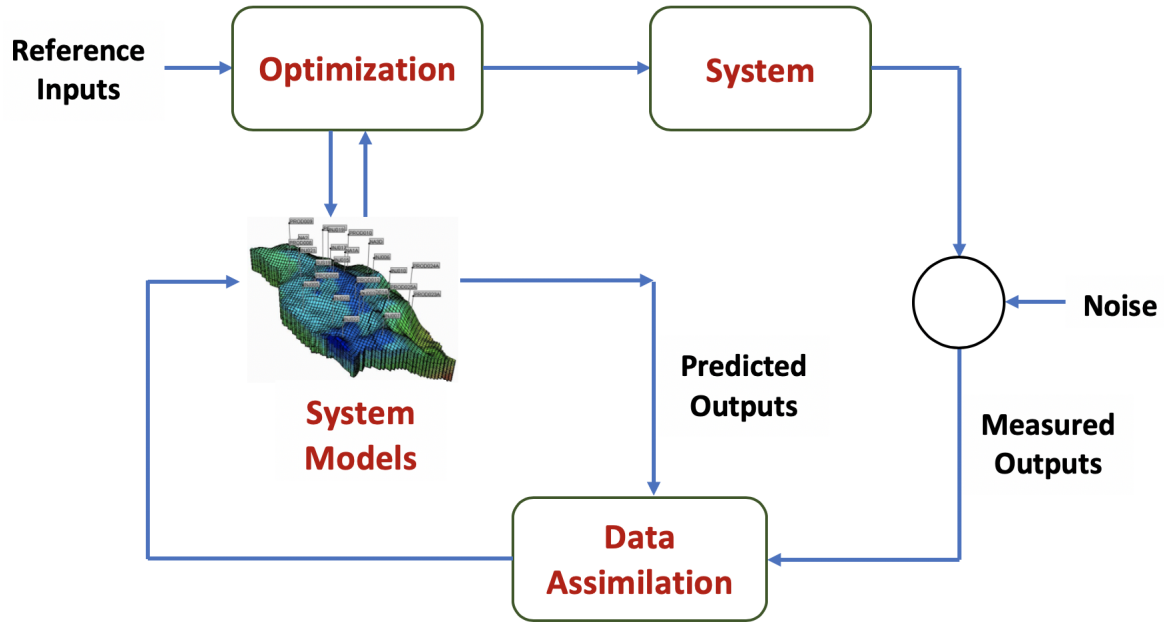


Figure 1.2: Key elements of the closed-loop reservoir management process

Figure 1.2 shows key elements involved in the CLRM workflow. *System* corresponds to the actual reservoir or well and *system models* represent the geological or well models that are constructed based on current knowledge. Multiple models are usually considered as system models to account for uncertainty in our knowledge of the subsurface. *Optimization* algorithms on the system models determine optimal control strategies of the wells for actual system and the output of the system like fluid production rates are observed by sensors in the field. Observed data from sensors are used to update the current models using *data assimilation* algorithms. This process is referred to as History Matching. These new system models are then used to design new input parameters using optimization algorithms. This process is repeated several times during the course of field development.

However, for CLRM, only the well controls (bottom hole pressure (BHP) or well injection rates) are considered as inputs to be optimized for the system during each stage of the cycle. Closed Loop Field Development (CLFD) which is an extension to CLRM, involves optimization of well number, type, location, and controls based on the current geological knowledge (system models) for inputs to the system. Then, sequentially new wells are drilled and data is collected which is used for history matching step to update geological information. This procedure is repeated until the optimal number of wells are drilled. These workflows like CLRM and CLFD, repeated during the development of reservoir, can help make better reservoir management decisions than the heuristic approaches.

The optimization and history matching steps in each of these workflows require repeated fine scale reservoir simulation runs, which corresponds to solving a discretized parametric PDE which is time dependent and highly non-linear in nature. And since the process is repeated several times during the development phase, accurate high-fidelity computational models can incur substantial computational costs. To avoid this bottle neck of costly evaluations, researchers have developed various surrogate modeling techniques that are approximations of high-fidelity models, which aim to provide large computational savings while preserving the accuracy.

The main approaches to surrogate model construction are based on upscaling or grid coarsening, data-driven methods, reduced physics models and reduced-order models (ROMs), as shown in Figure 1.3. Assumption of spatial homogeneity and grid coarsening attempt to reduce computational burden of the high-fidelity models by neglecting some physics or coarsening the mesh size [3, 4]. Reduced physics models try to approximate the physics to large extent that it compromises with the accuracy of the model [5]. Data-driven models that are non-intrusive in nature are based on supervised machine learning techniques to predict the input-output relationship of the models [6, 7]. Since, the data-driven methods do not take into consideration the underlying physics, they are usually less robust. They require a large amount of data for good accuracy but this is limiting in many engineering

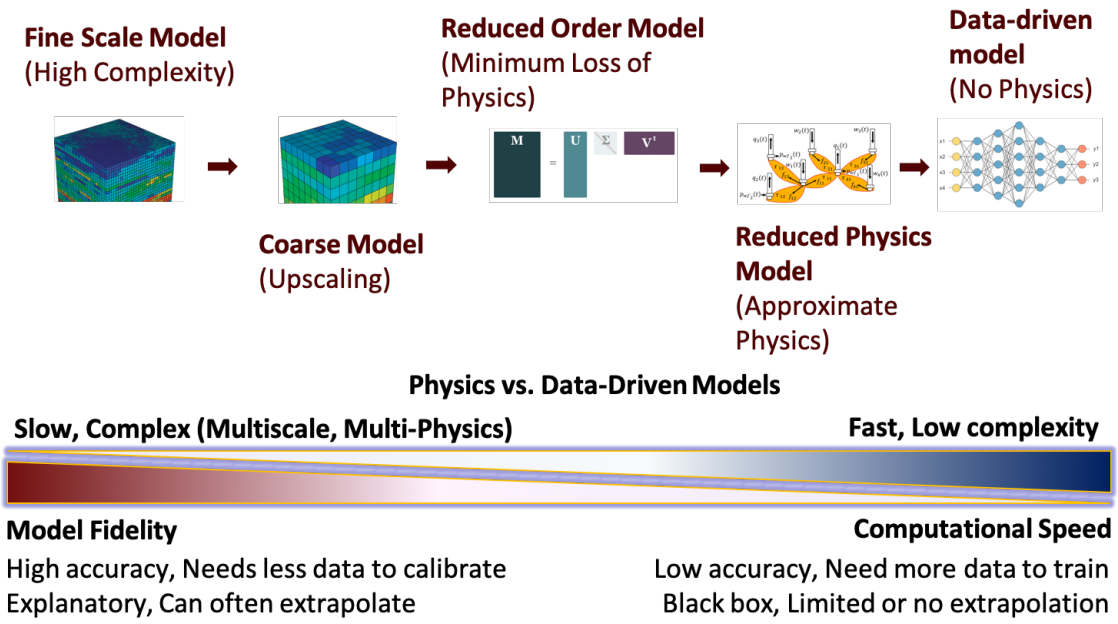


Figure 1.3: Hybrid Modeling Methods

applications. While ROMs are intrusive or non-intrusive projection-based methods that do not approximate the physics and rely on reducing the dimensionality of the state-space of fine scale models [8, 9], they are usually computationally cheaper than coarsened models and more robust than the data-driven models. Thus, for this research, we focus on developing model order reduction strategies for integrated workflows like CLRM and CLFD that can be used specifically for various optimization problems.

1.2 Literature Review

1.2.1 Model Order Reduction

High CPU burden is incurred from high fidelity simulations as result of large degrees of freedom obtained by spatial discretization of PDEs. Model order reduction (MOR) is a technique to extract dominant dynamic modes of the model. It captures the essential spatial or spatio-temporal characteristics of the underlying dynamics that can later be used for cheap forward simulations. The approach is to approximate a high dimensional state space

model to a lower dimension model with much smaller degrees of freedom that has the same response characteristics as the original system and computationally faster to simulate. Thus, Reduced Order Models (ROMs) show a potential of being operated in near real time. ROMs can be classified into two major categories namely intrusive and non-intrusive methods. Intrusive methods require modifications in the source code of the simulator used, whereas, non-intrusive methods do not require access to the simulator and hence can be used with any commercial simulators where accessing the source code is difficult.

A widely used approach to yield ROMs is Galerkin projection. In ROMs, the projection basis can be obtained through balanced truncation [10], Krylov subspaces [11], reduced basis method [12] or proper orthogonal decomposition (POD) [9]. Galerkin projection uses the same state projection basis and test basis, which when applied to the original equations (strong or a weak form) or on the spatially discretized system, yields a low-dimensional algebraic system for steady problems or an ordinary differential equation (ODE) system in time for dynamical problems. The most widely used technique for PDE systems is POD, in which we generate the solutions from discretized full-order model at selected time instances (snapshots) to get the approximating subspace (spatial modes). POD provides computational advantage at the linear solver level. Non-intrusive projection based MOR method includes methods like dynamic mode decomposition which captures spatio-temporal modes of the dynamical system [13, 14] and capacitance-resistance model [5]. ROMs have been applied in various applications including aerodynamics [15, 16], semiconductors [17], electrochemical [18], electrothermal [19] etc. In the Oil and Gas industry, MOR, especially using POD-based techniques, has shown to be a viable way of mitigating the large-scale nature of the reservoir simulation models by preserving the accuracy and accelerate the computations in some of the subsurface applications. The concept of MOR for reservoir simulation applications is shown in Figure 1.4. For most engineering applications, the major problem lies in building ROMs for design optimization, control and uncertainty quantification etc., where they lack robustness with respect to parameter variations. This is a major limitation since it allows

the use of ROM at only the parameter value at which it is computed.

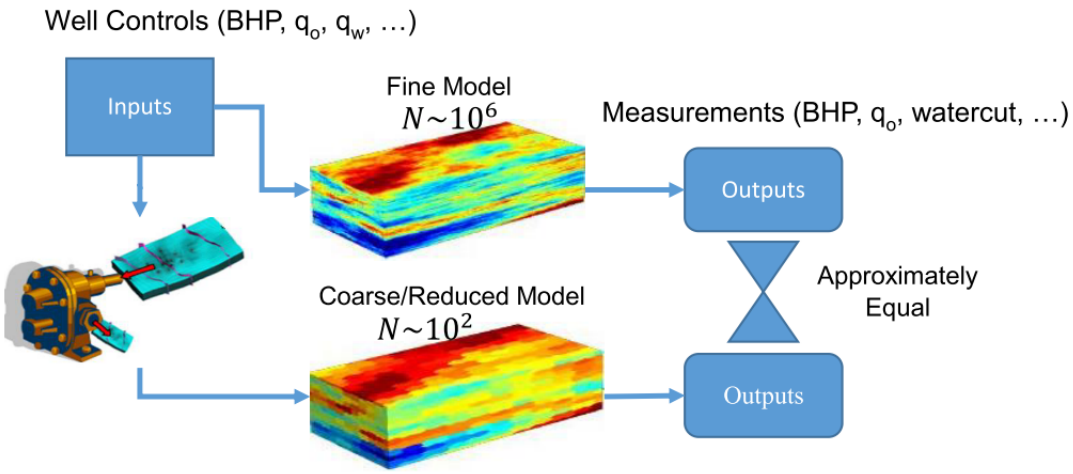


Figure 1.4: Model Order Reduction concept for reservoir simulation models (reprinted from [20])

My research here focuses on a class of problems for which the reservoir simulation equations representing the dynamics of reservoir depend on a set of parameters, and the goal is to characterize system response for different value of parameters. These parameters may enter the models in many ways, representing, for example, reservoir properties, well geometry, initial conditions, and boundary conditions. For my research, these set of parameters correspond to changing boundary conditions of the system i.e., well controls and well locations, for well control optimization and well placement optimization problems respectively. This parametric dependence presents a unique set of challenges for model reduction. In order to build a reduced order model for a new set of parameters, it is required to run an expensive full model again and thus becomes infeasible. Hence, the desired approach is to generate a reduced model or models, that approximate the original full-order dynamical system with high fidelity over a range of parameters. This is the goal of parametric model order reduction (PMOR), which is our area of focus for both the problems considered. We use POD-based

methods for both, changing well control and changing well placement problems.

PMOR methodology is pictorially shown in Figure 1.5, which involves offline construction of a database $\mathbb{D}$ of representative parameters and then using this database to compute local/global ROM or ROB using POD for a new parameter in the online stage [21].

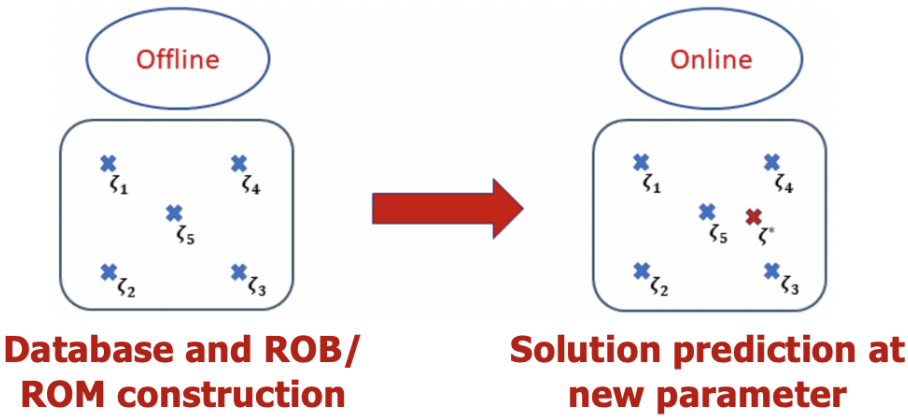


Figure 1.5: PMOR methodology where the offline stage computes fine scale simulations at sample points in database and ROMs or ROBs are constructed, and in the online stage, solutions are predicted at a new parameter ζ^* based on the information from offline stage (adapted from [21])

However, a number of challenges are associated with application of POD to dynamic, non-linear parametrized PDEs which mainly include (1) constructing a basis that is accurate across parameter space, (2) limited data availability, (3) high-dimensional parameter spaces, and (4) efficient construction of reduced-order non-linear functions of the state variables for different parameters. These challenges have been addressed across many engineering domain applications for building reduced order models or reduced order basis (ROBs) for parametric systems. The main approaches include (1) using a global ROM constructed offline that is meant to be robust in the entire parameter space, (2) using database of local reduced models or basis and then quickly interpolating the ROBs or ROMs at the unknown parameters of interest, and (3) approximating the snapshot matrices at new parameter points.

1.2.2 Model Order Reduction Background for Well Control Optimization

The goal of well control optimization problem is to seek for optimal BHP schedule of the wells that maximizes an objective function like financial gain of a project or maximizing the recovery from a field. This process of finding an optimal solution usually requires hundreds to thousands of fine scale simulation runs depending on the size of the reservoir. In order to tackle this problem of a computationally expensive strategy, initial work in reduced order modeling was applied in the oil and gas industry for the well-control optimization problem using POD [22, 23]. Since POD projects the state-space solution to a subspace of much lower dimension, it achieves a significant speedup in solving system of equations. Solution at any time during forward simulation can then be estimated by a linear combination of the POD basis. When POD is applied to a non-linear problem, evaluation of the non-linear terms requires to project back to the fine scale solution with similar computational cost as the original system. However, there is not much speed-up reported by using POD since it only targets the linear solver. Full Jacobian and residual matrices are still computed during each iteration in simulator.

In order to overcome this limitation of POD, other techniques were developed for fast computation of non-linear terms in the reservoir simulator. One such technique developed was TPWL applied in conjunction with POD for production optimization [24]. Before subsurface application, POD-TPWL was applied in other areas where it showed significant speed-ups [25, 26, 27, 28]. The idea of TPWL for production optimization application is to save the pressure and saturation states and Jacobian matrices generated during a few training runs to predict new solutions. This is accomplished by linearizing the states around previously saved states using Taylor series expansions. Computational speed-ups of the $\mathbb{O}(10^2 - 10^3)$ in the online testing stage were shown in [29, 24] . Thus, TPWL provides computational benefit in quickly predicting the non-linear Jacobian and residual and POD provides advantage at the linear solver level. However, due to linearization, many non-linear terms are not approximated accurately and thus lead to loss of accuracy and instabilities in the

final solution. This method also requires saving a large number of states and huge Jacobian matrices which demands a lot of hardware capability even for small sized problems. Later, TPWL was extended to include the quadratic terms for more accurate representation of non-linear functions and thus called Trajectory Piecewise Quadratic (TPWQ) method [30]. This method proved to be more accurate as compared to TPWL but the overhead time increases significantly because it involves third-order matrix-tensor products. It showed potential as a good error indicator for TPWL method during production optimization. A machine learning based framework for surrogate model error prediction was proposed which was implemented for POD-TPWL application on reservoir waterfloooding example [31].

The method used for my research to reduce the cost associated with non-linear function computations in well control optimization problem is Discrete Empirical Interpolation Method (DEIM) [32]. This method is a variant of Empirical Interpolation Method (EIM) that aims at reducing the dimensions of discretized parametric PDEs by computing the non-linear terms at discrete locations in the spatial domain and then interpolating them to the rest of the locations using projection based interpolation. This helps prevent projecting back to the fine scale to evaluate non-linearities. The method DEIM uses to select the locations at which to compute the non-linear functions is based on a greedy algorithm. It differs from the method used in Missing Point Estimation (MPE) method [23] which is just based on POD basis. Whereas, DEIM uses different POD basis for states and the non-linear terms. POD-DEIM has been widely used across different engineering domains [33, 34, 35, 20]. Localized DEIM was introduced in [36, 37] using concepts in Machine Learning, that computes several local subspaces, tailored to a particular region of characteristic system behavior. Higher accuracies can be obtained by using POD-DEIM since DEIM evaluates non-linear terms at specific locations, and then interpolate these terms at other locations. However, less speed-ups have been reported for POD-DEIM as compared to POD-TPWL as a result of linearization performed in the latter and it is also more invasive with respect to the simulator as compared to POD-TPWL. Moreover, POD-DEIM still lacks a successful implementation

on a full well control optimization procedure which is the main idea of a part of my work here. Other methods were later implemented for this application of changing well controls, such as, POD - Trajectory based DEIM (TDEIM) [38] which is a combination of TPWL and DEIM techniques to utilize their benefits and avoid their shortcomings; bilinear approximation and quadratic bilinear formulation [39, 40] and Gauss-Newton with Approximated Tensors (GNAT) method [41].

In this research work, we use the strategy of global PMOR when using POD-DEIM for well control optimization problem, where we construct a global reduced basis for the entire parameter domain corresponding to the BHP profiles of the wells. This global basis is obtained by concatenating the snapshot matrices corresponding to different representative system parameters (BHP profiles in this case). The case of multiple training and retraining for global basis construction in the context of well control optimization can be found in [42, 38]. However, designing a good training set for the entire control optimization has still been a challenge, which is tried to address in this work. For a global basis, the parameters that are representative of the parameter domain are chosen a priori or using adaptive greedy algorithms [43]. The a priori sampling of parameters is fast but can include unnecessary samples or miss important samples. The adaptive sampling strategy is more accurate but computationally expensive as it needs repeated evaluations to look for parameters that produce the maximum ROM error. We introduce a novel training procedure during the optimization run to obtain a global ROB (Reduced Order Basis) which will be explained in detail later.

1.2.3 Model Order Reduction Background for Well Placement Optimization

Well location optimization in the context of surrogate or low complexity models have been developed in [44, 45]. However, until now, for projection based reduced order models, the focus mainly has been developing them for well control optimization problem to the best of our knowledge. But these ROM workflows were not meant for integrated schemes like Closed Loop Field Development which involve changing reservoir or well configurations. So

here in the second part of research, our focus is to address this issue specifically developing Reduced Modeling strategies for changing well locations during well location optimization problem.

The choice of MOR technique for this problem is POD. As discussed before, there are a number of challenges associated with POD for changing system parameters. Considering the complexity and after some observations for the changing well locations, we propose local and global PMOR techniques for developing ROMs. Researchers from other communities have developed various strategies for local parametric ROM construction. One approach to construct ROMs/ROBs for different parameters, is to get a database of local ROMs, ROBs or snapshot matrices at different parameters offline and then, interpolate these matrices in the online stage, computed cheaply. [46] applied a ROM adaptation technique for a new parameter for Linear Time Invariant (LTI) systems based on interpolation of local ROMs in a tangent space to a Grassmann manifold. Similarly, ROBs for flight parameters are interpolated in the tangent space in [47].The challenges here are construction of database that still requires large number fine scale simulations, choosing representative parameters for ROB and ROM interpolation for high dimensional parameter space and its limitation to LTI system application. MOR for dynamic linear and non-linear PDEs was proposed in [48], that uses manifold learning techniques to interpolate the snapshots evaluated at different parameters. Thus, as opposed to the global basis, these interpolation methods to get a ROB/ROM for new parameter are not obtained directly from the underlying model. They try to get accurate reduced basis or models during the online phase with much lower dimensions as compared to global method. The accuracy however depends on the interpolating techniques used. These methods have been employed for parameters of lower dimensional space.

Other ROM approaches were based on error estimates of the ROM. ROM error surrogate (ROMES) [49] aimed at developing kriging error model of ROMs as a function of error indicators like rigorous error bounds and reduced-order residual norms. The multi-fidelity correction (MFC) models [50, 51] were developed for lower fidelity models in context of

optimization that predicts the model error given the system parameters. Both ROMES and MFC methods use a global reduced-order model with a fixed basis dimension as the inputs. In [52], an error based framework to predict the characteristics of the local parametric reduced order models was developed called multivariate prediction of local reduced order models (MP-LROM). It consists of mapping the parameters of the system and dimensions of the ROM to ROM error. All these methods take advantage of concepts in Machine Learning (ML) to build the complex input-output relations. Our first methodology here is based on the observations that a dictionary of local reduced order basis at a few well locations may be sufficient to describe the dynamics for the entire domain of parameters (well locations in our case). The workflow relies on prediction of time averaged ROM errors at different production well locations (error maps) using ML algorithms for each of the local ROB which is an extension of work carried out in [52].

For complex non-linear problems with non-affine parameter dependence, as mentioned before, direct (simulator invasive) use of POD do not provide major computational advantage. This is because of the cost to compute projection coefficients of the non-linear functions that depend on the dimension of high fidelity model. As the dimension of a system increases, the computational advantage of using pure POD method decreases. Thus other techniques were developed in conjunction with POD method as described in the MOR for well control literature review. Other alternative methods were also developed that still uses pure POD implementation, but the coefficients are not obtained by projection process, rather by interpolation over the parameter domain of reduced order basis [53]. Since ROBs belong to non-linear matrix manifolds, standard interpolation techniques cannot be applied as they would not preserve the characteristics of the ROBs, unless a large number of samples are used [54, 46]. To overcome this issue, researchers have developed ROB method using POD and then employed Machine Learning techniques to interpolate the basis coefficients. [55] developed POD-NN where he used Neural Networks (NN) for high dimensional coefficient prediction. [56] compared performance of different ML techniques to predict the basis

coefficients for an aerodynamic and a structural problem. It considered the use of particular solutions in the POD expansion as a way to embed physical constraints that must be preserved during coefficient prediction. These methods however, did not consider the time-dependent problems. Such a non-intrusive ROM scheme have been applied here for changing well locations which is highly non-linear in nature and also involves time dependency. The proposed method is based on global ROB for the entire parameter domain.

1.3 Scope of Work

The two main objectives of this work are the development of a POD-DEIM workflow when integrated with a production control optimization problem and development of a ROM method for changing well placement during well location optimization problem, to facilitate in filling the gap of fast simulations using projection based ROM in the field development optimization procedures. Figure 1.6 shows the main contributions of this work where, the blue path shows research direction with green boxes being the areas focused on and the red path is yet to be developed in the context of MOR. We note here that, for the current work, we look for developing reduced models for changing well locations as a precursor to well location optimization. We do not employ the proposed methods on well location optimization problem in this work. Following are the main contributions that are addressed in this research work:

- We first introduce a new control optimization procedure that takes into account the polynomial control parameterization. POD-DEIM which was introduced for changing well control strategy [20] did not apply it to a full production optimization problem. Thus, we use the new production control parameterization strategy and apply global PMOR using POD-DEIM workflow for a full production control optimization problem to assess its performance as compared to when used with traditional control parameterization. The global ROB is constructed that takes advantage of the global-local optimization procedures used.

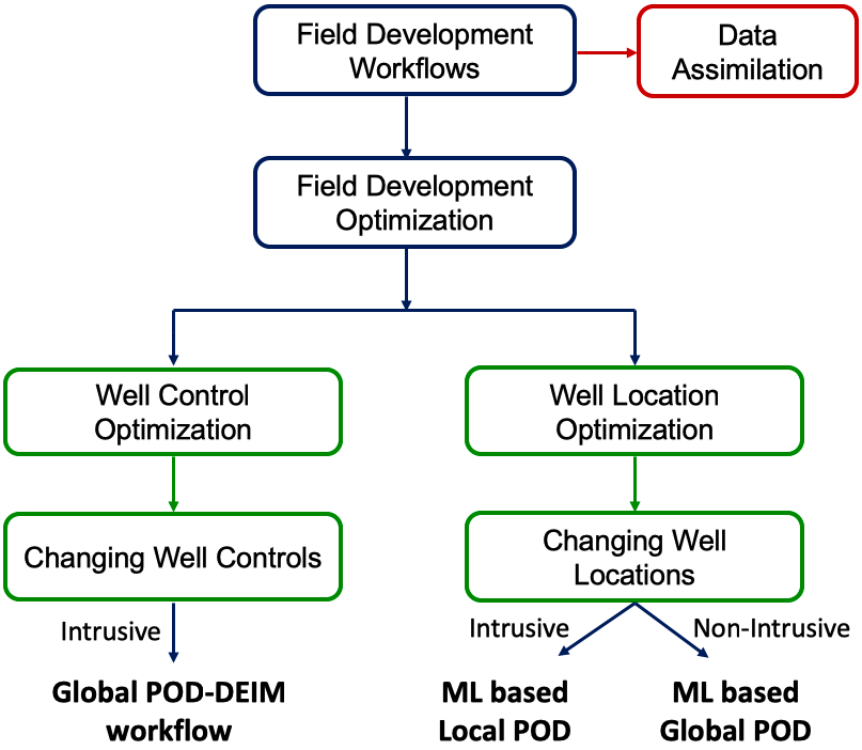


Figure 1.6: Scope of Work. Green boxes are the area of focus.

- Next part of the research is the major focus of my work, i.e. development of MOR methodology for changing well locations that has not been developed yet. We first develop a POD-based local parametric reduced modeling strategy with the aim at running a minimum number of fine scale simulations and still construct reduced order basis over the domain of parameters. We use concepts from flow diagnostics and develop a machine learning framework that estimates the error of a reduced basis at all the well locations in the reservoir and thus construct an error map, which eventually will help us adaptively pick the most accurate ROB. This methodology, however, was applied for the cases of small simulation run times considering the challenges of this problem. This ROM technique is simulator invasive in that it requires the projection of fine scale non-linear functions.

- In order to address the limitations from the previous proposed method, we develop

a global PMOR methodology for changing well locations that is non-intrusive to the reservoir simulator source code. The method is based on POD, but uses ML techniques to learn a map between the input parameters, that describe the well locations in the reservoir, and the output basis coefficients. This idea was proposed in other areas for time independent problems, but we extend it to time dependent problems for reservoir simulation. The quantities of interest (QoI) are then corrected using a error surrogate model similar to the concept used in [31]. This POD-ML based technique provides significant speedup as it does not involve any non-linear function evaluations and the time steps are independent of each other, hence can make use of the parallel computing resources effectively. This method will be applied to changing well locations for reservoir waterflooding problem to assess its accuracy and feasibility.

1.4 Dissertation Outline

The dissertation proceeds as follows. In Section 2, we begin with reservoir simulation, describing discretized governing equations for incompressible and slightly compressible oil-water flow in the subsurface. We describe this governing equations as a parametric system to develop the PMOR workflows for optimization. Then, we introduce the concept of POD for basis construction with concepts in controllability and reachability. We will also briefly introduce the topics of Galerkin projection error estimation and stability.

In the Section 3, we introduce the well control optimization formulation and propose a new polynomial control parameterization strategy. Then we briefly describe about the optimization strategies used for waterflooding optimization. This work was pursued in collaboration with Nadav Sorek, a previous Ph.D. colleague in our group. This production optimization strategy is then implemented with the POD-DEIM procedure where we introduce a new global PMOR training strategy to be used for the entirety of optimization run without retraining. The proposed workflow is validated with a case study.

Section 4 introduces the multi-parametric machine learning framework for local parametric ROBs for changing well location problem based on intrusive POD technique. This

framework estimates the error maps corresponding to each ROB in the database and then select the best ROB for a new well location. We build this technique by analyzing a few observations first. Then we introduce the Machine Learning techniques used and provide a brief understanding on them. Later, we formulate the problem required to be solved in terms of regression and classification problem. Then, the features representing the well locations are introduced using a case study and some concepts in flow diagnostics. Finally, we apply the findings for some case studies of waterflooding and changing well locations to analyze the framework.

In Section 5, we develop a non-intrusive global PMOR strategy that is based on POD for changing well locations, developing on the shortcomings of the previously introduced method. This method is also based on using machine learning techniques for constructing a map between input features and the output POD expansion coefficients. We first formulate the problem and describe its advantage over the intrusive POD method. Then, we apply this method on a few case studies to predict the QoI. Later, we also construct an ML based error correction model in the states at well location to correct for the predicted QoI.

We conclude the dissertation in Section 6, with a summary and discussion about the future work, especially for parametric reduced modeling problems like well location and history matching and the direction of using machine learning techniques.

2. RESERVOIR SIMULATION AND PROJECTION BASED MODEL ORDER REDUCTION FOR PARAMETRIC SYSTEM

In this section, we first introduce the governing equations for two-phase flow in the reservoir for incompressible and slightly compressible fluids, as they will be used in the later part of the research. We provide details of discretization and the boundary conditions corresponding to well control and well locations. We then discuss in detail about the projection based model order reduction techniques, especially POD (Proper Orthogonal Decomposition) method, which is used throughout the scope of this work.

2.1 Reservoir Simulation

The subsurface flow models in porous media are derived by coupling equations representing conservation of mass, momentum, and energy as a function of pressure, temperature, and saturation or fraction of each phase or component [57]. This results in space and time dependent partial differential equations. The two phase flow equations used here, are generally used for scenarios like waterflooding, where water is injected into the reservoir from injector wells to help provide pressure support to the reservoir and increase the oil production rate and, ultimately the oil recovery.

2.1.1 Governing Equations

The governing equations for a two phase isothermal immiscible oil-water system is obtained by combining the mass conservation law and the Darcy's law for two phases. This governing equation considering the gravity effects is a PDE of the form:

$$\frac{\partial(\varphi\rho_l S_l)}{\partial t} = \nabla\cdot(\rho_l \mathbf{v}_l) - \widetilde{m_l} \tag{2.1a}$$

$$\mathbf{v}_l = \lambda_l K(\nabla p_l - \rho g \nabla z) \tag{2.1b}$$

where, l indicates liquid, oil (o) or water (w), φ is the porosity of the reservoir, ρ_l indicates density of the fluid per unit volume, and $\widetilde{m_l}$ (mass flow rate) $= \rho_l q_l$ denotes the external sources and sinks, q_l being the volumetric flow rate. $\mathbf{v}_l$ here represents the Darcy velocity of liquid (l). $\lambda_l = k_{rl}(S_l)/\mu_l$, where μ_l, k_{rl} are the viscosity of fluids and relative permeability of fluids, respectively and K is the absolute permeability. p_l and S_l are the pressure and saturation of each phase respectively. g is the gravity acceleration constant and z accounts for vertical coordinates. Equations (2.1a) and (2.1b) for oil and water, along with the constraints, saturation equation ($S_o + S_w = 1$) and capillary equation ($p_{cow} = p_o - p_w$) completes the oil-water model. In these equations, p_o and S_w are considered to be the primary unknowns, and once they are solved, the rest of the unknowns p_w and S_o are computed easily from the constraint equations.

2.1.2 Incompressible and slightly compressible flow

For an *incompressible flow*, the phase densities ρ_l and μ are constant. Under these conditions and constant porosity φ, the governing equations for incompressible flow takes a simplified form of 2.1 as:

$$\phi \frac{\partial S_l}{\partial t} = \nabla \cdot (\mathbf{v}_l) - q_l \tag{2.2a}$$

$$\mathbf{v}_l = \lambda_l K(\nabla p_l - \rho g \nabla z) \tag{2.2b}$$

where, q_l indicates the volumetric flow rate. After some manipulations of these equations, it can be shown that saturation is coupled to pressure through the total Darcy velocity of the fluids.

For *slightly compressible flow*, $c_l \Delta P \ll 1$, where ΔP is the characteristic pressure drop across the system. Thus, it is assumed that the fluid compressibility is small and remains constant within the pressure range of interest. The properties of the fluids and rock such as viscosity (μ_l), density (ρ_l) and hence formation volume factor (B_l), and porosity (φ) are dependent on pressure. Thus, in the equation 2.1a, the accumulation term is expanded and

the fluid and rock compressibilities are used to rewrite the derivatives of phase densities and porosity. In the incompressible case, total Darcy velocity is the only quantity that couple the saturation equation to fluid pressure, however, for slightly compressible case we have coupling through the porosity and density which are pressure dependent. This makes the slightly compressible flow scenario more challenging to solve as compared to the incompressible flow problem.

We use fully implicit formulation using finite difference discretization to solve these incompressible (Equation 2.2) and slightly compressible flow PDEs (Equation 2.1). For our work, we neglect the effects of capillary pressure, and hence, $p_o = p_w$.

2.1.3 Discretized parametric equations for well control and well location

In order to solve the PDEs in equations (2.1 and 2.2), we perform spatial discretization of these equations that result in the following parametric system of ODE in time to solve for the state variables p_o and S_w represented by $\mathbf{x}$:

$$-\mathbf{D}(\mathbf{x},\zeta)\dot{\mathbf{x}}(t,\zeta) + \mathbf{T}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) + \mathbf{G}(\mathbf{x},\zeta) + \mathbf{Q}(\mathbf{x},t,\zeta) = \mathbf{R}(\mathbf{x},\zeta) \tag{2.3}$$

$\mathbf{x}(t,\zeta) = [P_o, S_w] \in R^{N_d}$, is a state vector with N_d degrees of freedom and $\zeta \in \mathbb{P} \subset R^{N_\zeta}$ is a vector of parameters. $\dot{\mathbf{x}}(t,\zeta)$ is the derivative of the states with respect to time, $\mathbf{D}$ is the accumulation matrix, $\mathbf{T}$ is the transmissibility matrix, $\mathbf{G}$ is the gravity vector, $\mathbf{Q}$ contains the (volumetric) sources and sinks terms (q_l) and $\mathbf{R}$ is the residual vector. We satisfy $\mathbf{R} = 0$ upon convergence to the solution for each time step. We consider the fully-implicit procedure to solve equation (2.3). The source/sink volumetric term correspond to production/injection from a well and is written as the Peaceman equation:

$$q_l = W I_l \, \lambda_l(S_l) \, (p_o - p_{wf}) \tag{2.4}$$

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

The parameter ζ of the system depends on the problem under consideration. Here, we are interested in well control and well location optimization and hence we define the system parameters for these two problems. For well control optimization, $\zeta = p_{wf} \in R^{N_t}$ for each well, where N_t is the number of simulation time steps. For well placement optimization, ζ represents the indices in sparse vector $\mathbf{Q}$, corresponding to spatial location of the well to be optimized in the source/sink vector, which is the main focus of this paper.

$$\zeta = \begin{cases} p_{wf} \ \ or \ \ q_{inj}, & \text{for well control optimization} \\ indices \ \ in \ \ vector \ \ \mathbf{Q}, & \text{for well location optimization} \end{cases} \tag{2.5}$$

Using the fully implicit method, for a given system parameter, at each time step, this non-linear system of equations (2.3), is solved using Newton's method:

$$\mathbf{J}(\zeta)^{n+1} \, \boldsymbol{\delta}(\zeta)^{n+1} = -\mathbf{R}(\zeta)^{n+1} \tag{2.6}$$

$$\mathbf{x}(\zeta)^{n+1} = \mathbf{x}(\zeta)^{n} + \boldsymbol{\delta}(\zeta)^{n+1} \tag{2.7}$$

Here, n and $n+1$ represent the previous and current time levels respectively and $\mathbf{J}^{n+1} = \frac{\partial \mathbf{R}^{n+1}}{\partial \mathbf{x}^{n+1}}$ is the Jacobian matrix. We satisfy $\mathbf{R}^{n+1} = 0$ upon convergence to the solution for each time step, thus requires multiple newton iterations (see ([58]) for detailed derivation). The size of the Jacobian matrix is N_d x N_d and that of the residual is N_d x 1. The degrees of freedom in a reservoir simulation usually ranges from hundreds to millions. Thus, the size of the system matrices (Jacobian and Residual) significantly increases the time to solve the linear system in the Newton method. For a detailed explanation on oil-water flow simulation, one can refer to some classical literature ([57, 58, 59, 60]).

2.2 Controllability of reservoir flow

As an introduction to MOR, we begin with some system theoretical concepts of subsurface flow. It provide insights on significant level of reduction that can be obtained by MOR. The control of the positions of fluid front and phase arrival times during optimization are related to the capacity to control streamlines by manipulating well controls [61, 62]. The controllability of streamlines are strongly related to the controllability of pressure in the reservoir, since streamlines are governed by the spatial derivatives of the pressure field.

A system is fully state controllable if an external input $\mathbf{u}$ has an ability to move the internal state of the system from any initial state $\mathbf{x}_1$ to any other final state $\mathbf{x}_2$ in a finite time interval $[0, T]$, i.e., $\mathbf{x}(T, 0, \mathbf{x}_1, \mathbf{u}) = \mathbf{x}_2$. The concept of controllability for reservoir simulation can be considered by borrowing concepts from linear systems theory [63]. The reservoir simulation flow equations are non-linear, but with some assumptions for simplified cases (e.g., single phase slightly compressible flow), it can be represented by a linear expression as:

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{u}_n \tag{2.8}$$

where $\mathbf{x}$ represents pressure state and n represents the time step ($n = 0, 1, ..., n-1$). u is the control matrix consisting of BHPs or rate controls of the wells. $\mathbf{A} \in R^{N_d \times N_d}$ is the system matrix and $\mathbf{B} \in R^{N_d \times N_w}$ is the input matrix that can be adjusted to account for changing well locations. Here N_w is the number of wells and N_d that corresponds to total degrees of freedom is equal to the number of grid blocks for single phase system since, $\mathbf{x}$ correspond to pressure at each grid block.

It can be shown that such a system is fully controllable if the controllability matrix defined as

$$\mathscr{C} = [\mathbf{B} \;\; \mathbf{AB} \;\; \mathbf{A}^2\mathbf{B} \;\; ... \;\; \mathbf{A}^{n-1}\mathbf{B}] \in R^{N_d \times N_d N_w} \tag{2.9}$$

has a full rank. Also, controllability Gramian can be used for qualitative analysis of control-

lability which is defined as:

$$\mathscr{P} = \mathscr{C}\mathscr{C}^T \tag{2.10}$$

Fully controllable system requires the controllability Gramian to be full rank i.e., the controllable subspace $\mathbf{x}^{con} = \text{im}(\mathscr{C}) = R^{N_d}$. If $\text{rank}(\mathscr{C}) < N_d$, then $\mathbf{x}^{con} = \text{im}(\mathscr{C}) \subset R^{N_d}$.

However, this rank test analysis gives an idea if the system is full state controllable or not, it does not provide any quantitative measure of controllable states. This information can be obtained by applying singular value decomposition (SVD) of the controllability matrix $\mathscr{C}$:

$$\mathscr{C} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T \tag{2.11}$$

Here, $\mathbf{U} \in R^{N_d \times N_d}$ is the left singular matrix, that represent an ordered set of linear combinations of the states $\mathbf{x}$. $\mathbf{\Sigma} \in R^{N_d \times l}$ is a diagonal matrix with singular values in decreasing order. It can be shown that the states in $\mathbf{x}^{con}$ that require the least energy to reach i.e., are the most controllable, correspond to large singular values and have a significant component in the span of the singular vectors $\mathbf{U}$. Thus, $\mathbf{U}$ has the singular vectors that are decreasingly controllable. l in the dimension of singular values matrix represent the number of singular values with magnitude above machine precision. In reservoir simulation, typically $l \ll N_d$, where N_d is the total degrees of freedom for a full order system. This was studied for single phase flow using BHP and rate controls for the wells [64], that showed the controllable subspace to be of much lower dimension than the state space, and the most controllable pressure gradients are those close to the vicinity of the wells. These observations, however, are valid only locally in the state space, which refers to controllability of the states close to a reference state ('reference trajectory' in the state space).

For two phase flow, similar restrictions exist, in that the pressure controllability results are similar to single phase flow problem and the controllability of saturations is restricted close to the fluid fronts [65]. To investigate the controllability for non-linear models, empirical Gramians have been used [66]. Empirical Gramians cannot represent the global behavior of

the non-linear model but is used to analyze these models for specific inputs and specific initial conditions. Empirical controllability Gramians are computed by collecting a large number of state 'snapshots' for different input trajectories (control vector). It can be written as:

$$\mathscr{P}_{emp} = \sum_{j=1}^{t} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_s \end{bmatrix}^j \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_s^T \end{bmatrix}^j \tag{2.12}$$

where s is the number of snapshots and t is the number of different control vectors. Similar to the linear case, it is possible to take SVD of the empirical Gramian $\mathscr{P}$ to compute the local controllable subspace by taking the first l singular vectors. It is usually observer that for most cases $l \ll N_d$ i.e., the controllable subspace has much lower dimensions as compared to the system state space. This is the motivation behind using MOR for reservoir simulation applications like well control and well location optimization.

2.2.1 Case study to analyze controllability properties for changing well controls and well locations

As we have briefly discussed about the theory on controllability of reservoir models that shows the motivation behind using reduced order models, we now provide a simple case study for analyzing the controllability properties for changing well control and well locations.

We consider a case of 2D homogeneous reservoir having dimensions $20 \times 20 \times 1$ (400 grid blocks) with 1 producer and 1 water injector. The model is discretized using Cartesian grid of size $30ft \times 30ft \times 30ft$. The porosity of the field is set constant to 0.2. Figure 2.1 shows the permeability field and the well locations with injector at $(20, 1)$ and producer at $(1, 20)$ gridblocks. Equation (2.3) is used to solve for the states during each simulation time step. We neglect the capillary pressure effects and consider an incompressible flow of oil and water. The initial saturation of oil and water is 0.8 and 0.2 respectively. The Corey-type relative permeability curves with exponent of 2 are used for both the fluids.

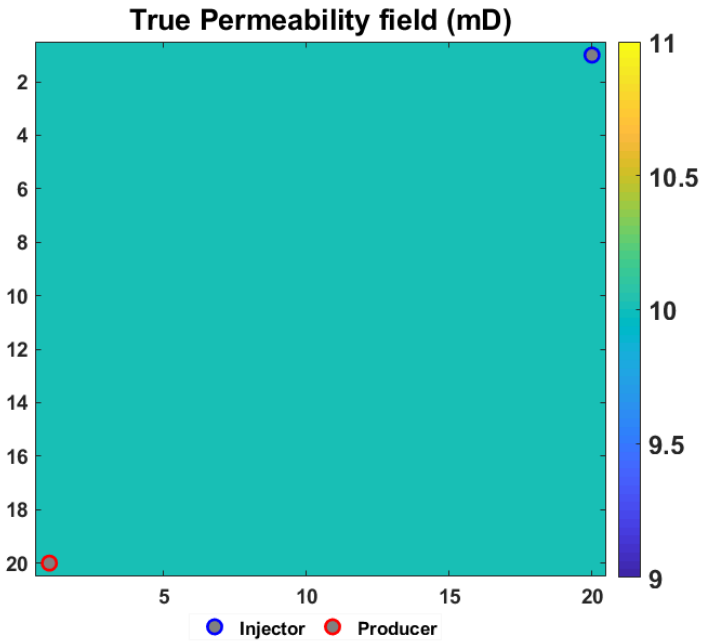
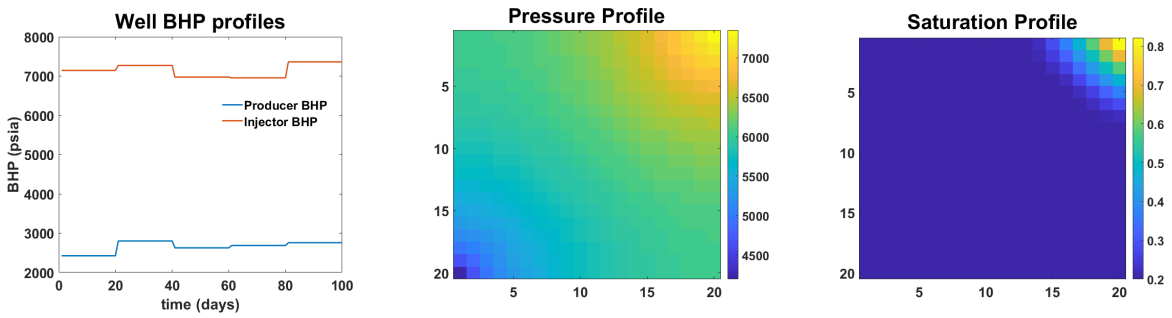


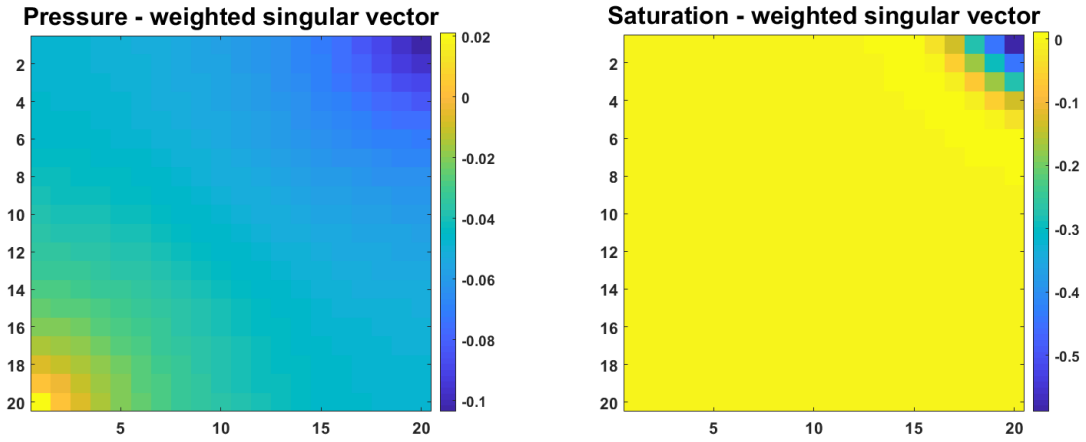
Figure 2.1: Homogeneous reservoir permeability field with 1 producer and 1 injector

Let us first consider the problem of well control change in the wells. So the first step to quantify the controllability is to construct an empirical controllability Gramian (for non-linear models). This is done by collecting the states in a single matrix as depicted in equation (2.12). The BHP control profiles for injector and producer are shown in Figure 2.2a. The simulation is run for a small time of 100 days. We then take the SVD of empirical Gramian to get first l singular vector and their corresponding singular values. $l \ll 800$ (400 pressures and 400 saturation values for each gridblock), which is determined by the energy criteria described in the next section (2.3). Here, $l = l_p + l_s = 12$ ($l_p = 5$ corresponding to pressure and $l_s = 7$ corresponding to saturation). As mentioned before, each singular vector in the left singular matrix correspond to states that are decreasingly controllable. For compact representation of the spatial patterns of states, we show the weighted singular vectors that is the sum of l_p or l_s singular vectors for pressure or saturation respectively, weighted by the corresponding singular values. For example for pressure, this can be written as:



(a) (b) (c)

Figure 2.2: (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation



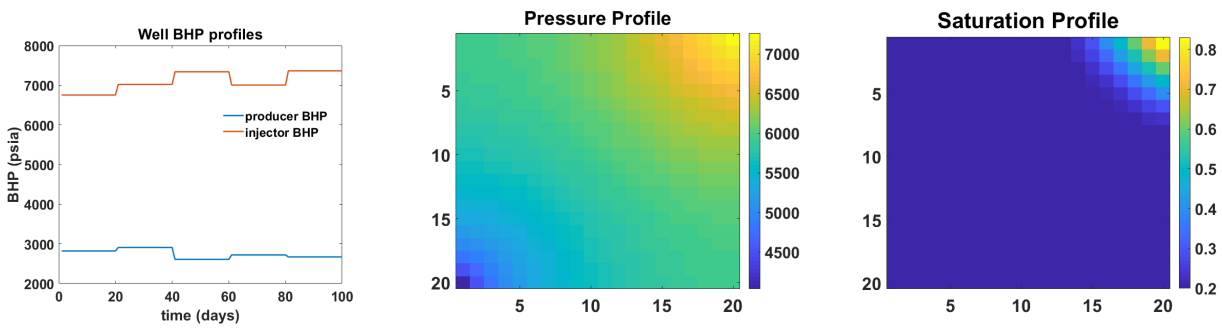
(a) (b)

Figure 2.3: (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for BHP case 1

$$U_{1:l_p} = \sum_{i=1}^{l_p} \frac{\sigma_i}{\sigma_1} \tilde{U}_i \tag{2.13}$$

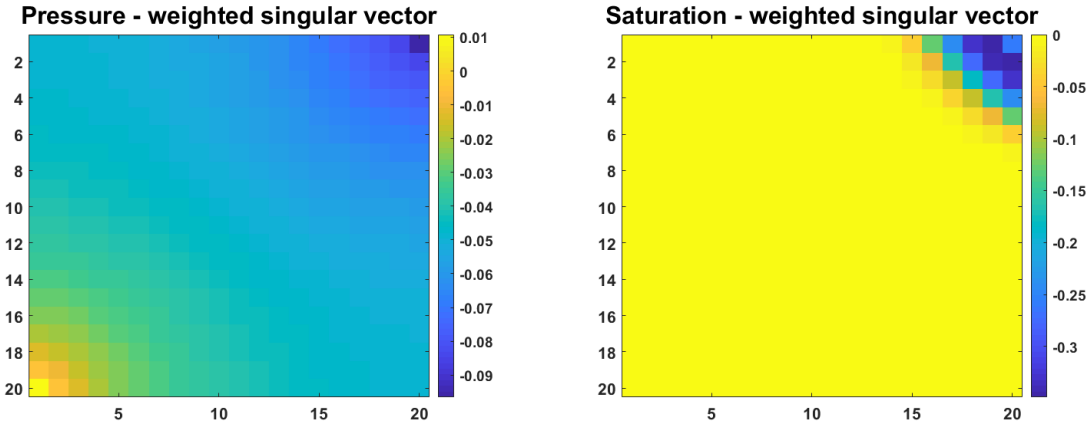
where, σ_i are the singular values in $\mathbf{\Sigma}$ from equation (2.11) arranged in descending order from $\sigma_1, ..., \sigma_{N_d}$.

Figures 2.2b and 2.2c show the pressure and saturation states at the initial time and final time of simulation. Since eigenvalues of pressure and saturation have a big difference in



(a) (b) (c)

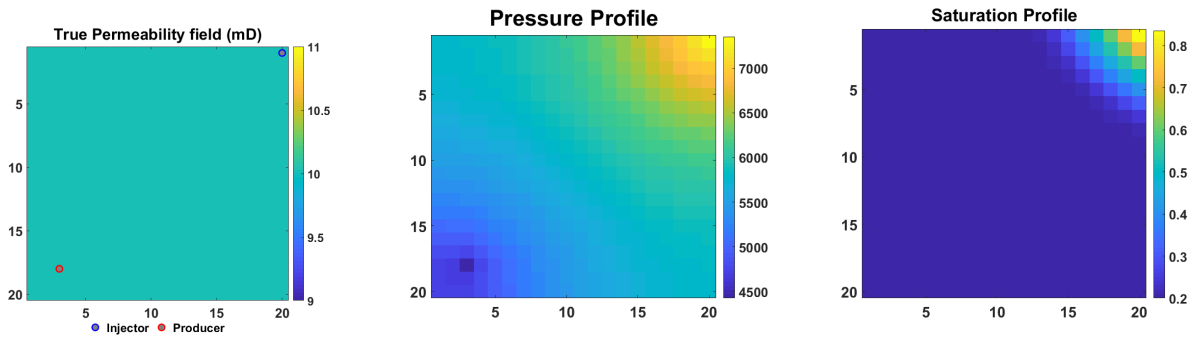
Figure 2.4: (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation



(a) (b)

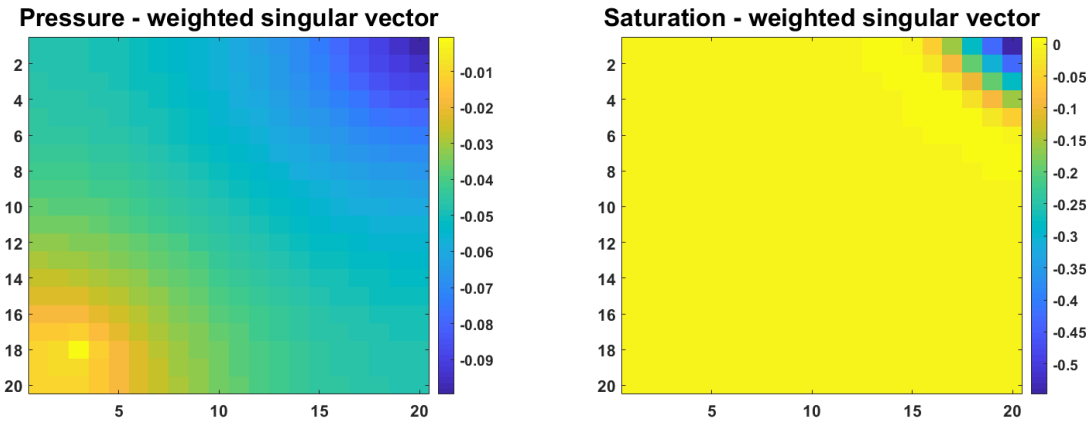
Figure 2.5: (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for BHP case 2

terms of absolute values, we separately analyze the controllability of pressure and saturation. Figure 2.3 shows the weighted singular vectors of the empirical controllability Gramian related to pressure and saturation. This shows the controllability properties of reservoir model that indicates more controllability of the pressure close to the wells and controllability of saturation at the fluid front. For MOR application on changing well controls, what we are interested in is to analyze the controllable subspace for changing BHP profiles. Figure 2.4 shows a new BHP profile that is completely different than the previous case and the cor-



(a) (b) (c)

Figure 2.6: (a) BHP profile of wells with corresponding (b) Pressure and (c) Saturation profiles at the end of simulation



(a) (b)

Figure 2.7: (a) weighted singular vectors of (a) pressure empirical controllability Gramian and (b) saturation empirical controllability Gramian for new well location

responding pressure and saturation solution at the end of simulation. Repeating the same procedure again, Figure 2.5 shows the weighted singular vectors of the empirical Gramians. As can be seen, we do not see a major difference in the controllability properties as compared to the case above, in that it has pressure states controllable near the well locations and saturation at the fluid front. Hence, for the problem of changing well controls, if different possible scenarios of fluid fronts have been captured during the empirical Gramian construction, the controllable subspace is expected to work well for a new set of BHP controls.

Now, we consider the case of changing well locations. So first, we consider the same location of wells as used above, i.e. the injector at location $(20, 1)$ and the producer at $(1, 20)$. We have the weighted singular vector of pressure and saturation controllability matrix as depicted in Figure 2.5. Now we change the producer well location to $(3, 18)$ and repeat the procedure. The BHP profile for new well location is the same as Figure 2.4a. Figure 2.6 shows the new well location and the pressure and saturation profiles at the beginning and end of the simulation and Figure 2.7 shows the new weighted singular vectors of the controllability matrix for this well configuration. It can be seen that the controllable subspace of the previous well configuration cannot satisfy the controllability properties of the new well configuration that requires the lowest pressure at the well location. Thus, unlike MOR for well control, finding the controllable subspace that can satisfy the controllability properties for new well locations is very challenging.

This analysis was shown to provide some general idea about different nature of MOR challenges for the two problems considered. One may also look at the observability properties as a motivation for use of MOR for reservoir simulation applications, as described in [67]. However, observability concept was not deemed very important to justify the complexity of two optimization problems considered, and hence not discussed here. Now we look into the MOR techniques focused on for these problems.

2.3 Projection based Model Order Reduction

In this section, we introduce a projection based MOR technique that projects a high dimensional reservoir simulation equation onto a subspace of much lower dimension (as justified by the controllability properties of the reservoir flow), thus reducing both the number of equations and variables involved. In this work, projection based reduced order models are considered for optimization applications.

2.3.1 Galerkin projection

The main idea of projection based MOR is to generate a dynamical system of much lower dimension r as compared to the full order dimension n as defined by equation (2.6), while still retaining its dominant dynamical properties. One way to achieve this is Galerkin projection that is employed in this research work. We begin with a state space representation of the reservoir simulation equation for two phase flow to explain Galerkin projection. Referring to equation (2.3), at convergence, neglecting gravity, it can be re-written as:

$$\dot{\mathbf{x}}(t,\zeta) = \mathbf{D}(\mathbf{x},\zeta)^{-1}\mathbf{T}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) - \mathbf{D}(\mathbf{x},\zeta)^{-1}\mathbf{Q}(\mathbf{x},t,\zeta) \tag{2.14}$$

which can be written as a state space form:

$$\dot{\mathbf{x}}(t,\zeta) = \mathbf{A}(\mathbf{x},\zeta)\mathbf{x}(t,\zeta) + \mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{2.15}$$

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

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

$$\mathbf{x}(t,\zeta) = \Phi(\zeta)\,\mathbf{x}_r(t,\zeta) + \Phi^\perp(\zeta)\,\tilde{\mathbf{x}}_r(t,\zeta) \tag{2.16}$$

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

$$\mathbf{x}(t,\zeta) \approx \Phi(\zeta)\,\mathbf{x}_r(t,\zeta) \tag{2.17}$$

Thus, $\mathbf{x}_r \in R^r$ represent the components of the state vector $\mathbf{x} \in R^n$ in the subspace $\Phi(\zeta)$. We can now write the equation (2.15) as:

$$\Phi(\zeta)\dot{\mathbf{x}}_r(t,\zeta) = \mathbf{A}(\mathbf{x},\zeta)\Phi(\zeta)\mathbf{x}_r(t,\zeta) + \mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{2.18}$$

The next step is to define the test basis, which for Galerkin projection is the same as the trial basis i.e., $\Phi(\zeta)$. This test basis is left multiplied in equation (2.18) to obtain lower order system equation.

$$\Phi(\zeta)^T\Phi(\zeta)\dot{\mathbf{x}}_r(t,\zeta) = \Phi(\zeta)^T\mathbf{A}(\mathbf{x},\zeta)\Phi(\zeta)\mathbf{x}_r(t,\zeta) + \Phi(\zeta)^T\mathbf{B}(\mathbf{x},\zeta)\mathbf{u}(\mathbf{x},t,\zeta) \tag{2.19}$$

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

$$\begin{aligned}
\epsilon_{ROM}(t,\zeta) &= \mathbf{x}(t,\zeta) - \hat{\mathbf{x}}(t,\zeta) \\
&= \mathbf{x}(t,\zeta) - \Pi_{\Phi(\zeta),\Phi(\zeta)}\mathbf{x}(t,\zeta) + \Pi_{\Phi(\zeta),\Phi(\zeta)}\mathbf{x}(t,\zeta) - \hat{\mathbf{x}}(t,\zeta) \\
&= (I_n - \Pi_{\Phi(\zeta),\Phi(\zeta)})\mathbf{x}(t,\zeta) + \Phi(\zeta)(\Phi(\zeta)^T\mathbf{x}(t,\zeta) - \mathbf{x}_r(t,\zeta)) \\
&= \epsilon_{\Phi(\zeta)^\perp}(t,\zeta) + \epsilon_{\Phi(\zeta)}(t,\zeta)
\end{aligned} \tag{2.20}$$

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

$$\| \epsilon_{ROM}(t,\zeta) \|_2^2 = \| \epsilon_{\Phi(\zeta)^\perp}(t,\zeta) \|_2^2 + \| \epsilon_{\Phi(\zeta)}(t,\zeta) \|_2^2 \tag{2.21}$$

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

will be large and the full ROM error $\epsilon_{ROM}(t, \zeta)$ will be even larger.

Also, it can be shown that Galerkin projection can lead to unstable solutions in the case of non-symmetric system matrix A, leading to unphysical system. Thus, it is very important for an appropriate choice of basis $\Phi(\zeta)$ for stability of the system. One way to address this issue was proposed in [68, 69] using least-squares Petrov-Galerkin (LSPG) method, which computes a different test basis than the trial basis. However, the computational cost associated with LSPG is higher and hence for the current scope of work, we rely on Galerkin projection method.

2.3.2 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition (POD) constructs a basis of dimension r by orthogonal transformation of the data observations such that it represents the data in certain least square optimal sense. The data can thus be represented by a linear combination of the basis vectors which are called the basis functions. POD can be implemented for infinite dimensional or finite dimensional data and does not assume the data source which can be from a linear or non-linear system.

POD is typically employed on the state solutions of the system. As a first step to generate the reduced basis, the full order system is solved for a given parameter ζ, which is called the training step, to generate an ensemble of snapshots which are basically the state solutions gathered at all simulation time steps by solving equations (2.6 and 2.7). More details on this can be found in [35]. Snapshot matrix $\mathbf{X}(\zeta)$ is defined as:

$$\mathbf{X}(\zeta) = [\mathbf{x}^1 \quad \mathbf{x}^2 \quad \mathbf{x}^3 \quad ... \quad \mathbf{x}^{N_t}], \quad \mathbf{X}(\zeta) \in R^{N_d \times N_t} \tag{2.22}$$

where, N_d is the full order dimension of states P_o and S_w, and N_t is the total number of snapshots collected over a time period. From now, we denote P_o and S_w as $\mathbf{p}$ and $\mathbf{S}$ respectively. In order to project a fine scale system to a low dimensional space, the projection

basis $\{\phi_i\}_{i=1}^{r}$ is obtained by solving the minimization problem:

$$\min_{\phi_i} \sum_{j=1}^{N_t} \parallel \mathbf{x}^j - \sum_{i=1}^{r} (\mathbf{x}^{j^T} \phi_i)\phi_i \parallel \tag{2.23}$$

The solution to this minimization problem [9] is given by the SVD of snapshot matrix $\mathbf{X}(\zeta)$, and selecting first r columns of the left projection matrix.

$$\mathbf{X}(\zeta) = \mathbf{U}(\zeta)\mathbf{\Lambda}(\zeta)\mathbf{V}(\zeta)^T \tag{2.24}$$

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

$$E = \frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{N_d} \sigma_i} \tag{2.25}$$

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

$$\mathbf{p}(\zeta) \approx \Phi_p(\zeta)\mathbf{p}_r(\zeta), \qquad \mathbf{S}(\zeta) \approx \Phi_s(\zeta)\mathbf{S}_r(\zeta) \tag{2.26}$$

Thus, for a given parameter, we may write:

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \Phi_p & 0 \\ 0 & \Phi_s \end{bmatrix} \begin{bmatrix} \mathbf{p}_r \\ \mathbf{S}_r \end{bmatrix} \tag{2.27}$$

which takes a short notation:

$$\mathbf{X} = \Phi \mathbf{X}_r \,, \quad r \ll N_d \tag{2.28}$$

Here, Φ is a diagonal matrix with pressure basis and saturation basis as its diagonal elements. Using Galerkin projection, the linear system of equations in equation (2.6) can be written as follows:

$$\Phi(\zeta)^T \mathbf{J}(\zeta)^{n+1} \Phi(\zeta)\, \boldsymbol{\delta}_r(\zeta)^{n+1} = -\Phi(\zeta)^T \mathbf{R}(\zeta)^{n+1} \tag{2.29}$$

which leads to a system of reduced order equations:

$$\mathbf{J}_r(\zeta)^{n+1} \boldsymbol{\delta}_r(\zeta)^{n+1} = -\mathbf{R}_r(\zeta)^{n+1} \tag{2.30}$$

and after solving the equation (2.30), the reduced state space variables are updated as:

$$\mathbf{x}_r(\zeta)^n + \boldsymbol{\delta}_r(\zeta)^{n+1} = \mathbf{x}_r(\zeta)^{n+1} \tag{2.31}$$

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

2.3.3 Parametric Model Order Reduction concept

Thus, as we saw above for POD, the main aim is to replace the computationally expensive solver in equation 2.6 to a reduced form of the Jacobian and Residual as shown in

equation (2.30), where, $r \ll N_d$. To this end, we aim to construct reduced order models or reduced order basis that are an accurate approximation of the fine-scale solution in the *entire* parameter domain i.e., for any ζ in equation (2.3). For clarity again, ROB is defined as the basis of lower dimension obtained by POD and the ROBs are used to construct ROMs which are lower dimensional system matrices the accuracy of which is reflected as the predicted output of the system. Thus, ROM error refers to the error in output predicted by the reduced system.

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

$$\| \mathbf{X}(t,\zeta) - \Phi(\zeta)\mathbf{X}_r(t,\zeta) \| < \eta \tag{2.32}$$

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

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

3. MODEL ORDER REDUCTION FOR WELL CONTROL OPTIMIZATION*

In this section, as we develop the MOR strategy for well control optimization, we begin by with a brief introduction about this work. The next section focuses on formulating the well control optimization problem with an introduction to polynomial controls and motivation behind using them. We then develop various control parameterization techniques later that will be evaluated for optimization and MOR performance. In the next section, we briefly discuss about various optimization strategies that will be analyzed for use with MOR technique. And finally, we propose the MOR methodology with the improved training strategy and evaluate its performance with a case study.

Waterflooding control optimization is a PDE-constrained optimization problem, where the PDE corresponds to the reservoir simulation equations as represented in equations (2.1 and 2.2). The aim is to seek optimal controls of the wells such that it maximizes certain objective functions like sweep efficiency, net present value (NPV), and cumulative oil production or minimizing water cut among others.

Reservoir flooding optimal control problems pose computational difficulties at both simulation and optimization levels. Extensive research has been attempted to alleviate these hurdles. At the simulation level, the attempt is usually achieved by constructing a surrogate model that is computationally cheap to evaluate as described in the sections above accomplished here using MOR. At the optimization level, the continuous nature of the control, which lie in an infinite dimensional space, call for a control parameterization procedure to cast the problem into a finite dimension problem that is cheaper to compute. This work combines these two methods to accelerate the process of reservoir flooding production optimization. In the reservoir modeling level, we apply Model Order Reduction (MOR) by utilizing the Proper Orthogonal Decomposition with Discrete Empirical Interpolation

*Reprinted with permission from "SPE-182652-MS Model Order Reduction and Control Polynomial Approximation for Well-Control Production Optimization" by N. Sorek, H. Zalavadia, E. Gildin, 2017, SPE Reservoir Simulation Conference, Copyright 2017 by SPE

Method (POD-DEIM). In the upper optimization level, we apply dimensionality reduction by parametrization of the infinite dimensional control using Polynomial Approximation (PA) and Piecewise Polynomial Approximation (PPA) as well as the traditional Piece-Wise Constant (PWC) for comparison.

3.1 Well Control Optimization Formulation

3.1.1 Optimal Control of Reservoir Flooding

It has been discussed widely in the literature that reservoir flooding optimization can be treated as an optimal control problem [70, 71, 72, 73]. In this approach, the well settings are the control variables (also called decision or optimization variables), which can be represented as either valve positions, Bottom Hole Pressure (BHP) of wells, well flow rates (injection or production) and concentrations. In general, there are two distinct classes of algorithms to solve an optimal control problem, namely indirect and direct methods. In the former, one analyzes the optimality conditions prior to applying numerical discretization. Such indirect methods are dynamic programing which solves the Hamilton-Jacobi-Bellman (HJB) equation (suitable to a few state variables problem) and the Pontryagin Maximum Principle (PMP) method [74]. In the direct method, on the other hand, the optimal control problem is approximated by a finite dimensional non-linear programming (NLP) problem. Then, different standard optimization algorithms (stochastic or deterministic, global or local) can solve the approximated problem. Direct methods include single shooting, multiple shooting and orthogonal collection approaches. The last two methods introduce many decision variables and constraints which might lead to prohibitive computational efforts in reservoir simulation problems. However, implementation of multiple shooting for reservoir flooding problem was recently introduced in [71].

In this current work we use a direct single shooting approach to solve the optimal control problem. This method parameterizes the control trajectories along time and the parameterization coefficients become the decision variables in finite-dimensional optimization problem.

Theoretically, prior to any parameterization procedure, the control variables are manipulated continuously in time, and thus the optimal solution trajectory lies in an infinite dimensional space. Practically, however, due to time discretization in a reservoir simulator, the control trajectory is projected into a finite dimension of piecewise-constant intervals, and usually the control values at each interval become the finite set of decision variables. This piecewise-constant projection is a finite dimensional approximation of the original infinite dimensional solution. Next, we investigate different parameterization approaches.

3.1.2 Optimization Dimensionality Reduction by Polynomial Approximation

Indeed, most of the reservoir flooding optimization works presented in the literature account for parameterization and dimensionality reduction via a simple piece-wise constant approximation. That is, the time discretization of the wells schedule (namely, the points in time where a change in the control is introduced into the simulator) is identical to the optimization parameterization. However, in recent work [75, 76] two different parameterization procedures were presented, namely Polynomial Control Method (PCM) using monomial polynomial approximation (PA) and Interpolation Control Method (ICM) using cubic Spline piece-wise polynomial approximations (PPA). In these methods, the control trajectory of each well is polynomial or piecewise polynomial, such that the input to each control-step in the simulator is regulated by an upper level polynomial functions. The decision variables in this case are the polynomial coefficients (in the PCM case) or the interpolation points (in the case of ICM). Therefore, smooth solutions were enabled, and the algorithmic search turned relatively efficient in the new dimension-reduced polynomial space. We note here also the work in [70] which presented both polynomial and trigonometric parametrization for reservoir flooding problems. Polynomial approximation is a well-known technique in scientific computing to approximate more complicated functions [77]. The procedure involves a weighted summation of finite number of orthogonal polynomials.

In optimization, infinite dimensional problem can be transformed to a finite problem by polynomial approximation of a continuous decision variable, which is an unknown input

function. For example, the work in [78] presents a case of linear objective and constraints, where a sequence of optimal solutions of finite dimensional linear subspaces converge to the optimal value of the original infinite-dimensional problem. Another interesting example is the work presented first in [79] for deterministic global optimal control, and then in [74] for a much more general framework for optimization in Hilbert space. The authors of these works introduced a new algorithm named Branch and Lift, in which the problem is projected into a polynomial subspace using orthogonal Legendre polynomials, starting with branching in a low dimensional subspace, and gradually lifting the subspace dimension till obtaining a convergence to a suboptimal global solution. The aforementioned previous attempts of polynomial approximation of reservoir flooding control used monomial basis functions (i.e., the set $\{1, x, x^2, ...\}$). However, this natural form of the polynomial might be inefficient during the optimization since all the monomials look very similar on $[0, 1]$, such that large changes in the polynomial coefficients yield only small changes in the control [80]. Therefore, in this work we consider orthogonal polynomial basis functions, which provide more variability and efficient search during the optimization. Some examples from the optimal control literature which used orthogonal polynomials can be found in [81, 79].

3.1.3 Infinite Dimensional Problem Statement

In this work, the objective function that we try to maximize is Net Present Value (NPV) that dictates the economic value of the project and in order to achieve this, the control variable that we seek to optimize is the BHPs of producer and injector wells. In order to formulate the optimal control problem, we consider the reservoir simulation equation in the form as shown in equation (2.14). Note that in this equation, the control input is assigned through the source-sink term q_l found in the $\mathbf{Q}$ vector, which is defined in equation (2.4). The control input can either be the value assigned as q_l (known as rate control) or as p_{wf} (known as pressure control).

To this end, as was discussed in [72], the controls enter linearly into the flow model, and

we can rewrite the flow equation as:

$$\dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\mathbf{u}(t) \tag{3.1}$$

where $\mathbf{u}(t) = [u_1(t), \ldots, u_j(t), \ldots, u_{N_w}(t)]$ is a column vector contains continuous control functions of each well j. Here, N_w is the total number of wells. Originally, prior to any discretization, each u_j function maps a continuous time input to a continuous control trajectory. Equation (3.1) represents the governing partial differential reservoir flow equation, which is now formulated in a well-known optimal control form, that describes the reservoir system dynamics in a state space form.

3.1.4 Objective function

We consider the NPV function as the objective function, which accounts for revenue associated with produced oil and for the cost of handling produced and injected water (which is incurred as a result of pumping and separation requirements),

$$J(\mathbf{u}) = NPV = \int_0^{T_f} \left(\sum_{j=1}^{N_p} r_o q_o^j(u(t)) - \sum_{j=1}^{N_p} c_{wp} q_{wp}^j(u(t)) - \sum_{j=1}^{N_i} c_{wi} q_{wi}^j(u(t)) \right) \frac{1}{(1+ir)^{t/t_{ref}}} dt \tag{3.2}$$

where q_o^j, q_{wp}^j, q_{wi}^j are the flow rates of the oil, water produced and water injected for well j, respectively. The revenue from a unit of oil produced, the cost of a unit of water produced and a unit of water injected are represented by r_o , c_{wp} and c_{wi}, respectively. t and T_f are continuous and terminal times, respectively. ir is the interest rate associated with a time reference t_{ref}. N_i and N_p are the total number of injection and production wells, respectively. Again, as can be observed from equation 3.2 and as was discussed in [72], the performance measure (the objective function) is linear in the control and can be rewritten in a typical optimal control form:

$$J(u) = \chi\big(\mathbf{x}(T_f), T_f\big) + \int_0^{T_f} L\big(\mathbf{x}(t), \mathbf{u}(t), t\big) dt \tag{3.3}$$

3.1.5 The Infinite Dimension Optimization Problem

Now, we can introduce the infinite dimension optimal control Bolza problem, as follows:

$$\begin{aligned}
\max_{u(t)} \quad & \chi\big(\mathbf{x}(T_f), T_f\big) + \int_0^{T_f} L\big(\mathbf{x}(t), \mathbf{u}(t), t\big) dt \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\mathbf{u}(t) \\
& \mathbf{u}(t) = [u_1(t), \ldots, u_j(t), \ldots, u_{N_w}(t)]^T \\
& u_j(t) \in \mathscr{U}, \forall t \in [0, T_f] \\
& \mathscr{U} = \{h(t) : u_{lb} \leq h(t) \leq u_{ub}\} \\
& x(0) = x_0
\end{aligned} \tag{3.4}$$

where the initial conditions are specified as $x(0) = x_0$, and the original bounded infinite dimension control space is defined by u.

3.2 Parameterization Strategies

Practically, the problem in equation (3.4) is parameterized (and hence approximated) in two different levels. In the reservoir simulation level, the constraint given in the state equation $\dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\mathbf{u}(t)$ is being discretized both in time and space in order to solve numerically the PDE (as was mentioned in Section 2). Note that different time discretization is often applied to the state variables and to the control variables. We shall note that here we define N_{sim} as the number of simulation time steps. In this case, a surjective function maps each simulation time step index (associated with the solution of the state variables) to a control variable index [71]. Next, in the optimization process, the continuous decision variables $u_j(t)$ is being parametrized and projected to a finite dimension space such that the problem can be tackled with any existing optimization algorithm.

We discuss here three different parameterization approach: piecewise constant (PWC), polynomial approximation (PCM) and cubic spline (ICM). The first and the third, both belong to the control vector parameterization (CVP) approach where the time horizon is

divided into intervals in which the control variable is approximated by a constant value or with some higher order approximations [82]. On the other hand, the PCM describes the control at all times with one continuous polynomial function constructed by weighted summation of orthogonal polynomial basis functions.

3.2.1 Piecewise Constant Parametrization

Traditionally, in reservoir flooding optimization the decision variables values enter directly as input to the simulator using a surjective function $h : R^{N_{sim}} \rightarrow R^{N_u}$, where each simulation time step of the total N_{sim} simulation time steps is associated with one of the N_u control time intervals. In each interval the control is kept constant, which render a piecewise constant approximation of the original continuous control (decision) variable.

To formalize this concept, for each well with index $j = \{1, \ldots, Nw\}$, we can obtain a piecewise constant parameterization over N_u intervals of equal duration $h = T_f/N_u$. By applying a linear combination of basis functions $\psi_1(t), \psi_2(t), \ldots, \psi_{N_u}(t)$, where:

$$\forall i \in \{1, 2, ..., N_u\}, \;\; \forall j \in \{1, 2, ..., N_w\}, \;\; \psi_{i,j}(t) \begin{cases} 1 \;\; , & \text{if } \; (i-1)h \leq t \leq ih \\ 0 \;\; , & \text{otherwise} \end{cases} \tag{3.5}$$

such that the control of each well j is approximated as follows:

$$\forall t \in R^{N_{sim}}, \;\; \forall j \in \{1, 2, ..., N_w\}, \;\; u_j(t) \approx \tilde{u}_j(t) = \sum_{i=1}^{N_u} a_{i,j} \psi_{i,j}(t) \tag{3.6}$$

Note that the equality $u_j(t) \approx \tilde{u}_j(t)$ holds when $N_u \rightarrow \infty$. Also, $\tilde{\mathbf{u}}(t) = [\tilde{u}_1(t), \ldots, \tilde{u}_j(t), \ldots, \tilde{u}_{N_w}(t)]^T$, as the vector contains approximations for all the wells at every time step and by inserting equation (3.6) into equation (3.4), we project the infinite dimensional optimization problem into a finite dimension (of a degree $N_u N_w$) optimization problem. Taking into

account the simulation time discretization we get:

$$\begin{aligned}
\max_{\tilde{u}(t)} \quad & \chi\big(\mathbf{x}(T_f), T_f\big) + \sum_{k=1}^{N_{sim}} L_k\big(\mathbf{x}(t), \tilde{\mathbf{u}}(t), t\big) \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\tilde{\mathbf{u}}(t) \\
& \tilde{u}_j(t) = \sum_{i=1}^{N_u} a_{i,j}\psi_{i,j}(t), \forall j \in \{1, ..., N_w\} \\
& u_{lb} \leq a_{i,j} \leq u_{ub} \\
& x(0) = x_0
\end{aligned} \tag{3.7}$$

Note that the integral from equation (3.4) was replaced in equation (3.7) by a summation over N_{sim} simulation time steps, such that the flow rates are now associated with discrete values with an index k. Also note that rather than a continuous functions $u_j(t)$, the decision variables are now the control values a_i at each well at each one of the N_u control time steps.

3.2.2 Polynomial Approximation using Chebyshev Polynomials

Let us define the following linear combination:

$$\forall t \in R^{N_{sim}}, \;\; \forall j \in \{1, 2, ..., N_w\}, \;\; u_j(t) \approx \tilde{p}_j(t) = \sum_{i=1}^{N_p} c_{i,j}\Theta_{i,j}(t) \tag{3.8}$$

where $\Theta_{i,j}(t)$ are polynomial basis functions, $c_{i,j}$ are the variables to be optimized, and $\tilde{p}_j(t)$ is the resulting polynomial approximation for the control of well j. Again, note that equality $u_j(t) = \tilde{p}(t)$ holds when $N_p \to \infty$.

The main advantage of this approach is that the optimization dimension is decoupled from the reservoir simulation control discretization, and a fine control approximation can be optimized only with a few optimization (decision) variables. This cardinality reduction of the decision variables set gets important, as optimization algorithms might not converge in a reasonable time when the number of variables grow unbounded [74].

We note $\tilde{\mathbf{p}}(t) = [\tilde{p}_1(t), \ldots, \tilde{p}_j(t), \ldots, \tilde{p}_{N_w}(t)]^T$, as the vector contains all approximations

for all wells in all time steps and by inserting equation (3.8) into equation (3.4), we project the infinite dimensional optimization problem into a finite dimension (of a degree $N_p N_w$) optimization problem. With this parameterization the optimal control problem can be defined similarly to equation (3.7), as follows:

$$\begin{aligned}
\max_{\tilde{u}(t)} \quad & \chi\big(\mathbf{x}(T_f), T_f\big) + \sum_{k=1}^{N_{sim}} L_k\big(\mathbf{x}(t), \tilde{\mathbf{p}}(t), t\big) \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\tilde{\mathbf{p}}(t) \\
& p_j = \sum_{i=1}^{N_p} c_{i,j}\Theta_{i,j}(t), \forall j \in \{1, ..., N_w\} \\
& c_{lb} \leq c_{i,j} \leq c_{ub} \\
& x(0) = x_0
\end{aligned} \tag{3.9}$$

where N_p is the finite number of polynomial basis functions.

As we discussed before, previous works dealt with polynomial approximation for reservoir flooding [70, 75] using monomial basis functions $\Theta_i = \tilde{t}^{i-1}$, where $\tilde{t}$ is a scaled time. This natural form of the polynomial might be inefficient and therefore here we consider orthogonal polynomial basis functions. Some examples from the optimal control literature are Lagrange Polynomials [81], Legendre polynomials [79] and Chebyshev polynomials [83]. The latter is also our choice in this work.

The Chebyshev polynomials of the first kind takes the following form:

$$T_n(x) = \cos\big(n \arccos(x)\big) \tag{3.10}$$

which take the following recursion form:

$$T_0(x) = 1, T_1(x) = x, T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \tag{3.11}$$

These polynomials are orthogonal on the interval $-1 \leq x \leq 1$ with respect to the following

weight function:

$$w(x) = \frac{1}{\sqrt{1-x^2}} \tag{3.12}$$

such that:

$$\forall i, j \in (0, ..., N_p), \quad \frac{1}{\sigma_n} \int_{-1}^{1} T_n(x) T_m(x) w(x) dx = \delta_{n,m} = \begin{cases} 1 \ , & \text{if} \quad n \neq m \\ 0 \ , & \text{otherwise} \end{cases} \tag{3.13}$$

where σ_n is a scaling factor.

Our choice of Chebyshev polynomials stemmed from an inspection of their image on the support $[-1, 1]$. By scaling both time and controls coordinates to the box $[-1, 1] \times [-1, 1]$, the Chebyshev polynomials of the first kind are promising candidates as their image is bounded to this box on the support $[-1, 1]$, as shown in Figure 3.1:

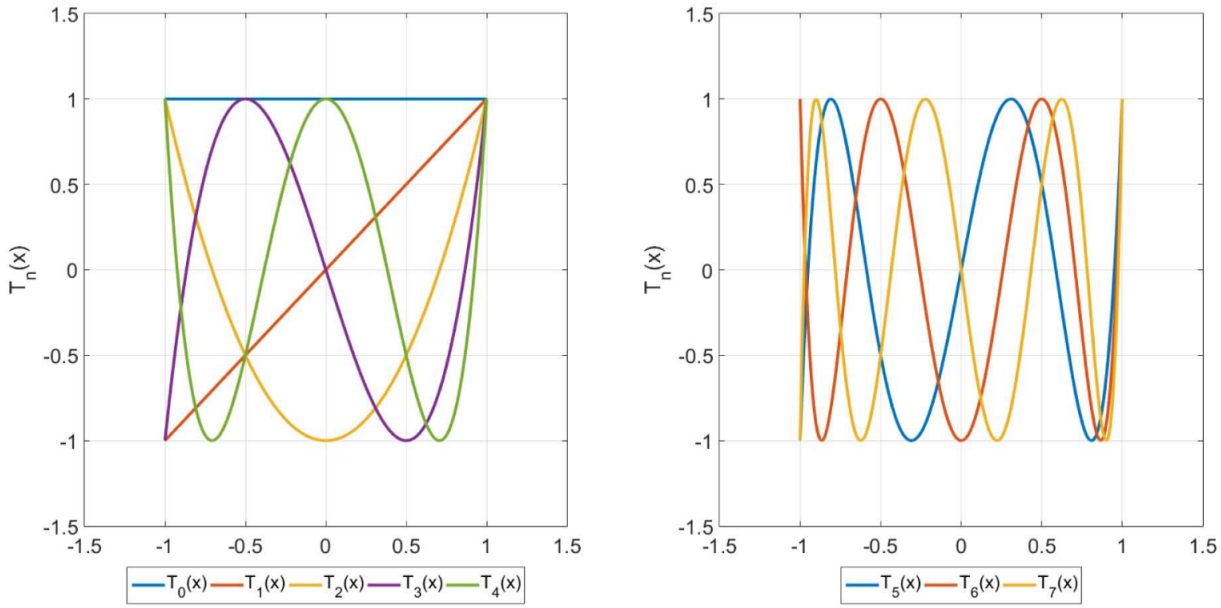


Figure 3.1: Chebyshev polynomials of the first kind. Left: degrees zero up to four. Right: degrees five, six and seven

Note that for monomial basis functions $\{1, x, x^2, \ldots, x^{N_p}\}$, the bounds c_{lb} and c_{ub} are

simply the bounds on the polynomial coefficient. In more general, for a non-monomial basis, the bounds are imposed on the coefficient in front of the polynomial basis functions. The bounds values can be set heuristically (as in [75, 70]) or deterministically (as in [79]). Before setting those bounds it is important to consider that optimal controls of a reservoir flooding problem might result in a combination of bang-bang and singular arcs controls [72]. That is, control trajectories might be found either on the bounds or continue smoothly between bounds. Thus, we set the bounds for the decision variables as follow $-1 \leq c_{i,j} \leq 1$ such that we allow at any arbitrary time interval to obtain a control trajectory on the bounds. Figure 3.2 demonstrates this concept with three polynomials resulted from a linear combination of Chebyshev polynomials. Note that we allow the trajectories to get slightly out of bounds to enable bang-singular control (the control is set to the nearest boundary to the polynomial curve).

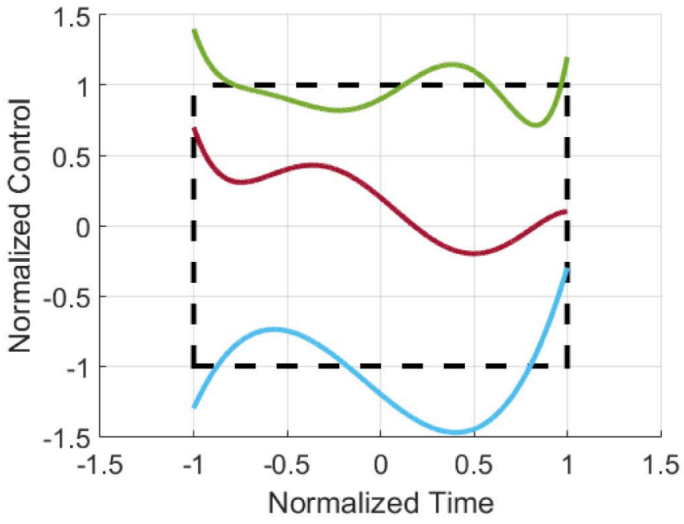


Figure 3.2: Examples of polynomial trajectories obtained by weighted sum of Chebyshev polynomials. Intervals outside of the box are treated as (nearest) boundary control.

3.2.3 Piecewise Polynomial Approximation using Cubic Spline Interpolation

In order to avoid the question which bounds to impose on the variables $c_{i,j}$, one can obtain a polynomial-control representation using piecewise polynomial interpolation between points sampled from the original u space. Some examples of interpolated curves used in the optimal control literature are B-spline functions [84] and Bézier curves [85]. The advantages of piecewise polynomial over a polynomial interpolation in terms of numerical stability is well-known [86, 77]. Our choice here is cubic spline interpolation with a not-a-knot condition (for details see [86]). Cubic spline interpolation guarantees continuity up to the second derivative and thus impose a smooth control trajectory that might be beneficial from a production engineering point of view [87].

Let us denote the vector $z_j \in R^{N_z}$ as the vector contains N_z interpolation point for each well j. Then, a cubic piecewise polynomial approximation can be defined as follows:

$$\begin{aligned} &\forall i \in \{1, 2, ..., N_{z-1}\}, \ \ \forall j \in \{1, 2, ..., N_w\} \\ &\Omega_{i,j}(\mathbf{z}_j, t) \begin{cases} \gamma_{0,i,j} + \gamma_{1,i,j} t + \gamma_{2,i,j} t^2 + \gamma_{3,i,j} t^3 \ , & \text{if } (i-1)h \leq t \leq ih \\ 0 \ , & \text{otherwise} \end{cases} \end{aligned} \tag{3.14}$$

where a piecewise cubic spline interpolation between the values in z_j determines the coefficient γ_0 , γ_1, γ_2 and γ_3 for each $\Omega_{i,j}(zj, t)$. Here, the control is approximated as follows:

$$\forall t \in R^{N_{sim}}, \ \ \forall j \in \{1, 2, ..., N_w\}, \ \ u_j(t) \approx \tilde{s}_j(t) = \sum_{i=1}^{N_z - 1} \Omega_{i,j}(\mathbf{z}_j, t) \tag{3.15}$$

We note $\tilde{\mathbf{s}}(\mathbf{z}, t) = [\tilde{s}_1(\mathbf{z}_1, t), \ldots, \tilde{s}_j(\mathbf{z}_j, t), \ldots, \ \tilde{s}_{N_w}(\mathbf{z}_{N_w}, t)]^T$ as a column vector contains approximations for all wells at every time step, where $\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_j, ..., \mathbf{z}_{N_w}]^T$ is the concatenated interpolation vector. By inserting equation 3.15 into equation 3.4, we project the infinite dimensional optimization problem into a finite dimension (of a degree $N_z N_w$) optimization problem. Then, the finite dimensional optimization problem takes the following

form:

$$
\begin{aligned}
\max_{\tilde{u}(t)} \quad & \chi\big(\mathbf{x}(T_f), T_f\big) + \sum_{k=1}^{N_{sim}} L_k\big(\mathbf{x}(t), \tilde{\mathbf{s}}(\mathbf{z}, t), t\big) \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f_1(\mathbf{x}(t)) + f_2(\mathbf{x}(t))\tilde{\mathbf{s}}(\mathbf{z}, t) \\
& \tilde{\mathbf{s}}_j(\mathbf{z}_j, t) = \sum_{i=1}^{N_z - 1} \Omega_{i,j}(\mathbf{z}_j, t), \forall j \in \{1, ..., N_w\} \\
& \mathbf{u}_{lb} \leq \mathbf{z}_{i,j} \leq \mathbf{u}_{ub} \\
& x(0) = x_0
\end{aligned}
\tag{3.16}
$$

Note that the decision variables in this case are the interpolation points inside z_j vectors.

These are the control parameterization techniques we use for well control optimization and evaluate their performance with model order reduction technique developed in the later section (3.4).

3.3 Motivation for Polynomial Controls

In this section, we demonstrate why polynomial control strategy is the choice of control parameterization using a case study for varying number of control variables.

We show a test case for three different parametrization methods, namely PWC, Spline and Chebyshev polynomials, each for ten different levels of parametrization and for two types of algorithms (gradient-based and gradient-free). Thus, in total, we performed 60 optimization runs for this section. Our goal here is to compare the different parametrization schemes for each of the optimization strategies used here namely gradient-based and gradient-free methods. For each parameterization, we tested the range from one decision variable per well (constant value) up to ten variables per well. We reiterate that this number translate to the number of control intervals for PWC, to the number of interpolation points in Spline and to the number orthogonal polynomial basis in the Chebyshev method.

As for the gradient-based algorithm, we used the Interior Point Optimization (IPOPT) method with limited memory BFGS (L-BFGS) inverse Hessian quasi-Newton approximation. More details on this method can be found in [88]. Average values were used as initial guess

for all gradient-based optimization runs. For the gradient- free method, we used Particle Swarm Optimization with dynamic neighborhood and dynamic inertia [89], with a swarm of 30 particles.

3.3.1 Case Study

Here, since we tested many cases, we used a fast and simple incompressible simulator as discussed in Section 2. In the next section, where we couple different parametrization methods with reduced order modeling, we use a more accurate black oil simulator. Figure 3.3 shows the model used in this section, with a five-spot pattern (four producing wells and one injector) in a 51 by 51 grid. The grid cell dimensions are $20 \times 20 \times 20\, ft^3$. There are three permeability sub-regions in three distinct geological facies: facies 1 with the original low-permeability (about $100\, mD$) geological environment, facies 2 with east-west moderate permeability (about $680\, mD$) channels formed by ancient rivers and facies 3 with newer high-permeability (about $1370\, mD$) channels whose flow eroded the environment from north to south. The porosity, the initial and residual water saturation are all assumed equal to 0.2. The relative permeability curves are shown in Figure 3.3.

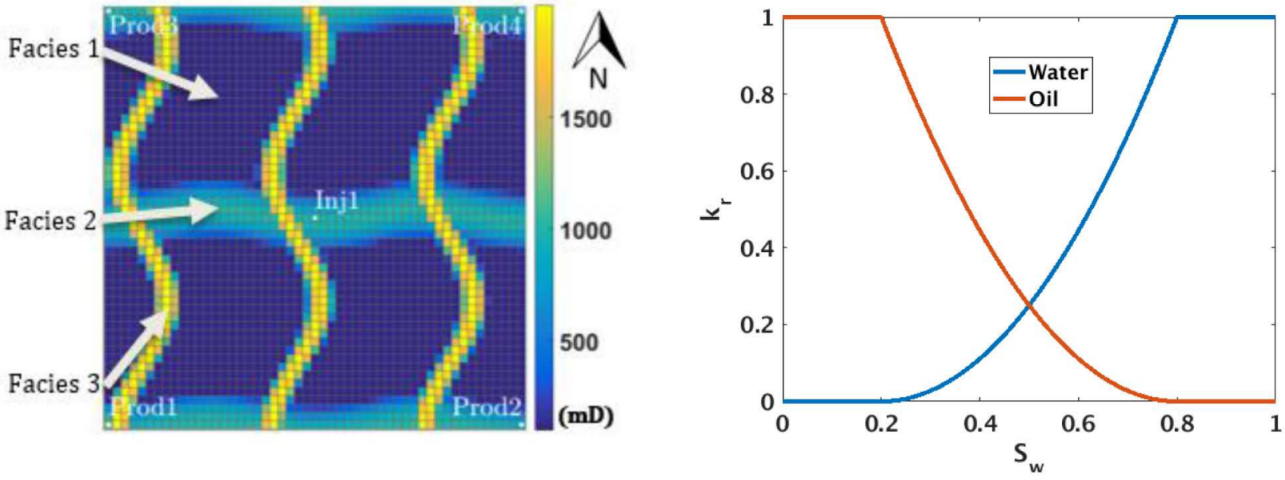


Figure 3.3: Synthetic two-dimensional reservoir, with its relative permeability curves

We ran the swarm simulations and generated the forward finite difference gradients in parallel on a High Performance Computing (HPC) machine (Texas A&M High Performance Research Computing facility using Ada IBM NeXtScale Cluster operated with Linux, CentOS 6.6, using 31 nodes with 64 GB, DDR3 1866 MHz). We assumed a ratio of ten to one between oil revenue and cost of water produced and injected (100:10:10 $/STB, respectively), with an interest rate of 10%. Note that though these prices might not reflect the actual market prices for a given point in time, same optimal control solution would be obtained for different prices scenarios as long as the oil-water price ratios are kept the same, as can be inferred from equation (3.2). The only inequality constraints we considered are hard bounds. Output inequality constraints are out of the scope of this work. For all producers, we set a lower bound of $500\,psia$ and an upper bound of $2200\,psia$. For the injector, the upper and lower bounds are 2100 and $3000\,psia$, respectively.

3.3.2 Parametrization Sensitivity Computational Results

Figure 3.4 shows the relative comparison results and Figure 3.5 shows the number of iterations to convergence for all ten levels of parameterization, and for both gradient-free using PSO and gradient-based using IPOPT methods. All the values are given in percentage relative to the best solution ($17.73 million) obtained by Chebyshev parameterization with 10 polynomials basis for each well.

The convergence criterion for the PSO was 10 stall iterations without a relative improvement above a threshold of 10^{-8}. All the IPOPT cases stopped after the relative norm of the step (change in the decision variables) reached below a threshold of 10^{-8}. We note that at the time of convergence with respect to this criterion, all gradient base cases did not satisfy the first optimality condition. Thus, the algorithm did not guarantee the existence of KKT conditions, implying the local optimality was not necessarily obtained.

From both Figure 3.4 and Figure 3.5, it is hard to derive a general conclusion which parameterization methods yield the best results. Note that parameterization level 1 trans-

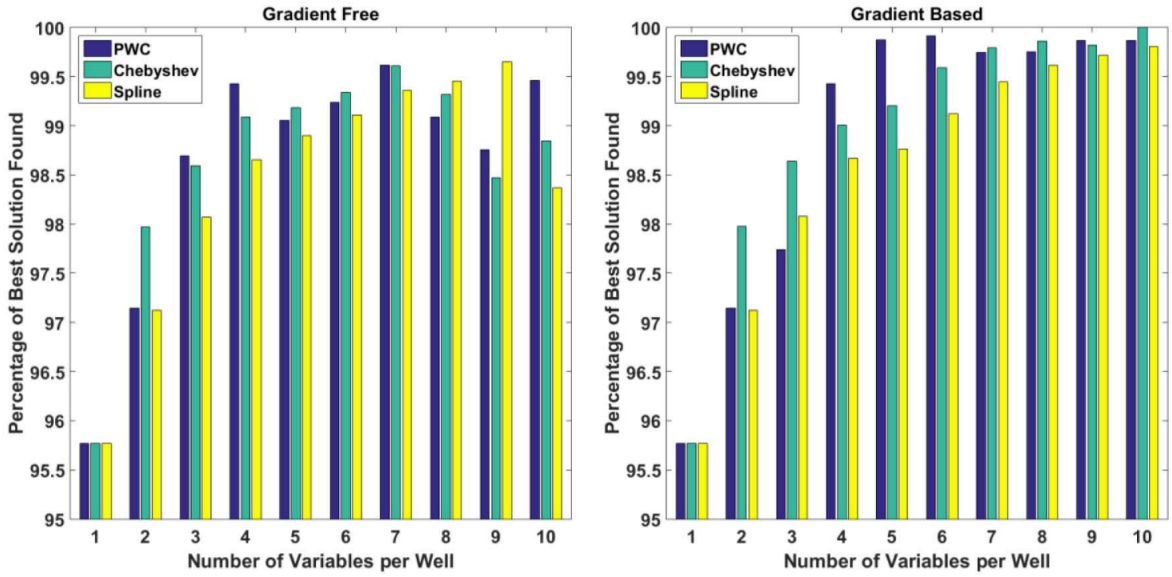


Figure 3.4: (Left) Gradient free (PSO) and (Right) gradient based (IPOPT) results for different numbers of basis functions, normalized to the best solution found ($17.73 million)

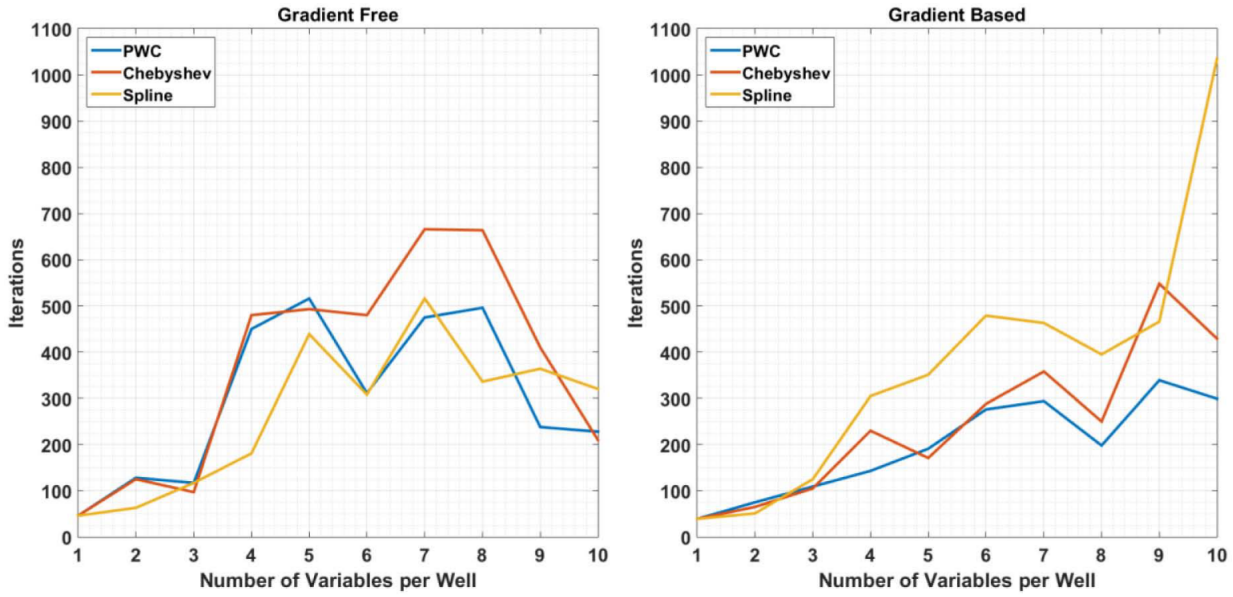


Figure 3.5: (Left) Gradient free (PSO) and (Right) gradient based (IPOPT) number of iterations

lates into optimization of one constant value, and thus all methods are equivalent in this level. In the gradient-free approach (Figure 3.4)(Left), PWC achieved the highest objective

function at four levels (3, 4, 7 and 10), Chebyshev polynomials at three levels (2, 5 and 6) and Spline at two levels (8 and 9) which had the highest value in the gradient free group ($17.67 million). Note that due to the stochastic nature of PSO, testing more realizations per each level of parameterization might lead to different observations.

In the gradient-based approach (Figure 3.4(Right)), PWC achieved the highest objective function values at four levels (4, 5, 6 and 9), and Chebyshev polynomials at five levels (2, 3, 7, 8 and 10, the last with the highest value in all 60 runs). Note that Spline method underperformed at all levels in the gradient-based approach. In terms of computational cost, note from Figure 3.5(Left) that, in the gradient-free approach, Spline achieves the fastest convergence in most of the cases, and Chebyshev is the slowest in most of the cases. From Figure 3.5(Right), we can see that, in the gradient-based approach PWC converge the fastest in most of the cases, while Spline is the slowest in most of the cases.

Though local optimality was not proven, we can see from Figure 3.4 how the gradient-based method consistently outperforms the gradient-free method. We performed tests (not shown here) with more tight gradient-free convergence criteria, but they yielded only slight solution improvements with much higher computational cost (number of iterations). We note here the work of [90] which had a different observation, that the gradient free methods outperform SQP (Sequential Quadratic Programming) gradient based algorithm, where the gradients were acquired via finite difference derivatives (same as this current work). If the adjoint implementation is available, the gradient based methods become much more efficient in terms of the number of simulation runs.

From Figure 3.4(Right), the gradient-based method consistently produced better objective values upon adding more variables for most of the cases. PWC is an exception where, in some cases (in both algorithms) adding more variables causes a decrease in the obtained solution. Thus, it seems that PWC is more prone to be trapped in the local solutions. From the previous observation of improving solutions as the number of basis increases, we reinforce the claim that as the number of basis approaches infinity, the approximation to the infinite

solution turns more accurate.

Figure 3.6 shows the optimal control trajectories obtained by each parametrization method for a level of 10 variables per well. The control changes every 10 days in these cases. The results are given for both gradient-free (PSO) and gradient-based (IPOPT) methods. The control trajectories shown for IPOPT with Chebyshev polynomial approximation is the solution which produced the highest NPV.

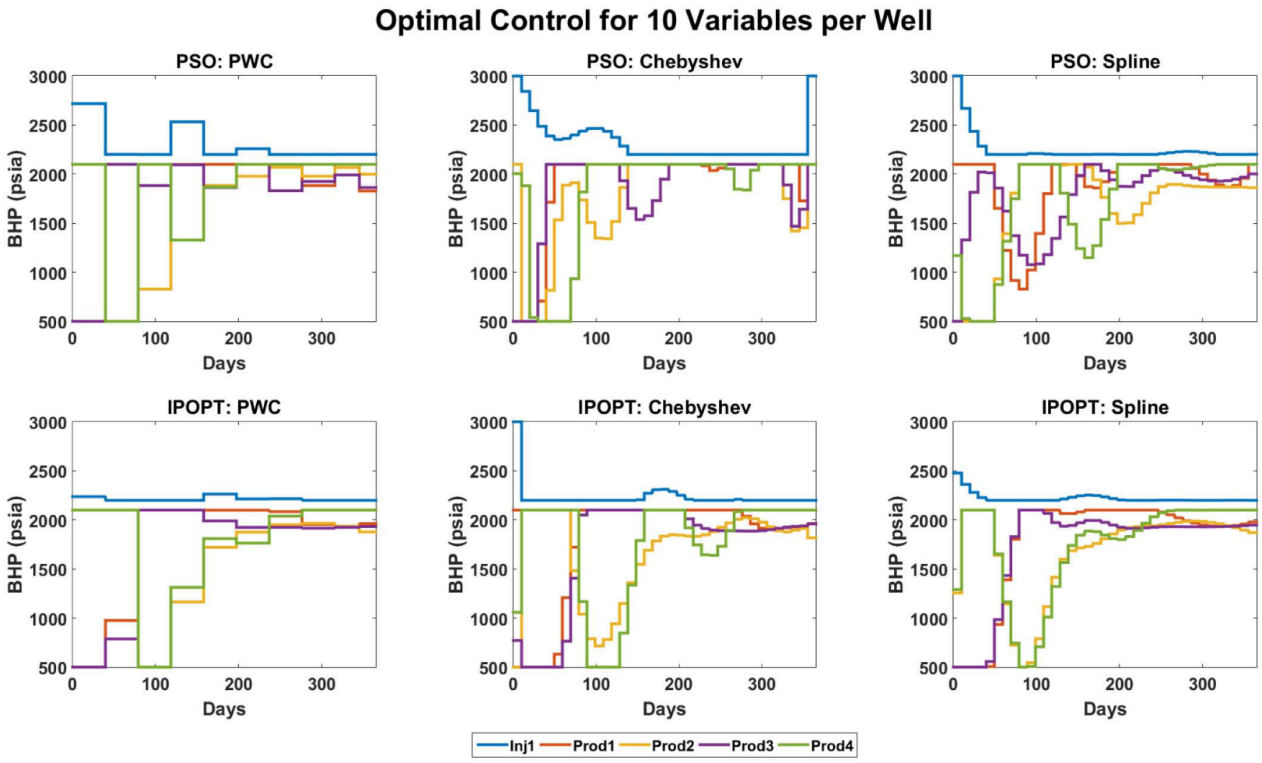


Figure 3.6: (Optimal controls of level 10 obtained by each parametrization method. Top: Gradient-free (PSO). Bottom: Gradient-based (IPOPT)

Thus, from this case study, it is evident that the polynomial control strategy (especially Chebyshev polynomial) obtains better NPV values for many levels of parameterization along with smooth control changes over a time period. This is much more practical as compared to the conventional piecewise constant parameterization that causes erratic changes in controls

between high and low values of BHP (as can be seen in Figure 3.6), which can cause equipment fatigue and hence impractical. Thus, the polynomial strategy is recommended here for practical purposes that has the ability to maintaining the near maximum or maximum NPV values.

3.4 Global Parametric Model Order Reduction

In this section, we introduce the PMOR strategy to be used with the different control parametrization schemes during control optimization. As mentioned before, the parameter in this case used is $\zeta = p_{wf}$ for all the wells. One challenge that needs to be addressed while developing PMOR method using POD is, we need to compute fine scale non-linear functions before projecting them to the reduced dimensional subspace, that does not significantly add to the computational advantage. And the second challenge is, during the control optimization, each optimization iteration will introduce a new set of BHP profiles, and thus, it is not feasible to run a fine scale simulation for each BHP set and develop a ROM. The former is solved here using DEIM (Discrete Empirical Interpolation Method) that constructs a separate subspace of non-linear terms, selects interpolation points by greedy algorithm, and then approximate the non-linear terms in the subspace by a combination of projection and interpolation [32]. The application of POD-DEIM for BHP control changes in reservoir simulation was introduced in [35, 20]. However, its full scale implementation for a control optimization problem is still lacking. The latter challenge is solved using global PMOR technique, where we seek to train the model with representative BHP profiles, that are deemed to represent the dynamics of the reservoir for different BHP profiles throughout the optimization run, to obtain a global PMOR basis.

3.4.1 Discrete Empirical Interpolation Method

DEIM is a simplified form of EIM (Empirical Interpolation Method), where the non-linear functions are evaluated at only a selected grid points and then approximated in the full grid by a combination of projection and interpolation.

Let $f(t) \in R^n$ be a non-linear function of time or other parameter. f can be approximated by projecting it into a subspace spanned by the basis function $\mathbf{U} = (U_1, U_2, ..., U_m) \in R^{n \times m}$ as

$$f(t) \approx Uc(t) \tag{3.17}$$

This basis function is determined by assembling the function evaluations in a matrix $S_f \in R^{n \times ns}$ and then SVD is employed to this matrix to compute m modes that are used as the basis functions.

The coefficient vector $c(t)$ can be determined uniquely from the following:

$$P^T f(t) = (P^T U)c(t) \tag{3.18}$$

where, $P = [e_{\gamma_1}, e_{\gamma_2}, ..., e_{\gamma_m}] \in R^{n \times m}$, and $e_{\gamma_i} = [0, ..., 0, 1, 0, ..., 0]^T \in R^n$ is the γ_i^{th} column of the identity matrix $I_n \in R^{n \times n}$. These interpolation indices P used for determining the coefficient vector $c(t)$ are selected inductively from the basis U by the greedy algorithm to have the non-singular matrix $P^T U$. The greedy algorithm is shown in Table 3.1.

Thus, from equations (3.17) and (3.18), we may write

$$f(t) \approx U(P^T U)^{-1} P^T f(t) \tag{3.19}$$

The non-linear functions (in our case the Transmissibility, Gravity, Accumulation, Source or Sink, Jacobian and the Residual) are thus computed only at the selected interpolation points by the greedy algorithm ($P^T f(t)$) and then approximated in fine scale by the above expression. For in depth discussions on the applications of POD-DEIM in reservoir simulation, the reader is referred to [20].

DEIM Algorithm [32]

$INPUT \quad \{u_l\}_{l=1}^{m} \subset R^n \ \ linearly \ \ independent$

$OUTPUT \quad \vec{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_m]^T \in R^m$

1. $[|\rho|, \gamma_1] = max|u_1|$
2. $U = [u_1], \ \ P = [e_{\gamma_1}], \ \ \vec{\gamma} = [\gamma_1]$
3. $for \ \ l = 2 \ \ to \ \ m, \ \ do$
4. $\quad Solve \ \ (P^T U)c = P^T u_l \ \ for \ \ c$
5. $\quad r = u_l - Uc$
6. $\quad [|\rho|, \gamma_l] = max\{|r|\}$
7. $\quad U \leftarrow [U \ \ u_l], \ \ P \leftarrow [P \ \ e_{\gamma_l}], \ \ \vec{\gamma} \leftarrow [\gamma_l \ \ \vec{\gamma}]^T$
8. $end \ \ for$

Table 3.1: DEIM procedure

3.4.2 Global PMOR training strategy

As we have discussed the PMOR method, POD-DEIM, that we use for well control optimization, the next task is to define the basis Φ for POD and U for all the non-linear functions in reservoir simulation during DEIM. The method we use here is global ROM where the aim is that the ROB yields a sufficient reduction in the dimension of the model such that it accurately represents the full-scale outputs over the entire parametric space (all BHP profiles encountered during the optimization process) as represented in equation (2.32).

For well control optimization, the quality of the resulting reduced order models (ROMs) is highly dependent on the parameter choice, which is BHP profiles over time, used during training period. Major issues arise in selecting a good parameter set for a representative ROB. One is the number of samples that need to be selected, which in general is an ad hoc

precess, where usually, the representative parameters are selected based on the knowledge of application being used for. The other challenge is computation of fine scale simulations for selected parameter which specially becomes prohibitive in case of high dimensional parameter space.

The way the samples are chosen are using either a priori sampling [42, 38] or adaptive sampling approaches [43, 91, 92, 21]. Adaptive sampling approaches tend to pick a database of samples that are close to the optimal database but are usually computationally expensive as they require evaluation of errors of the models. We use a priori sampling based on random selection of BHP profiles but we use the optimization strategy as part of the training procedure for efficient representation of parameter space.

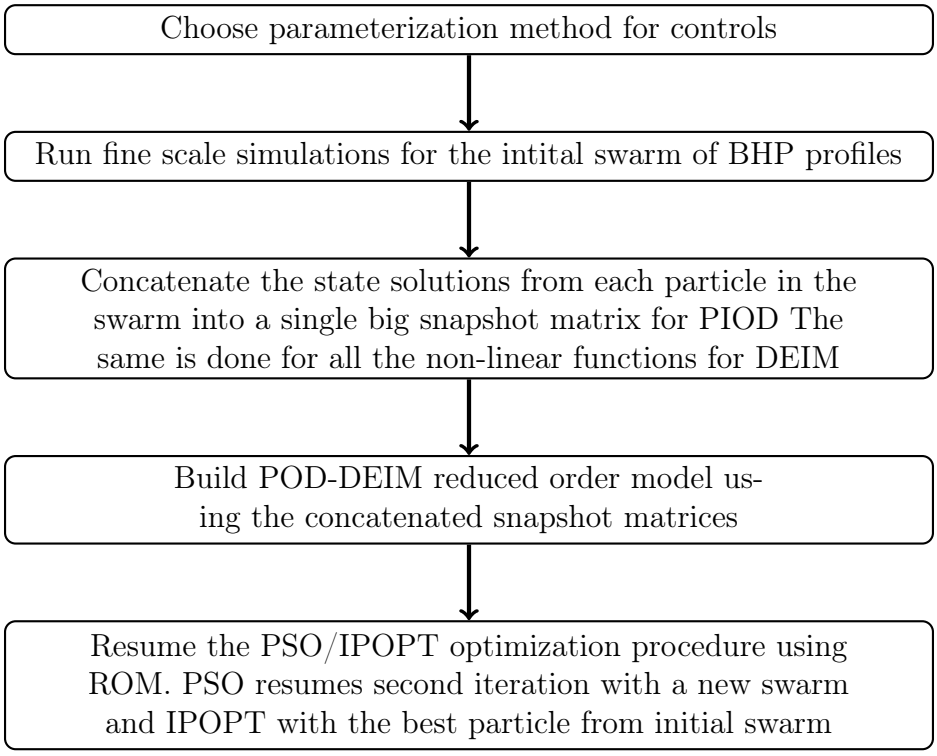


Figure 3.7: Workflow for optimization-online training of POD-DEIM

Our approach here, as shown in Figure 3.7, is to start the optimization with a few

fine scale simulation runs for different parameters and use them as an online training set. During the optimization process within the PSO framework, we spend a good amount of effort simulating different particles in parallel. Thus, assuming the initial swarm profiles will adequately represent the dynamics of the reservoir during the subsequent optimization iterations, we use the fine scale solutions of these particles from the first optimization iteration to construct the snapshot matrix. In more detail, in the gradient-free approach using PSO, the simulation state outputs of the initial random swarm are concatenated into one snapshot matrix. After constructing the POD-DEIM basis, the optimization proceeds to the second iteration with the reduced order model. Similarly, in the gradient-based approach we used a swarm of random BHP profiles for training, where the profile which produced the best NPV was chosen as the initial guess for the optimization procedure. By doing so, we utilize the training effort also to advance the optimization process.

3.5 Case Study

Our goal here is to analyze and compare the results of optimization using fine scale simulations and reduced order modeling with POD-DEIM for three different parameterization methods. Again, the optimization parameterization strategies used are Piece-wise Constant (PWC), Chebyshev Polynomial Approximation and Spline interpolation. Here we consider only one level of parameterization with 4 optimization variables per well for each one of the three methods. Thus, the number of decision variables for a five spot pattern is 20. We compare the results for both the gradient-based (IPOPT with LBFGS) and gradient-free (Particle Swarm Optimization (PSO)) optimizers. For this section, we used 8-core machines, and thus we set the swarm size to 8 particles to be evaluated in parallel. Here, we used a tighter convergence criterion of 20 stall iterations without a relative improvement above a threshold of 10^{-6}.

3.5.1 Problem Statement

In this section, we test our POD-DEIM methodology for the three different parameterization methods on a small model, which is a modified version of the one shown in [90]. The 2D model used is a $40 \times 40 \times 1$ reservoir (1600 grid blocks) with 4 producers and 1 injector placed in the channelized area. The model is discretized using Cartesian grid of size $20ft \times 20ft \times 20ft$. The porosity of the field is set constant to 0.2. Figure 3.8 shows the permeability field and the well locations. Here, we use a black-oil fully implicit simulator which solve equation (2.3) during each simulation time step. As mentioned earlier, we neglect the capillary pressure effects and consider a slightly compressible flow of oil and water. The initial saturation of oil and water is 0.8 and 0.2 respectively. The Corey-type relative permeability curves with exponent of 2 are used for both the fluids.

Initial reservoir pressure is 2100 *psia*. The injector has an upper bound of 3200 *psia* and a lower bound of 2200 *psia* while the upper bound for all four producers is set to 2100 *psia* and lower bound to 500 *psia*. Control time step for each well is 10 days. Oil price is considered to be $50\$/STB$ and the water-injection and production costs are each $10\$/STB$.

3.5.2 PMOR training

The snapshot matrix here is a (3200×1530) matrix collected from the 8 particles. The POD basis is constructed using 40 columns from the pressure left projection matrix and 70 columns from saturation left projection matrix. The number of basis for pressure and saturation are selected to capture 99% of the energy of snapshot matrix (see Figures 3.9a and 3.9b). The DEIM greedy algorithm selected 65% cells that need to be used for function evaluations (see Figure 3.9c). All optimization and parameterization methods use the same number of basis and interpolation cells. Here, we do not try to determine the strategies to reduce the number of interpolation points. This would be a future scope of study.

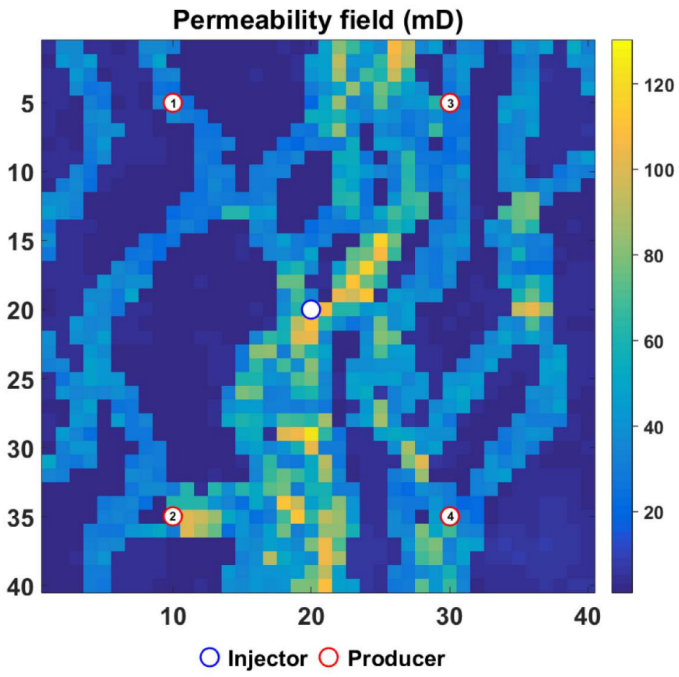
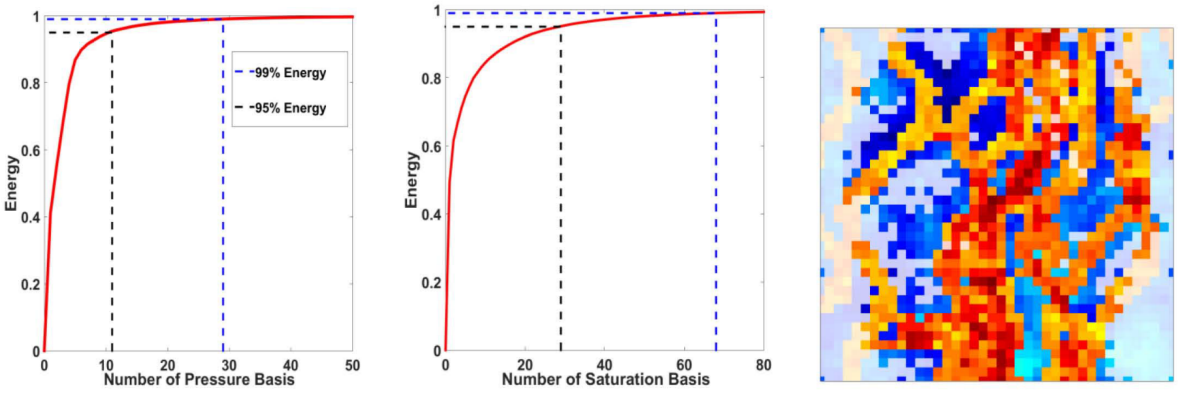


Figure 3.8: Reservoir permeability field with 4 producers and 1 injector (after [90])

3.5.3 POD-DEIM Performance for a Representative Case

We will now show a detailed analysis of the POD-DEIM performance. Out of the 6 POD-DEIM optimization runs (3 parameterization methods with 2 different algorithms), we chose one representative case to analyze its POD-DEIM performance. Our choice is Chebyshev Polynomial Approximation coupled with PSO. As will be shown in the next section, Chebyshev method obtained the best results in both optimization search methods. Figure 3.10 shows three training samples from the eight initial random Chebyshev profiles of four producers and an injector.

Next, we validate the accuracy of the reduced order model, by taking the Chebyshev optimal control from the POD-DEIM optimization as an input and comparing the simulation outputs from both full and reduced order models. As can be seen in the results below, the POD-DEIM produces accurate results. Figure 3.11 compares the saturation profiles for the POD-DEIM optimal control at three points in time: at the first time-step, after 1 year and



(a) (b) (c)

Figure 3.9: (a) Pressure and (b) Saturation basis selection for POD and (c) cells selected for DEIM (in strong colors)

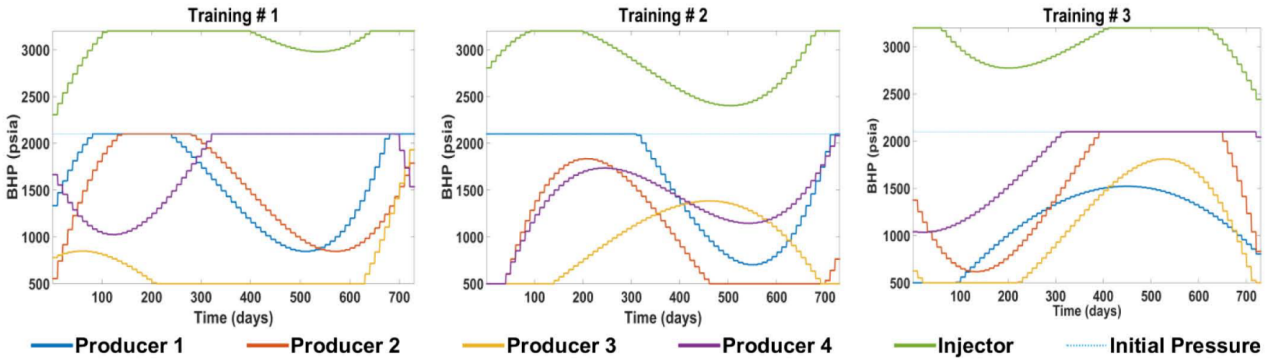


Figure 3.10: Three training samples from the initial eight-particles swarm of Chebyshev profiles

after 2 years. The comparison is between fine scale and reduced scale simulations. It also shows the relative error for POD-DEIM saturation at these times.

The oil production and water production rates for the POD-DEIM optimal control using fine scale and reduced scale simulations are shown in Figure 3.12 and Figure 3.13, respectively, which show a good agreement in the production rates between the two. The production rates are off in a few initial time-steps but do not have a large effect on the NPV values since these time-steps are very fine. Although we can increase the number of saturation basis to remedy this issue, we found it unnecessary at this point.

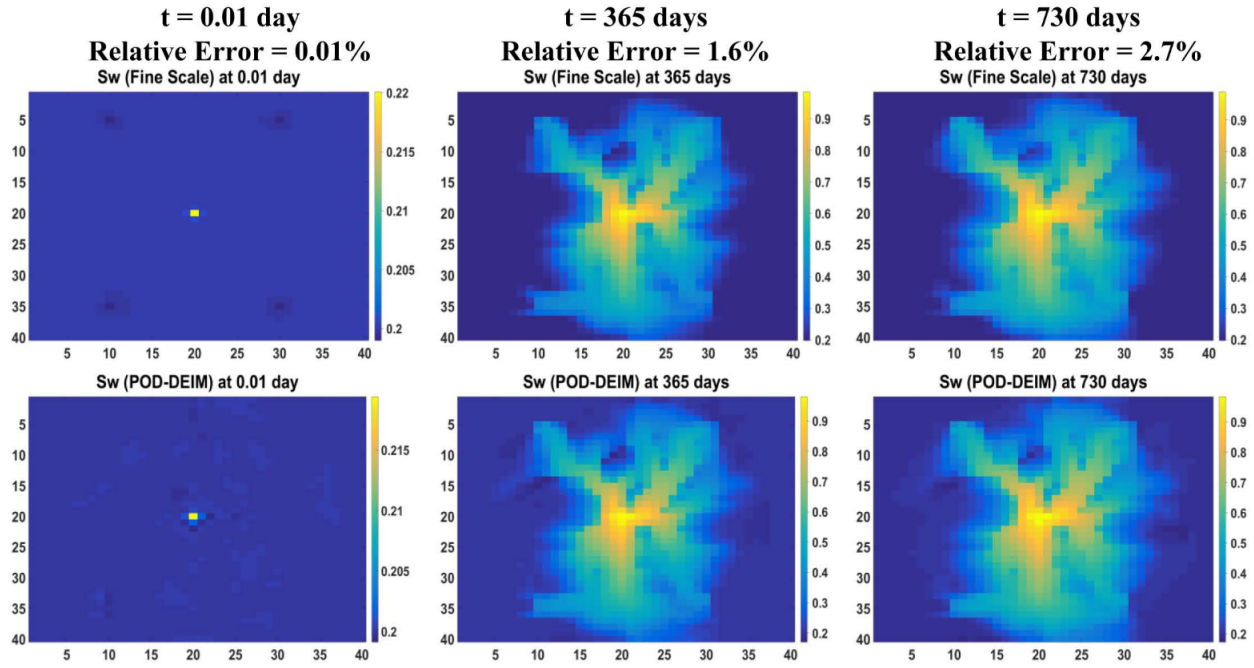


Figure 3.11: (Top) Saturation profiles from POD-DEIM Chebyshev optimal control using fine scale simulation and (Below) using the reduced model at different times and their relative errors

We performed similar accuracy analysis to all optimization techniques mentioned and they show a reasonable accuracy as the Chebyshev results. In the next section, we perform a different accuracy comparison, where we compare the optimal solutions obtained by an optimization using a full-order model to an optimization using a reduced order model.

3.5.4 Comparison: PSO Optimal Control Solutions

In this subsection, we show that global search method such as PSO with POD-DEIM can obtain reasonable optimization accuracy. For that purpose, we use a relative error indicator for the control trajectories, as defined below:

$$Error_{BHP}(\%) = \frac{\parallel BHP_{POD-DEIM} - BHP_{fine} \parallel}{\parallel BHPfine \parallel} \times 100 \tag{3.20}$$

where, $BHP_{POD-DEIM}$ is the optimal control obtained from optimization with POD-

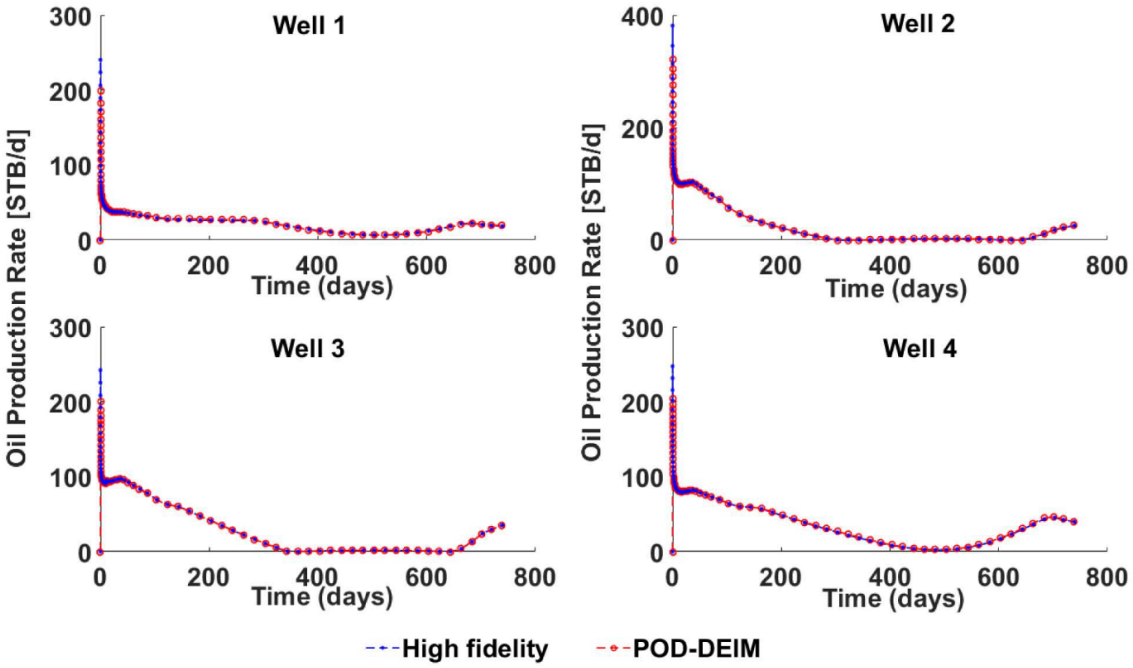


Figure 3.12: Oil production rate from the high fidelity simulation and reduced model simulation for 4 production wells using the Chebyshev optimal control solution from POD-DEIM.

DEIM and BHP_{fine} is the optimal control from fine scale optimization. Figure 3.15 shows the relative error, as defined in equation (3.20), for the control trajectories obtained by the PSO algorithm for the Chebyshev control, Spline control and piece-wise constant control. Figure 3.14 shows these optimal control trajectories. As is evident from these optimal control and relative error plots, the polynomial and piece-wise polynomial approximations from POD-DEIM and fine scale optimization show similar control strategies.

We can see from the tables that POD-DEIM proves to be a viable model reduction technique for production well control optimization problems especially for Chebyshev and Spline controls. We can see again that IPOPT outperforms the PSO.

In Figure 3.16, we compare the relative errors of the NPV estimated for each parameterization method using fine scale and POD-DEIM simulations. The relative NPV error is computed as follow:

$$Error_{NPV}(\%) = \frac{|NPV_{POD-DEIM} - NPV_{fine}|}{|NPVfine|} \times 100 \tag{3.21}$$

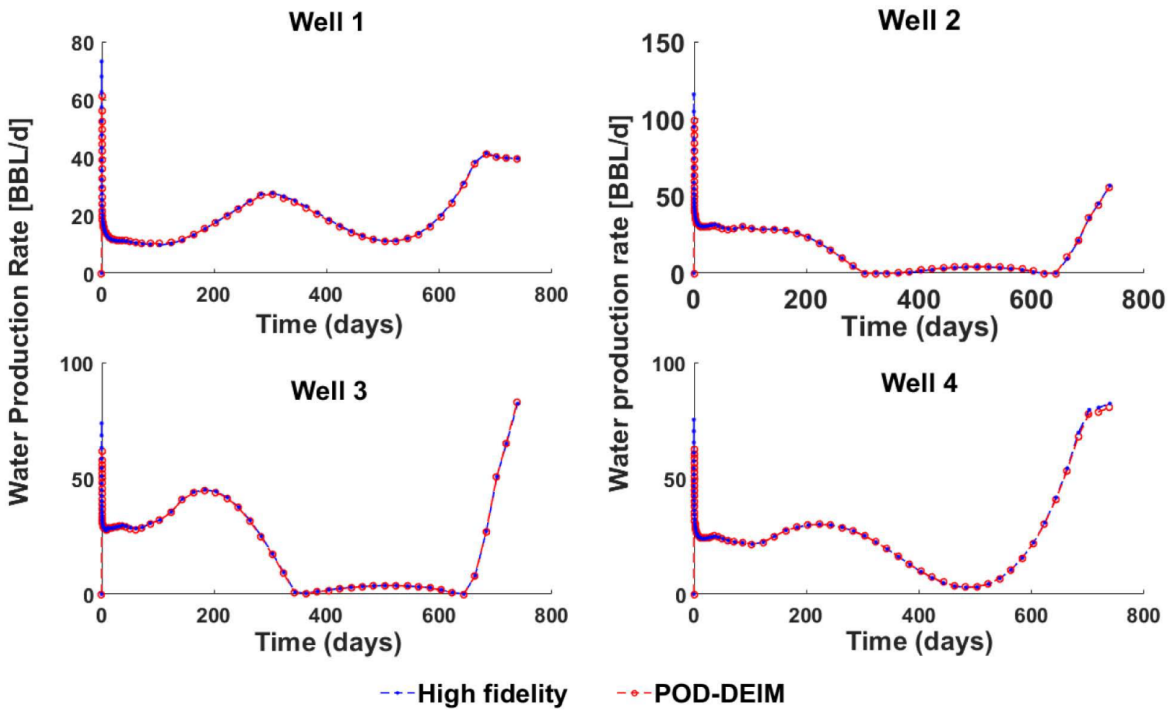


Figure 3.13: Water production rate from the high fidelity simulation and reduced model simulation for 4 production wells using the Chebyshev optimal control solution from POD-DEIM

where $NPV_{POD-DEIM}$ refers to the NPV obtained by using the optimal control from POD-DEIM with fine scale simulation and NPV_{fine} is the NPV from fine scale optimization. We can see from Figure 3.16, Table 3.2 and Table 3.3 that Chebyshev control optimization showed the best accuracy and the highest NPV for the two optimizers. Also, note from Table 3.2 and Table 3.3 that it had a fast rate of convergence when used with IPOPT for this example. NPV with PWC and Spline was less than that of Chebyshev, and Spline showed the least convergence rates for all the cases.

Overall we can say that the POD-DEIM with PWC did not show good results. One of the reasons could be a need for retraining during optimization or a large number of basis need to be selected for such control profiles to represent the dynamics of all the controls. Regarding runtime saving, one simulation run during the optimization on fine scale is about 20 seconds, whereas, using POD-DEIM takes on an average 12 seconds. Since the optimizations for all the

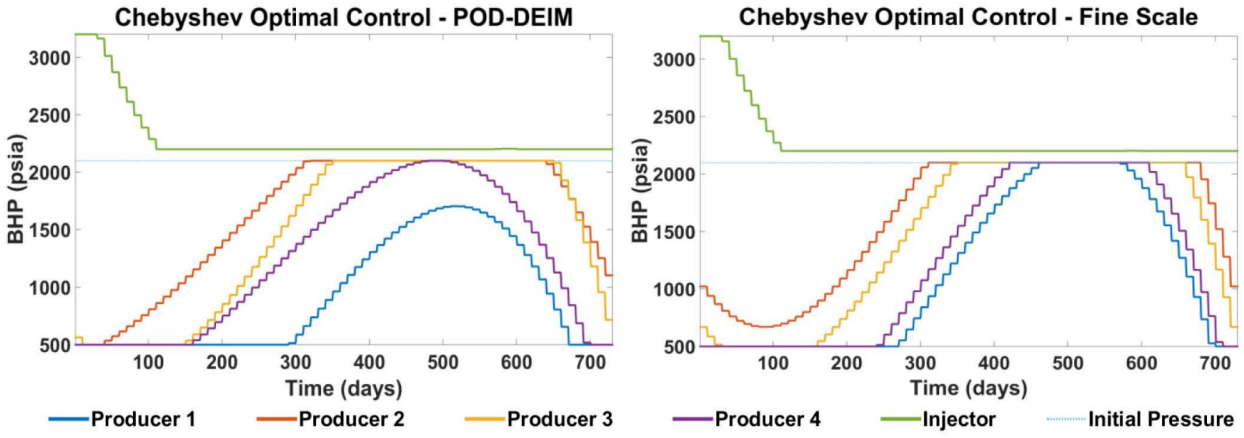
	Fine Scale		POD-DEIM		Fine Scale with POD-DEIM Control
	NPV	**Iterations**	**NPV**	**Iterations**	**NPV**
PWC	1.871	92	1.817	87	1.801
Spline	1.870	204	1.864	178	1.846
Chebyshev	1.881	46	1.903	25	1.875

Table 3.2: NPV and number of iterations for fine scale and reduced scale optimization using the 3 methods with IPOPT. NPV values are in million USD

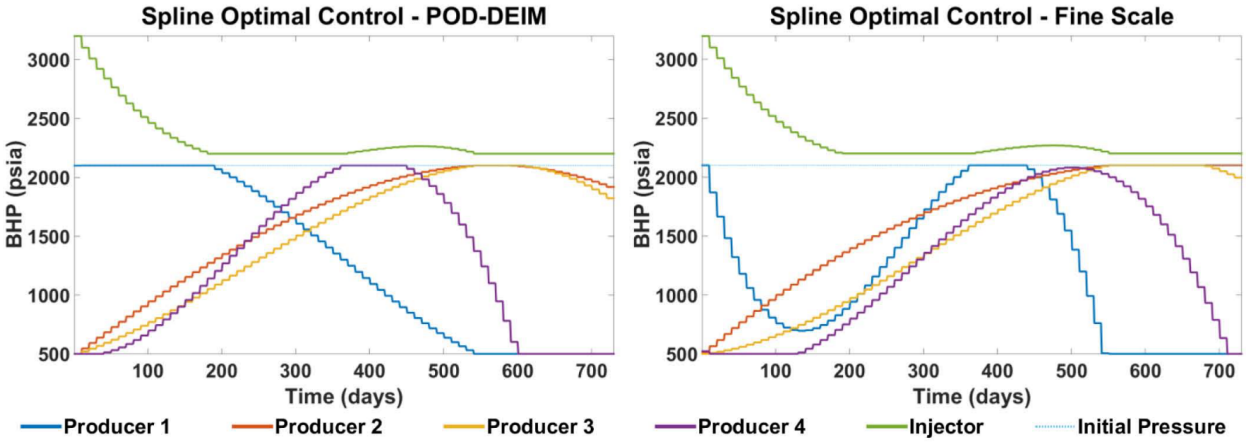
	Fine Scale		POD-DEIM		Fine Scale with POD-DEIM Control
	NPV	**Iterations**	**NPV**	**Iterations**	**NPV**
PWC	1.828	205	1.79	159	1.74
Spline	1.814	413	1.801	446	1.803
Chebyshev	1.84	262	1.878	280	1.839

Table 3.3: NPV and number of iterations for fine scale and reduced scale optimization using the 3 methods with PSO. NPV values are in million USD

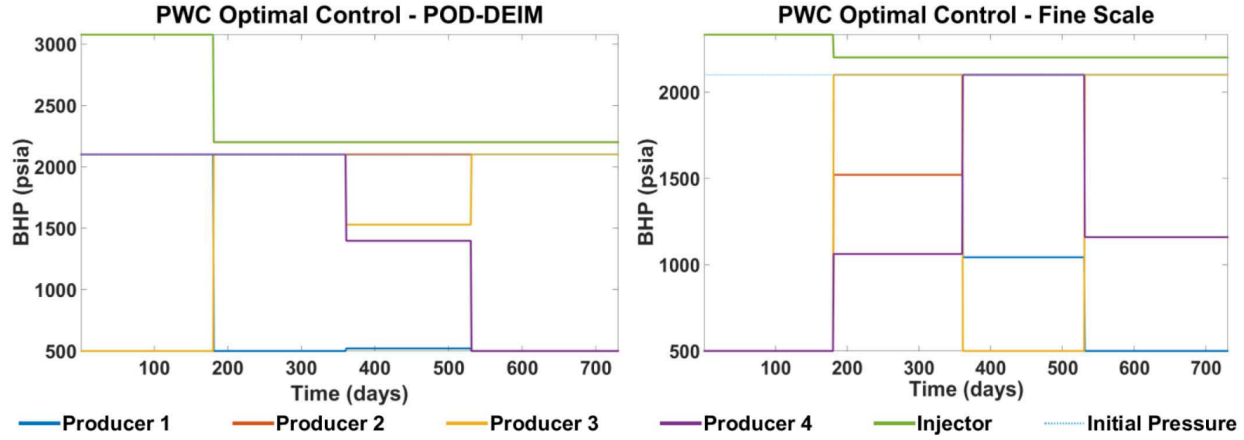
methods were performed on different platforms, we do not report the overall time. But the total computational saving on the entire optimization run in all the cases using the reduced model was about 45-50% (including the offline basis computation stage) when compared to the fine scale optimization. These results justify the use of POD-DEIM for well-control optimization problem and also a good set of training that is representative of all the BHP profiles during the optimization run particularly for Chebyshev and Spline control methods.



(a)



(b)



(c)

Figure 3.14: PSO optimal control solution comparison from POD-DEIM and fine-scale using all three parameterization methods (a) Chebyshev polynomial control (b) Spline control (c) Piecewise constant control

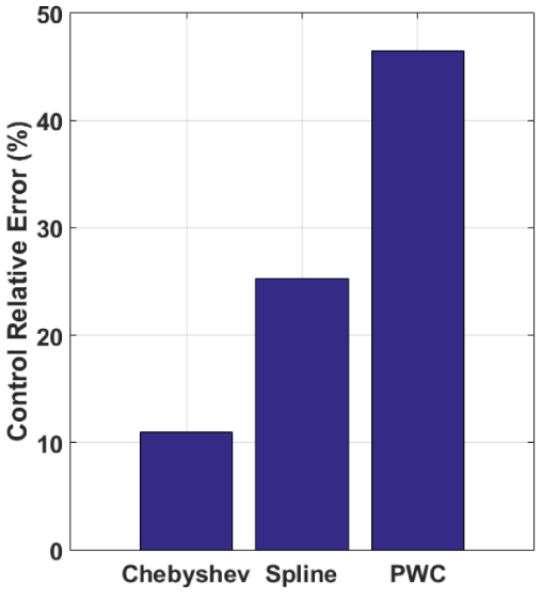


Figure 3.15: Control trajectories relative error for a PSO algorithm

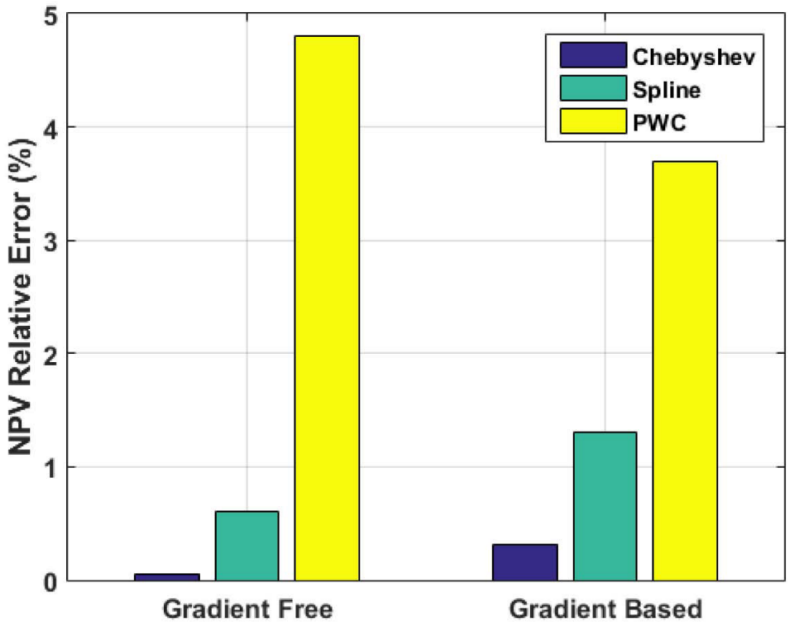


Figure 3.16: Control trajectories relative error for a PSO algorithm

3.6 Summary

In this work, we develop a global PMOR strategy using POD-DEIM, applied to a well control optimization procedure. We begin by introducing different parameterization methods to reduce the cardinality of the original infinite set of control-decision variables to a finite set. The parameterization techniques include a traditional piecewise constant (PWC) approximation, a newly introduced here, polynomial approximation by Chebyshev orthogonal polynomials and a piece-wise polynomial approximation by cubic Spline interpolation. These parameterization of controls during control optimization procedure is combined with POD-DEIM as the global model order reduction method for computational advantage at the reservoir simulation level, where a new training strategy is proposed that can be used as part of an optimization run.

In the first example, the parameterization methods are tested for increasing level of refinement using fine scale simulation to evaluate their performances for gradient-free (PSO) and gradient-based (IPOPT) methods. The results showed an increasing performance of optimization with respect to the objective function upon increasing the number of decision variables. For the highest refinement level, Chebyshev polynomial approximation found the maximum NPV for the case considered. PWC exhibited the fastest convergence, but with more likelihood to be trapped in local solutions. Spline method suffered from slow convergence when gradient-base method was used.

In the second example, we tested the control parameterization strategy with POD-DEIM for a given level of parameterization refinement. We observed that Chebyshev polynomials performed better than Spline control and PWC control. The training selection for POD-DEIM here utilizes the information from the first fine scale optimization iteration on multiple random particles run in parallel, which is then used as a snapshot matrix. The optimization then follows from the next swarm chosen by PSO in case of gradient free method and from the best solution as the initial guess for the IPOPT method.

Choice of such a training set proved to perform consistently well for the polynomial

and piece-wise polynomial control approximations. POD-DEIM when used for optimization using Chebyshev polynomial approximation, produced very good accuracy with the fine scale optimization as was evident from the errors in optimal control solution and NPV. A 45-50% speedup on the reduced scale optimization runs was achieved for all the cases compared to high fidelity runs. This work demonstrates the use of control parameterization with model order reduction as an efficient methodology for optimization particularly when used with polynomial controls.

In future work, the use of this methodology on benchmark models and bigger duration of well control optimizations should be investigated. These cases performed significantly well without a need for retraining. However, it might be necessary to retrain the model during the course of optimization run for more complex cases, which may require a robust error indicator to quantify the errors from POD-DEIM.

Until now, we discussed about the use of global basis for well control optimization problem. MOR for well control changes can also be described in terms of local basis in time which we try to address in the following section. Thus, this is a preliminary work shown to address the idea of generating multiple realizations of local ROM solutions using Bayesian formulation as an initial step for future research direction.

3.7 Bayesian framework for probabilistic sampling of basis functions

This section describes the preliminary work on a Bayesian framework to sample the basis function for well control optimization problem. We note that this topic is not in continuation with the PMOR strategy discussed until now in the above sections. Here, the idea is to describe the solution of a new well control from multiple realizations of inexpensive reduced models rather than a single global reduced model. Thus, this method, rather than providing a deterministic solution, can take into account multiple low confidence solutions and thus, obtain a probabilistic description of approximate solutions. This concept was applied for a multiscale approach based on finite element method in [93]. The work that we show here is not complete but provides a good starting point as a proof of concept and some areas need to be addressed in the future, as will be discussed later.

As we seek to obtain realizations of basis functions from a given set of basis functions, we adopt the following procedure:

1. The first step is to compute the global basis functions for a training set of BHP profiles using POD. We then use a subset of these basis functions as permanent or "fixed" basis functions and the rest are assumed to be the "tail" basis functions. Fixed basis functions provide a good solution approximation.

2. We then use the residual information at each time step to compute the prior distribution of the "tail" basis functions.

3. Once the prior distribution is obtained, we then define the posterior that includes residual minimization and then sample from the posterior using sampling techniques.

In the next section, we show these steps in detail. But before that, we do not go into the

details of the offline basis construction as it has been already shown in the snapshot based POD method.

3.7.1 Bayesian Formulation

The POD basis functions which we denote by $\phi(x)$ are a function of space x and the solutions are obtained as a linear combination of these basis functions at each time step. For the current work, we just show the working of this method for time interval $[t_{n-1}, t_n]$. Since we consider two-phase flow system, we have different basis functions for both pressure and saturation. For the ease of notations we describe the method using a single state variable and its basis functions. Let us denote the reduced model predicted solution by $\tilde{\mathbf{x}}$ and the coefficients of the basis functions as β. In our case, β corresponds to the reduced order correction in the states obtained at each time step as represented by δ_r in equation (2.31). The reduced order model solution can then be written as:

$$\tilde{\mathbf{x}}^n(x,t) = \sum_{i=1}^{r} \beta_i^n(t)\phi_i^n(x) \tag{3.22}$$

$$= \sum_{i=1}^{N_{fix}} \beta_i^n(t)\phi_i^n(x) + \sum_{i=N_{fix}+1}^{r} \beta_{i,+}^n(t)\phi_{i,+}^n(x) \tag{3.23}$$

Here, N_{fix} denotes the number of fixed basis functions ϕ_i and $\phi_{i,+}$ denotes the tail basis functions. r is the dimension of POD basis from which the basis functions are sampled. For the current problem we do not consider the time dependency of the basis functions since we consider a global basis in time and for each time interval, same fixed basis are considered. Let us denote the residual computed at the time interval $[t_{n-1}, t_n]$ by R^n which is computed using the fixed basis function ϕ_i. As we keep ϕ_i fixed, we then select N_{basis} basis functions for prior distribution, which is user specified and depends on the computer resources available. For each tail basis we find the correlation coefficient $\alpha_{k,+}$ as:

$$\alpha_{k,+} = corrcoeff(R^n, \phi_{k,+}) \tag{3.24}$$

We normalize the correlation coefficients $\alpha_{k,+}$ so that on average we have N_{basis} basis

functions.

$$\hat{\alpha}_{k,+} = \frac{\alpha_{k,+}}{\sum \alpha_{i,+}} N_{basis} \tag{3.25}$$

Thus, for prior distribution of basis functions, we choose k^{th} basis with probability $\hat{\alpha}_{k,+} \wedge 1$. Thus, we basically sample from a number randomly from a uniform distribution and select the basis that has the $\hat{\alpha}_{k,+}$ value greater than this number. As we use $\hat{\alpha}_{k,+}$ for prior distribution, we may use approximate residuals and hence for the cases shown here, we just use one Newton iteration to get the residual for correlation.

Posterior around previous time

Now that we have the prior distribution, we formulate the posterior function. Denoting the indices of the basis functions as $\mathcal{I}$, we sample from the posterior at each time interval based on the solution from the previous time interval. Thus from the Bayesian theorem we may write the posterior as:

$$P(\beta^{n+1}, \mathcal{I}^{n+1} \mid \tilde{\mathbf{x}}^n(x,t)) \sim P(\tilde{\mathbf{x}}^n(x,t) \mid \beta^{n+1}, \mathcal{I}^{n+1}) \pi(\beta^{n+1} \mid \mathcal{I}^{n+1}) \pi(\mathcal{I}^{n+1}) \tag{3.26}$$

$$P(\tilde{\mathbf{x}}^n(x,t) \mid \beta^{n+1}, \mathcal{I}^{n+1}) \sim exp\bigg(- \frac{\parallel R^{n+1} \parallel^2}{\sigma_L^2} \bigg) \tag{3.27}$$

where, R^{n+1} is the residual computed for time interval $[t_n, t_{n+1}]$ and σ_L represents numerical precision. Thus, the posterior is computed based on the residual information from the solution at previous timestep.

Sampling from the posterior

Here, we use Gibbs sampling method to sample the posterior given the prior, as stated in previous section. Let us consider our prior distribution as $\mathcal{I}_+^j = [\mathcal{I}_{1,+}^j,\ \mathcal{I}_{2,+}^j,\ \mathcal{I}_{3,+}^j,\ ...,\ \mathcal{I}_{n,+}^j]$. Here, each $\mathcal{I}_{i,+}^j$ represent the index of the tail basis functions. Given this prior, we sample a posterior $\mathcal{I}_+^{j+1}$. Gibbs sampling computes the probabilities $\hat{\pi}_k$ of each additional basis

function:

$$\frac{\hat{\pi}_{i,+}^{n+1}}{1-\hat{\pi}_{i,+}^{n+1}} = \frac{\hat{\alpha}_{i,+}^{n+1}}{1-\hat{\alpha}_{i,+}^{n+1}} \times \mathcal{R}_i \tag{3.28}$$

where, $\mathcal{R}_i$ is the multiplicative factor that defines how much a change in one basis function affects the residual change.

$$\mathcal{R}_i = exp\bigg(- \frac{\parallel R_i - R_{-i} \parallel}{\sigma_L^2} \bigg) \tag{3.29}$$

Thus, for example, when calculating the probability of first tail basis function, R_1 is calculated using $[\mathcal{I}_{1,+}^j = 1,\ \mathcal{I}_{2,+}^j,\ \mathcal{I}_{3,+}^j,\ ...,\ \mathcal{I}_{n,+}^j]$ and R_{-1} using $[\mathcal{I}_{1,+}^j = 0,\ \mathcal{I}_{2,+}^j,\ \mathcal{I}_{3,+}^j,\ ...,\ \mathcal{I}_{n,+}^j]$. Once $\hat{\pi}_{i,+}^{n+1}$ is calculated, we randomly sample from a uniform distribution $r \sim N(0,1)$ and set $\mathcal{I}_{1,+}^{j+1} = 1$ if $\hat{\pi}_{i,+}^{n+1} > r$, otherwise, it is set to 0. This step is repeated for each tail basis function to get a posterior sample $\mathcal{I}_+^j$.

3.7.2 Case Study

We demonstrate this formulation on a simple two dimensional heterogeneous reservoir model as shown in Figure 3.17. The reservoir model is discretized with a Cartesian grid of size 50 ft × 50 ft × 50 ft, and it contains 400 (20×20) active cells. We neglect capillary and gravity effects. The initial reservoir pressure is 4200 psi and the injector and producer are BHP controlled. The training and test BHP controls for both injector and producer are shown in the Figure 3.18. Simulation is run for 3 years and POD is used to obtain the basis functions from the state solutions obtained in this time duration of 3 years. We aim to use the methodology to sample the posterior and get multiple realizations of the basis functions for a time interval in the simulation run. The time interval that we choose here is the last time step to predict the new states using many but very fast low order models.

Once we get the basis functions, we fix first few basis for both pressure and saturation. Here 99.99% energy criteria was chosen to get the basis dimensions which resulted in 40

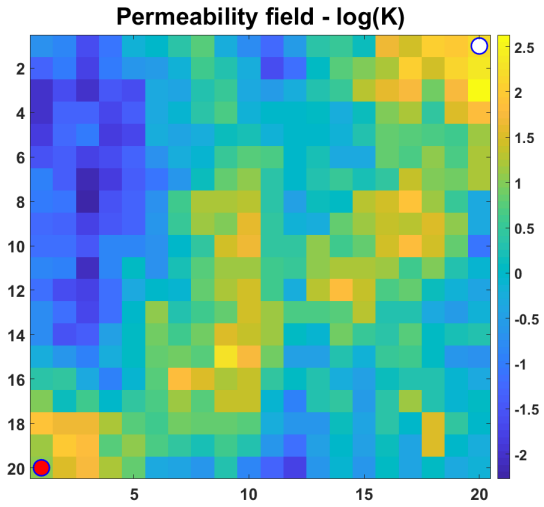
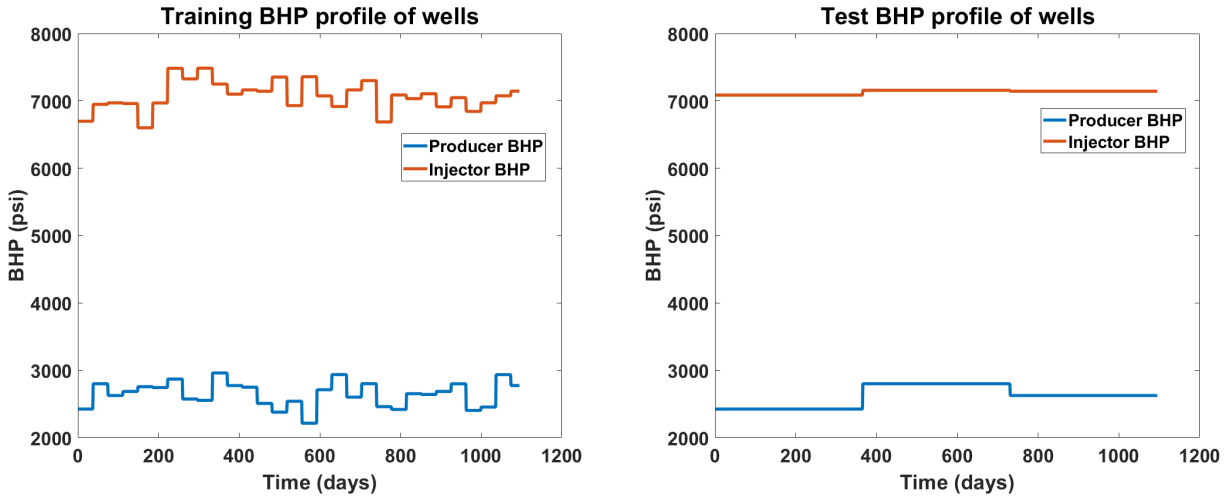


Figure 3.17: Heterogeneous permeability field with one producer (red) and one injector (white)



(a) (b)

Figure 3.18: (a) Training BHP profile for producer and injector (b) Test BHP profiles of producer and injector

pressure basis and 60 saturation basis. We fix first two basis for both pressure and saturation and then use the proposed methodology to sample the tail basis functions. One important parameter in this method is determining N_{basis} number of basis functions we wish to have

on average. As can be expected, the more this number, the closer it will be to the exact solution. Thus, here we try different values of N_{basis} to evaluate the performance. Another parameter is σ_L that represents the numerical precision. Thus, lower this number, we have less confidence in our samples and the sampling method will pick more number of basis functions on an average. We generate 20 posterior realizations of basis functions to compute the coefficients β and hence the POD predicted states at the given time interval and an average of these state solutions is compared to the true solution.

Figures 3.19 and 3.20 show the predicted pressure and saturation solution after 3 years (last time interval) using $N_{basis} = 8$ and $\sigma_L = 2$. Note, the predicted solutions are the average of solutions obtained from 20 posterior samples. Table 3.4 shows the relative error in prediction of pressure, saturation and oil production rate at the end of simulation for different values of N_{basis}. Thus, these results are more accurate with increasing number of basis required as expected. In this way, we obtain many approximate solutions that are extremely fast to solve because only a few basis functions are sampled in the posterior at each time interval. It is also interesting to see the basis functions selected for pressure and saturation. Table 3.5 shows the indices of the tail basis functions for pressure and saturation for one of the samples. It can be observed that as opposed to the convention to select first few basis functions, many basis functions towards the end are sampled in the posterior.

$\mathbf{N_{basis}}$	**Pressure**	**Saturation**	**Oil Production Rate**
5	1.9	1.4	4.7
6	2.1	1.4	2.1
8	1.7	1.3	0.9
10	1.2	1.3	0.02

Table 3.4: Relative error (%) in predicted solutions by posterior samples of basis functions

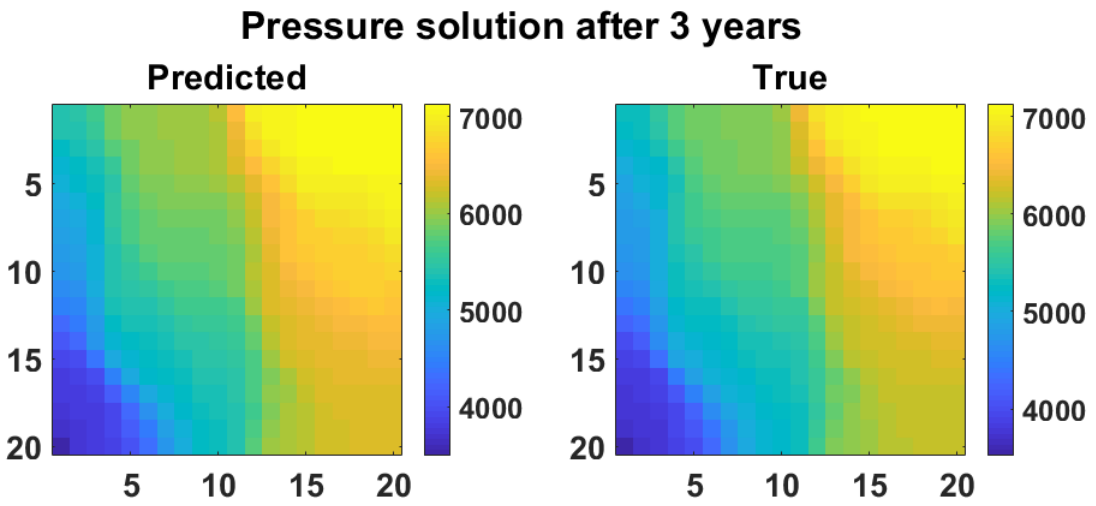


Figure 3.19: Pressure solution comparison at the last time step for which basis functions are sampled - $N_{basis} = 8$

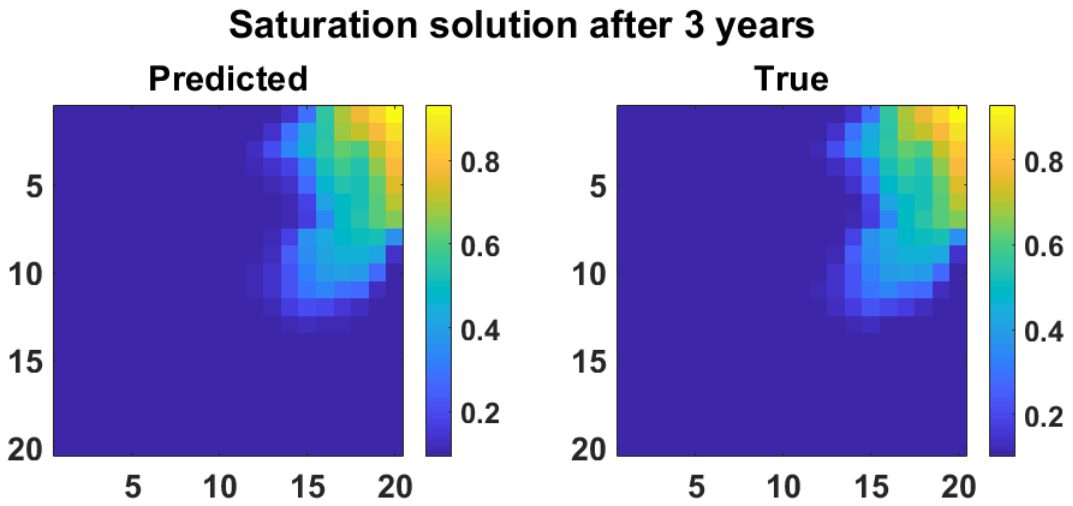


Figure 3.20: Saturation solution comparison at the last time step for which basis functions are sampled - $N_{basis} = 8$

3.7.3 Summary

In this work, we propose a Bayesian formulation for solving reduced order models within a probabilistic setup, which provides multiple inexpensively computed solutions to account for uncertainties in the reduced model solution. This method is applied to a reservoir simu-

	Indices of the basis functions
Pressure	6, 7, 9, 12, 14, 16, 25
Saturation	2, 4, 5, 24, 47, 58

Table 3.5: Indices of the pressure (38 basis) and saturation (58 basis) tail basis functions selected in one of the 20 samples

lation application of changing well controls during well control optimization. The proposed framework is based on the understanding that a first few basis functions, referred here as "fixed" basis functions, of the state space can account for the dominant dynamics in the reservoir and hence reduce the error significantly and, the rest of the basis functions, referred to here as the "tail" basis, reduce the error only slightly. Thus, as opposed to the convention of selecting global basis for all time steps, we sample a few tail basis functions at each time step which allows us to compute multiple realizations of the state solution, each of which is computed significantly faster. A residual based posterior is proposed, which is then sampled using Gibbs sampling technique. The validity of this method is demonstrated on a changing well control problem for a single time step and shows promising results. This work is in the preliminary stage and requires a full scale implementation in the future. One of the areas that need to be focused is the computation cost associated with residual calculation which is non-linear for reservoir simulations. This makes the workflow slow to implement and hence some quick approximation to residual must be sought. In general, this framework shows the promise to represent a state solution by generating multiple realizations of very fast computed reduced order solutions.

4. LOCAL MODEL ORDER REDUCTION FOR CHANGING WELL LOCATIONS DURING WELL PLACEMENT OPTIMIZATION*

After developing the PMOR strategy for well control optimization, we now focus on the PMOR development for the case of changing well locations. This is the main focus of my research from this point. However, it should be noted here again that, this is just a first step towards well placement optimization. We only address development of MORs for changing well locations that can ultimately be used for well placement optimization problem in the future.

In this section, we develop a local model order reduction technique where the idea is to use pre-existing reduced order basis (ROBs) for a new well location in the reservoir. Most parts of this work have been published in [94]. We begin with the main challenges underlying the PMOR development for changing well placement followed by some observations using a simple case study that aid in constructing the reduced models. The idea used here is based on construction of error maps for a ROB and these observations motivate us to use concepts in machine learning (ML). As we develop a POD-based strategy, we later formulate the basis selection process as a regression and classification machine learning problem. We then provide a brief overview of the machine learning algorithms used here with the reason behind using them. Later, we address the feature selection process for the formulated ML problem that is the main challenge for the case of new well locations, which is not as straightforward to define as the changing well controls. The features are defined based on the geometry of well locations and we also introduce flow diagnostics based features to capture the physics of flow. The validity of the proposed methodology is shown with some case studies. The current scope of work for this method is limited to low contrast reservoirs and small simulation run times such that we do not have water production in the wells. We also show the limitations

*Reprinted with permission from "Th A2 08 Parametric Model Order Reduction For Adaptive Basis Selection Using Machine Learning Techniques During Well Location Opt" by H. Zalavadia and E. Gildin, 2018, 15th European Conference on the Mathematics of Oil Recovery, Copyright 2018 by EAGE

of this method when considering the realistic case of water fronts reaching the wells and the possible ways to address them in the future.

4.1 MOR challenges for changing well locations

Determining the optimal well locations is a crucial decision during a field-development plan as it plays a significant role towards the value of the project. This optimization problem has been approached as a discrete optimal problem by researchers using gradient-free optimization algorithms like the genetic algorithm [95, 96] and neural networks [44]. It was formulated as a continuous optimization problem and solved with gradient based methods in [97] and using gradient free methods in [98]. However, these techniques require a large number of fine scale forward simulation runs to obtain convergence and thus require significant computer time, which motivates us to develop MOR strategies for changing well locations.

As discussed in Section 1 and Section 3, ROM approaches have been applied in the literature within the context of well control optimization problems with fixed well configurations and have reached a good level of maturity to achieve significant computational saving. During well control optimization, there is usually only a slight change in controls moving from one timestep to other. This makes the aforementioned MOR techniques more suitable for control optimization. But these approaches are not well suited for well placement optimization since a slightest change in well configuration can lead to a totally different dynamical system, something that has not been addressed yet and becomes very difficult to develop a ROM for new well location as was shown in the controllability concept explaining changing well location. The selection of parameters i.e. well locations during the training stage is thus a challenge that can act as an appropriate controllable subspace for new well locations.

The parameter ζ of the system depends on the problem under consideration as discussed before. For well control optimization, where, $\zeta = p_{wf} \in R^{N_t}$ for each well, with N_t being the number of simulation time steps, for well placement optimization, ζ represents the indices in sparse vector $\mathbf{Q}$, corresponding to spatial location of the well to be optimized in the source/sink vector, which is the main focus of this work. Thus, a major challenge is to

explicitly represent ζ for well locations such that it represents accurately the dynamics at new well locations in reservoir simulation equation (2.3), as opposed to the parameter for well control, which is a real valued number corresponding to BHP at different time steps.

4.2 Observations

As a starting point to tackle the challenges that are underlying the MOR development for changing well locations, we first try to solve a simple problem by making some observations that will lead us to formulating our main contribution. So we discuss now these observations that motivated us to develop the methodology proposed later in this section. In this section, we first analyze the results using widely used global PMOR techniques that show potential benefits and drawbacks that can be addressed by using local basis as will be discussed later in this section. The cases considered here are reservoirs with one producer and one injector. The parameter of interest here is the producer well location, so we keep the injector well position same throughout. The ROM methodology proposed in this work is based on the assumption that all the wells are controlled with same BHP profiles such that the well rates are affected only by the well position, represented by ζ. For this work, we also note that, the results are analyzed for the cases where water-front does not reach the producers for most well locations i.e. the simulations are run for a short time (1 year) and the permeability fields have low contrast, i.e. the range of order of magnitude change in the permeability values is small.

4.2.1 Global PMOR results

For a parametric PDE, construction of ROMs can be achieved by generating a global basis, by adding fine scale solution information from different well locations into a single snapshot matrix before computing SVD. As was discussed in the global ROB strategy used in my research for well control optimization, for a global basis, the parameters that are representative of the parameter domain are chosen a priori or using adaptive greedy algorithms. The a priori sampling of parameters is fast but can include unnecessary samples or miss

important samples. The adaptive sampling strategy is more accurate but computationally expensive as it needs repeated evaluations to look for parameters that produce the maximum ROM error. Just for the sake of analysis for the case of changing well locations, we use the adaptive selection strategy using a brute force technique, by computing ROM errors at all well locations for a small case and then recursively selecting the locations with maximum ROM error.

We demonstrate the global basis result for different well locations in a heterogeneous channelized reservoir. POD is applied to a two-phase flow (oil-water) reservoir model with one injector well and one producer well as shown in Figure 4.1, using method of snapshots described in Section 2. The reservoir model is discretized with a Cartesian grid of size 30 ft $\times$ 30 ft, and it contains 400 (20$\times$20) active cells. We neglect the capillary and gravity effects. The initial reservoir pressure is 4200 psi and the injector and producer are BHP controlled at a constant pressures of 7000 psi and 2500 psi respectively. After we get the basis for this well configuration using POD, we use it to find percentage ROM error in oil production rate (Equation. (4.1)) for all the other cells as potential locations for producer, as shown in Figure 4.2. POD dimensions for pressure and saturation are chosen based on the 99.99% energy criteria. Pressure basis dimensions is higher than that of saturation as the simulation is run for a short time and the saturation solution can be represented by fewer basis due to slow dynamics. The next well location in the database is chosen as the cell with maximum error. We run the fine scale simulation using this new parameter to get a new snapshot matrix. This is concatenated with the previous snapshot matrix to get a new ROB. The same process is repeated. Here, in the figures, L_p and L_s represent the number of pressure and saturation basis selected and N_{acc} represents the number of gridblocks with error less than 1%.

$$Error(\%) = \frac{\parallel q_{ROM} - q_{fine} \parallel}{\parallel q_{fine} \parallel} * 100 \tag{4.1}$$

As we note here that, concatenating the snapshots from 2 well locations result in error

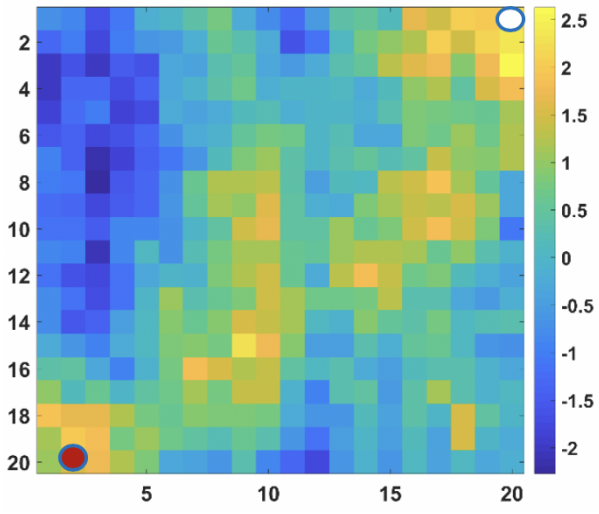


Figure 4.1: Normal scale k_{norm} of permeability field. The values of permeability are chosen as $5^{k_{norm}} mD$ to exhibit low contrast. Red indicates the producer well and white is the injector

reduction over many well locations. However, the decrease is not strict when a third location is added for concatenation. There are many well locations where the error actually increases. The basis dimensions however strictly increase as expected for both pressure and saturation as more snapshots are concatenated.

4.2.2 Local PMOR results

From the above results, we can see that a global basis lead to high dimensional subspaces and does not necessarily produce strictly decreasing errors at many well locations. In this section, we use local PROM using pre-existing ROBs from a database $\mathbb{D}$ of well locations to investigate their behavior. The first scenario is to check the behavior of a local ROB on the parameter space. The well configuration is as shown in the Figure 4.3a. The production well is located at grid block $(2, 19)$ and injector well at grid block $(20, 1)$. We use POD to get the ROB for this configuration with dimensions of pressure and saturation based on the 99.99 % energy criteria. This basis is now used for all the possible production well locations

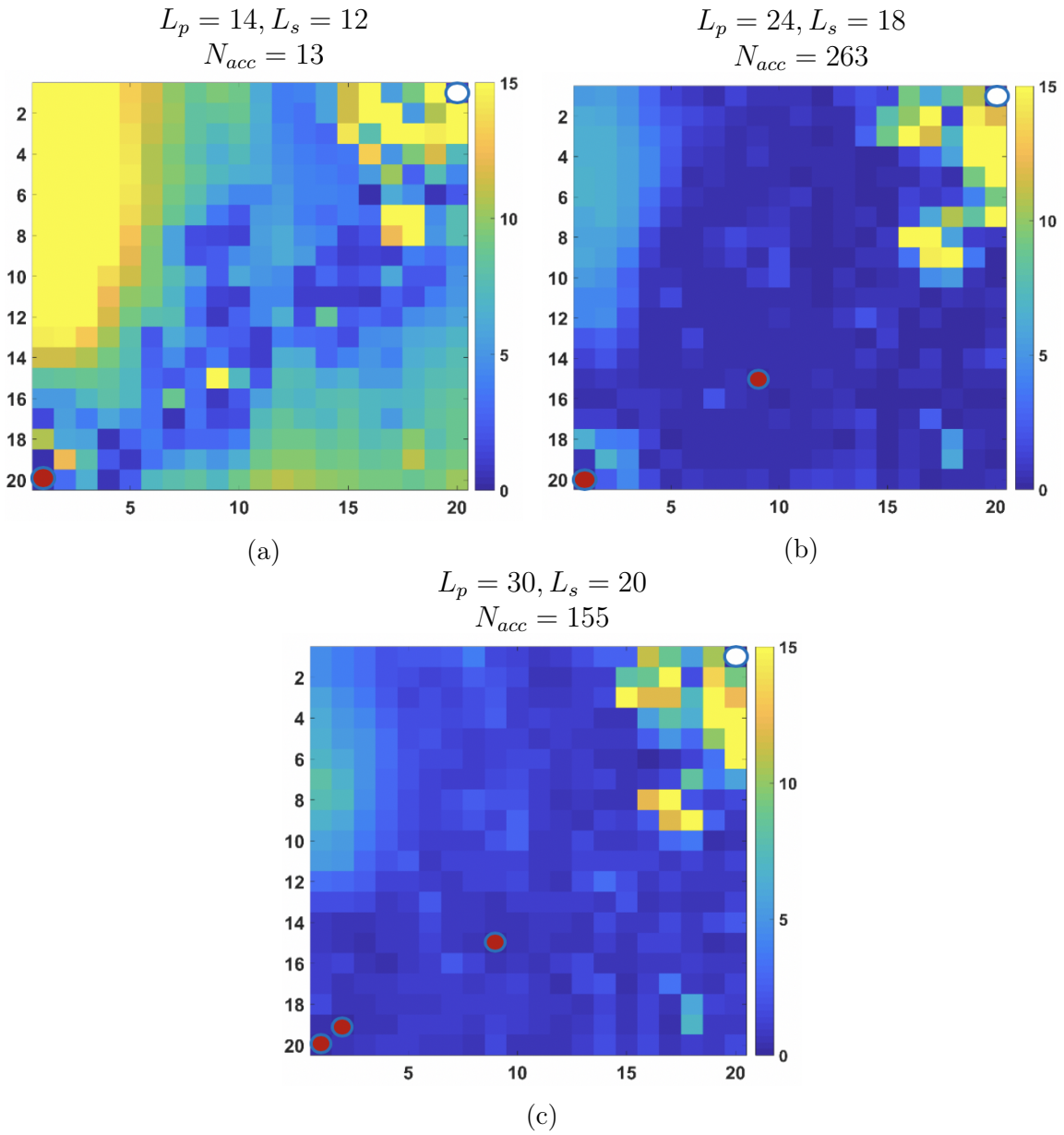


Figure 4.2: Global basis construction using greedy search. At each stage from (a) – (c), a producer well is set to the location of maximum error to concatenate the snapshot with that from previous configuration

and the error map is obtained as shown in Figure 4.3b. Many parameters produce a small ROM error when a same basis is used. For example, Figure 4.4 shows POD predicted oil production rate at one such well location (6,17), that is the low error location in the error map. The area in the top right corner (area swept by injected water) of the error map shows

high errors as expected, since the basis used have no dynamical information about the water flooded area.

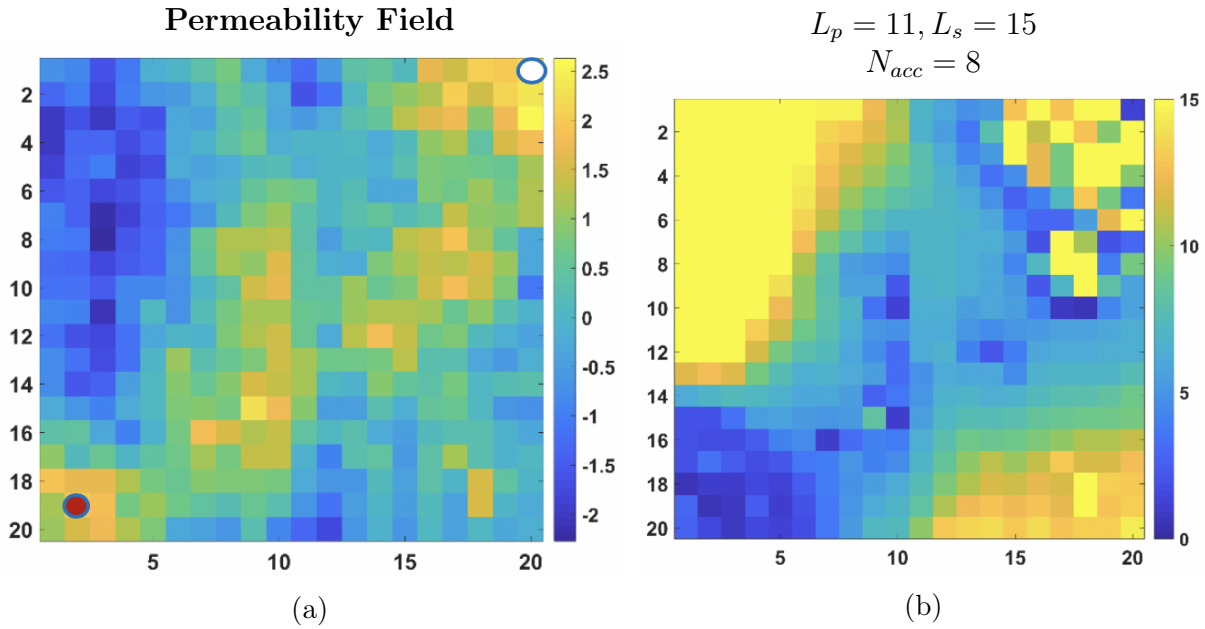


Figure 4.3: (a) is the permeability field with well configuration shown. (b) shows the error map for this well configuration and the given basis

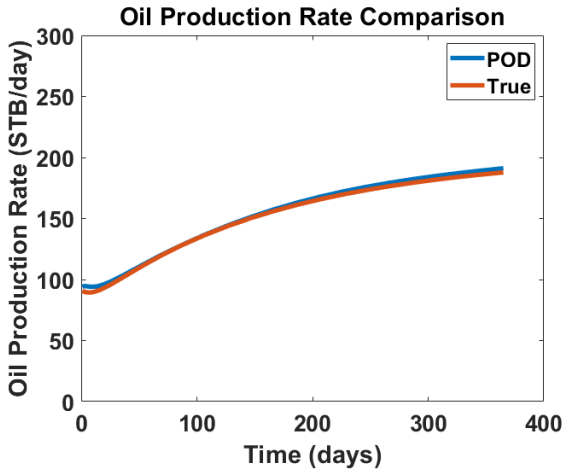


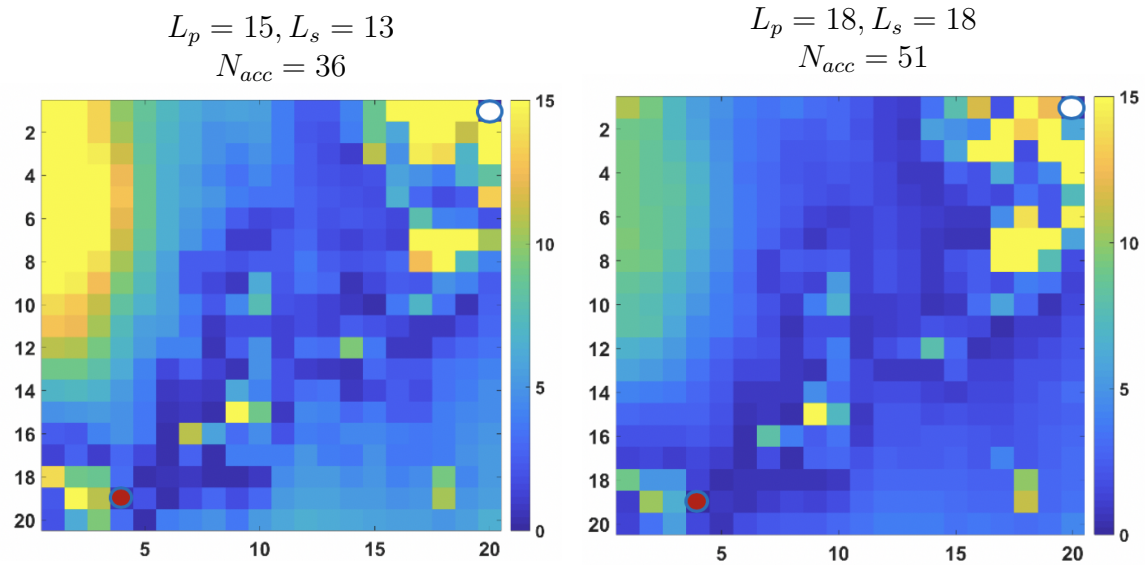
Figure 4.4: Oil production rate comparison at well location (7,16) between the true solution and that predicted by POD using a basis from well location (2,19)

Next, we analyze the effect of basis dimensions for a ROB. As we use the POD for PMOR, we project the pressure and saturation states to a subspace of much lower dimensions. So before we apply POD for analysis, we need to understand, a criteria to choose the dimensions of this lower dimensional subspace when we use it for a parameter space. For a fixed well configuration, we generally use, for an example, 99.99% rule to get the number of pressure and saturation basis. For these cases, we usually have more number of saturation basis as compared to pressure basis since saturation exhibits slower dynamics. However, when we use this basis for different parameters, this rule does not always provide a good subspace. For some new well locations, the saturation behavior might not change drastically as compared to that for the well location in the database of training parameters ($\mathbb{D}$). However, the pressure solution might be very different. So it is possible that a tail basis corresponding to low singular values for the basis well location might have a high importance for a new well location. It is also possible that a good subspace for a new parameter might require lower basis dimension than the basis of the well location in $\mathbb{D}$. So we keep these points in mind and explore a range of pressure and saturation dimensions of the basis. We still need a strong understanding on the behavior of the basis dimensions that can help us choose a good database of parameters.

From the Figure 4.5, we see that a change in the basis dimensions can significantly change the error map. This indicates that the basis dimensions need to be taken into account when determining the basis for a new parameter. There is no definite pattern that these error maps follow with change in the basis dimensions and thus this relationship is highly non-linear to understand and visualize.

These results show that a dictionary of local ROBs obtained from $\mathbb{D}$, can be used for the parameter space, and this can lead to projection to a much lower dimensional subspace during the online stage. Basis from snapshot concatenation can prove better for parameters that may not have a representative local ROB from $\mathbb{D}$. So, our aim will be to predict the error maps given a database of ROBs using the methodology as will be outlined in the next

section (4.3), to eventually select a pre-existing ROB for new well location.



(a) (b)

Figure 4.5: (a) and (b) shows the error map for different basis dimensions for the given well configuration

4.3 POD-based local PMOR problem formulation

A brief overview of the local PMOR techniques used in the literature was provided in Section 1, where majority of the methods were developed for LTI systems. They were based on construction of new ROBs or ROMs for a new parameter of interest [47, 46]. Some work was done for dynamic linear and non-linear PDEs was proposed in [48], that uses manifold learning techniques to interpolate the snapshots evaluated at different parameters.

In this work, we use a local ROB technique by analyzing some of its benefits compared to global ROBs (the main advantage being a much lower dimensional ROB for new parameter) for a new parameter of interest based on some case studies addressed in previous section (4.2). However, we also try to leverage some benefits of snapshot concatenation. As with the other local and global techniques, during the offline stage, we construct a database of parameters

($\mathbb{D}$) at which fine scale simulations are run. The ROB is constructed based on POD method for the application considered. The parameter ζ representing well location is defined in a high dimensional space as will be depicted in the next section (4.5). The methodology is based on the observations that a dictionary of preexisting ROBs (also includes a range of basis dimensions for each parameter) of the parameter database are sufficient to be robust in the parameter space. The selection of a preexisting ROB is based on prediction of the error maps for these basis. An error map is defined as the error in quantity of interest produced by a ROM for the entire parameter domain. We employ the methodology introduced in [52] for construction of the error maps. The idea is to use Machine Learning (ML) techniques to map the ROM errors for high dimensional parameters. During the online stage, the ML models will quickly predict the errors at a new parameter value to select the best preexisting ROB. Thus, this method relies on errors using a output quantity of interest (QoI) and does not necessarily ensure the physics of dynamics of flow to be captured correctly. For example, we will show later that, the preexisting ROB when used for a new well location does not project correctly that the new lowest pressure in the reservoir is well block pressure. This follows from the theory that was depicted by controllability study.

Thus, high dimensional regression ML models approximate the mapping:

$$[\zeta^*, \zeta_{\mathbb{D}}, L_X] \rightarrow \eta_{HF} \tag{4.2}$$

Here, $\zeta_{\mathbb{D}}$ is the parameter in $\mathbb{D}$ at which we obtain the ROB, ζ^* is the new parameter of interest not present in the database, L_X corresponds to the dimension of ROB for the state variable X and η_{HF} represents ROM error in quantity of interest produced by a ROB with dimension L_X, for parameter ζ^*. The benefit of this formulation is that it gives an idea about the error in QoI that should be expected given a ROB.

We also propose this mapping as a classification problem by setting a threshold in ROM error, for a qualitative response rather than a quantitative one. So the classification ML

models approximate the mapping:

$$[\zeta^*, \zeta_{\mathbb{D}}, L_X] \rightarrow C(\eta_{HF}) \tag{4.3}$$

$$C(\eta_{HF}) = \begin{cases} 1, & \text{if } \eta_{HF} < \eta_{th} \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

Here η_{th} is the maximum allowable error for the ROB to qualify as a good candidate basis for ζ^*.

4.4 Machine Learning Overview

In this section, we discuss about the machine learning techniques we use for comparison of their performance in accurately predicting these error maps.

4.4.1 Artificial Neural Network (ANN)

Artificial neural networks are non-linear statistical models that detect pattern in the data by discovering the input-output relationships, used for both regression and classification. The choice of ANN model is motivated by its capability to capture highly non-linear complex relationship between the input features and the output [99]. A feed-forward neural network used here consists of L layers with each layer consisting of predefined nodes and an input layer consisting of the features or independent variables used for prediction. Each of these layers have an associated transfer function and the nodes are connected to the nodes from previous layers by weights. A simple mathematical description is as follows:

$$Z^{l+1} = \Theta^{l+1} X^l \tag{4.5}$$

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

of the layer $l+1$ is given by:

$$X^{l+1} = G(Z^{l+1} + B^{l+1}) \tag{4.6}$$

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

The model is trained using backpropagation and optimization algorithms to adjust the weights Θ of the network that minimize the cost function. The detailed description on ANN and backpropagation algorithm can be found in [99]. Figure 4.6 shows a schematic of ANN. The neural network usually tend to overfit the data and thus, we use a regularized cost function for regression given by:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right] \tag{4.7}$$

And, for classification problem, the regularized cost function is:

$$J(\theta) \;=\; \frac{1}{2m}\left[\;-\;\sum_{i=1}^{m}\left(y^i\log\left(h_\theta(x^i)\right)\;+\;(1\;-\;y^i)\log\left(1\;-\;h_\theta(x^i)\right)\right)\;+\;\frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2\right] \tag{4.8}$$

Here, m represents total number of data examples, $h_\theta(x)$ represent the predicted output of the network and y is the expected output. θ represents each of the elements of the weight matrix Θ in each layer and λ is the regularization parameter to control feature importance and prevent overfitting. We use λ and number of units in each layer as the tuning parameters while training the ANN. The values of these hyperparameters are chosen here using K-fold

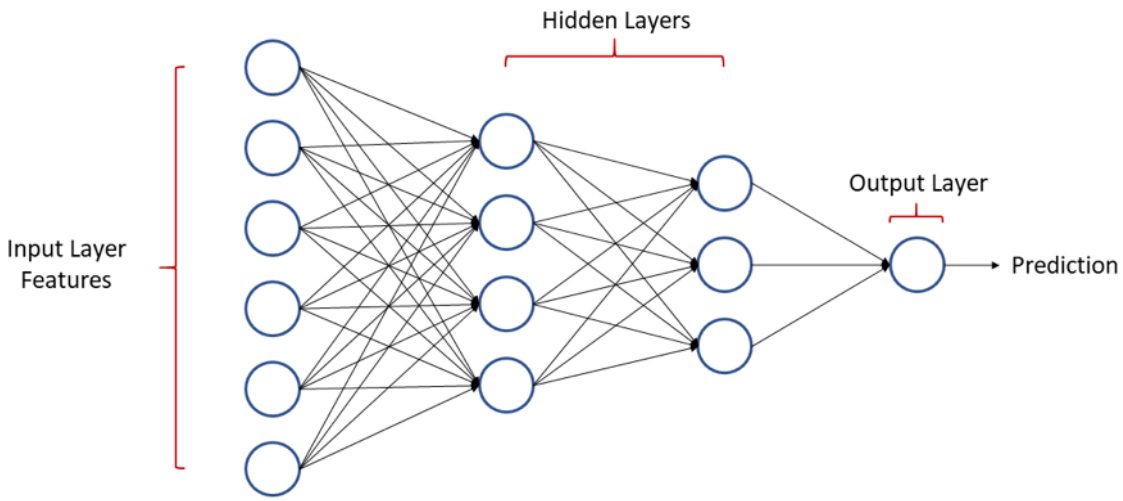


Figure 4.6: Schematic of a feedforward Artificial Neural Network

cross validation technique.

4.4.2 Random Forests (RF)

Random Forests (RF) is a supervised machine-learning technique that constructs an ensemble of decorrelated decision trees. It is based on the concept of bootstrap aggregating (bagging) to reduce the high variance from single decision trees and further using decorrelated trees to induce more randomness and variance reduction that eventually improves prediction performance for test cases. Like ANN, RF can be used for both regression and classification. We briefly describe the working of a RF here. The way a single decision tree is constructed is using a recursive binary splitting strategy. At each node of a tree, best split of a predictor variable is made that results in two branches and then successively splits the predictor space at each step. For the regression problem, the best split is the one that leads to the maximum possible reduction in error Residual Sum of Squares (RSS), and for the classification problem, Gini index or the cross-entropy are generally used to evaluate the best split. Bagging generates new training sets by sampling the data with replacement and then we generate an ensemble of decision trees for each of these training sets. A random forest further decorrelates the trees by selecting only a random subset of features for building

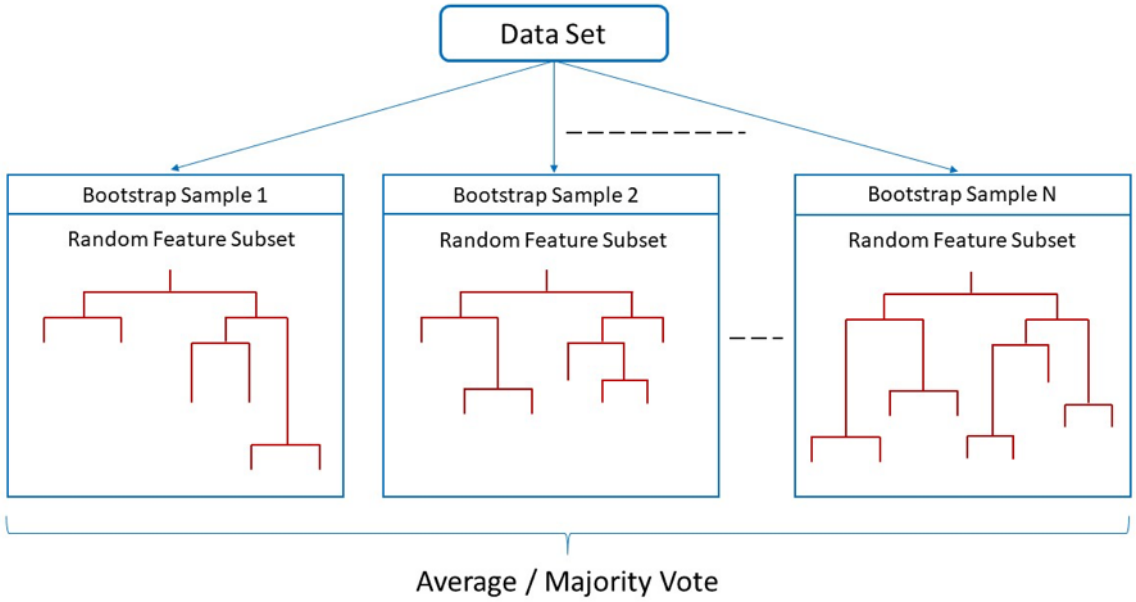


Figure 4.7: Description of Random Forests model where, each of the branches from sample dataset correspond to a bootstrap sample and a random feature subset is chosen for each branch to construct a decision tree

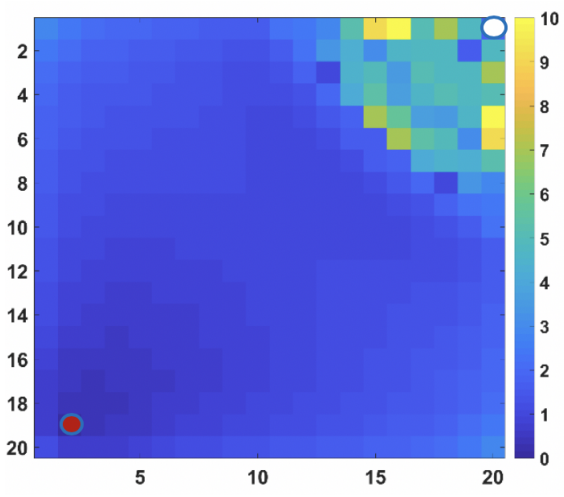
each tree. During prediction, for regression, the outcome is simply averaged over all the trees in the forest and for classification, the outcome is based on majority vote. For more details on Random Forests refer to [100]. Figure 4.7 shows the working of random forests pictorially.

4.5 Well Location Feature Determination

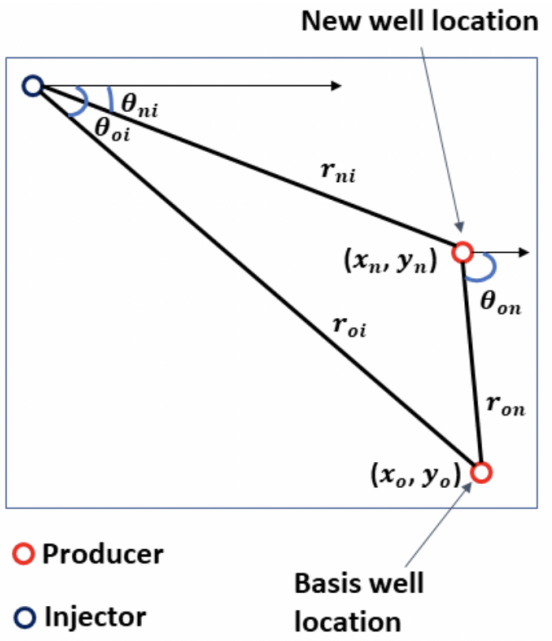
Before we employ the ML techniques to predict error maps, we need to quantify the features representing the parameter ζ. As we state before, the system parameter ζ for well location optimization corresponds to the index in the source/sink term vector. There is no explicit quantification of this parameter that represents changing well location in the PDE (Eq. 2.3). Thus, defining the input (parameter change) - output (ROM error) relationship is difficult. So, in this section, we first define geometric features for changing well locations based on a case study. Then, we introduce some physics based features using the concepts from flow diagnostics.

4.5.1 Geometric features

For the input features of prediction problems (equations (4.2) and (4.3)) one of the obvious choices of changing well locations is the coordinates of wells in the database and the new wells. We define more input features based on the analysis from a simple case of a homogeneous reservoir.



(a)



(b)

Figure 4.8: (a) shows the error map for a homogeneous reservoir. (b) shows the feature parameters chosen on the basis of results from (a)

Consider a homogeneous reservoir with a 400 (20×20) grid cells and permeability of 10mD. We run fine scale simulation for a well $\mathbb{D}$, referred to as a basis well, at location $(2, 19)$ to get ROB from the snapshot matrix and then, use this basis to get ROMs for all the production well locations. The error map is shown in Figure 4.8a. As we can observe, the error map shows symmetric pattern in error distribution. As we move radially away from the original well, we see a continuous increase in the oil production rate error. This

motivates us to use the distance and orientation information of the wells for defining $\zeta_{\mathbb{D}}$ and ζ^*. Again, we note that the top right corner of high error region refers to the water flooded area. We show a pictorial representation of all the features that we deem relevant to information about the changing well locations in Figure 4.8b. We also note that we need to add information about the basis dimensions to error prediction and thus, by adding the pressure and saturation basis dimensions as one of the input features, we aim to model the complex non-linear relation between basis dimensions and quantity of interest errors. Then, we also aim to add the information from snapshot concatenation to predict the error maps from corresponding basis by adding features of all the parameters used in concatenation. Table 4.1 shows all the geometric features used for error map prediction for basis from a single well. Table 4.2 shows all the geometric features in addition to those in Table 4.1 used for error map prediction for basis from concatenated solutions of 2 well locations.

Feature Set for local basis (single well)
x_o - X coordinate of the basis well
y_o – Y coordinate of the basis well
x_n – X coordinate of the new well
y_n – Y coordinate of the new well
r_{oi} – distance between basis well and injector
r_{ni} – distance between new well and injector
θ_{oi} – angle between basis well and injector
θ_{ni} – angle between new well and injector
r_{on} – distance between basis well and new well
θ_{on} – angle between basis well and new well
L_p – Pressure basis dimension
L_s – Saturation basis dimension

Table 4.1: Geometric features corresponding to the basis well (a single well) and new wells for error map prediction. Features corresponding to basis wells correspond to $\zeta_{\mathbb{D}}$ and those for new wells correspond to ζ^* in equations (4.2) and (4.3)

Feature Set for local basis (two wells)
x_{o2} – X coordinate of the basis well 2
y_{o2} – Y coordinate of the basis well 2
r_{oi2} – distance between basis well 2 and injector
θ_{oi2} – angle between basis well 2 and injector
r_{on2} – distance between basis well 2 and new well
θ_{on2} – angle between basis well 2 and new well
r_o – distance between basis well 1 and basis well 2
θ_o – angle between basis well 1 and basis well 2

Table 4.2: Geometric features corresponding to the basis wells (concatenate solutions from 2 wells) and new wells for error map prediction. Add these to features in Table 4.1, for basis with snapshot concatenation

4.5.2 Physics based features

As we have discussed the geometric features that correspond to changing well locations and deem important for relation to the error maps estimation, we now try to define the physics based features that are of major importance as they describe the physics of flow and their relation to the well locations.

The geometric features were defined based on the observations for a homogeneous reservoir. So when we consider a heterogeneous reservoir case, one obvious choice of physics based feature that need to be included is the permeability change in the well gridblock when the location of well changes. Thus, ΔK which defines the difference in permeability between the basis well gridblock and that of the new well gridblock is one important physics based feature we consider.

What is also of major importance for us to understand the change in physics of flow with changing well location is some representation of well connectivity and flow heterogeneity between well pairs. In order to understand such properties, we now introduce the concept of flow diagnostics and various properties that are calculated during flow diagnostics that serve as important features for our problem.

Flow Diagnostics

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

The reason behind this computational advantage is because the flow diagnostic tools provide quantitative information based on the steady-state flow. Flow diagnostic tools have been developed using streamline simulation techniques [101, 102] that has shown to be very effective for various applications, but have some limitations in terms of computational complexity and extensibility. Most of the existing commercial simulators are based on finite volume methods capable of simulating fluid flow with different spatial grid geometry because of their mass conservative nature. Keeping this in mind, the flow diagnostic properties were obtained using standard finite-volume discretization in [103, 104]. We use the flow diagnostics module of MRST (Matlab Reservoir Simulation Toolbox) [104] to evaluate these properties with the aim to aid in accurate construction of reduced order models for changing well locations. We now briefly introduce the governing equations that lead to quantitative measures of flow dynamics as explained in the works [103, 104].

Governing Equations

A single phase incompressible flow is considered to compute the representative flow field,

$$\nabla \cdot \mathbf{v} = 0 \tag{4.9a}$$

$$\mathbf{v} = -\frac{K}{\mu}\nabla p \tag{4.9b}$$

where, K is the permeability tensor, μ is the viscosity of fluid, $\mathbf{v}$ is the Darcy velocity and p is the phase pressure. Here, for the purpose of flow diagnostics, no flow boundary conditions are assumed and μ is set to 1.

Stationary Tracer Distribution and Time of Flight

To observe the transport properties of the flow field, we consider the equation (4.10) describing the neutral tracer transport which is injected into the injection well with a concentration c. For simplicity we neglect source/sink terms and assume zero concentration of tracer initially in the entire domain. The transport equation is written as:

$$\phi \frac{\partial}{\partial t} c + \mathbf{v} \cdot \nabla c = 0 \tag{4.10}$$

where, ϕ is the porosity of reservoir. At late times, the equation (4.10) takes the steady state form:

$$\mathbf{v} \cdot \nabla c = 0 \tag{4.11}$$

The stationary tracer equation gives an idea about the regions in the reservoir that are influenced by injectors and producers and ultimately help us understand swept volumes, well allocation factors etc. However, it does not give enough information about the impact of flow on the heterogeneity of the reservoir as dictated by permeability and porosity distribution. Thus, instead of considering the tracer distribution, we consider time of flight (TOF) coordinate $\tau(x)$ which indicates the time it takes for a tracer to travel from the nearest injector to a given point x in the reservoir. TOF is defined as:

$$\tau(x) = \int_{\psi(s)} \frac{\phi}{|\mathbf{v}(x)|} ds \tag{4.12}$$

where ψ and s denote the streamline and the arc length measured along the streamline, respectively. Operator identity shown in equation (4.13) is used to represent the 1D flow

along streamlines for the equation (4.10) [105].

$$\mathbf{v} \cdot \nabla = |\mathbf{v}| \cdot \frac{\partial}{\partial s} \tag{4.13}$$

Similarly, we can derive the TOF equation in Eulerian coordinates using this identity:

$$\mathbf{v} \cdot \nabla \tau = \phi \tag{4.14}$$

These equations (4.11 and 4.14) have a hyperbolic form. They can be written in a conservative form as:

$$\nabla \cdot (\mathbf{v}u) = b \tag{4.15}$$

where, $u = TOF$ or tracer concentration and b represents the source or sink. This form is used since it has a natural finite volume discretization that can be used to solve for TOF and c and is a generalization to the case where $\nabla \cdot \mathbf{v} \neq 0$.

For the cases considered here for my research, we only have one producer and one injector and hence we do not consider the tracer distribution to identify quantities like well allocation factors, well-pair connections etc. However, these should serve as important features for multiple injectors and producers. Thus, we only consider TOF for our cases. Two quantities are calculated called the forward and backward time of flight using equation (4.14) as:

$$\mathbf{v} \cdot \nabla \tau_f = \phi \tag{4.16a}$$

$$-\mathbf{v} \cdot \nabla \tau_b = \phi \tag{4.16b}$$

The forward TOF τ_f is the time required for a tracer to reach at a given point in the reservoir after injected at an inflow boundary (injector in our case). The backward TOF τ_b is obtained by reversing the flux field which indicates the time required for a tracer released at a given location within the reservoir to reach a producer/outflow boundary. The sum of both these

quantities gives the total time of flight also defined as the residence time of the particle.

$$\tau_f + \tau_b = \tau \tag{4.17}$$

Lorenz Coefficient

Time of flight and tracer concentration can also be used to assess displacement heterogeneity of the reservoir using a quantity called Lorenz coefficient. The heterogeneity is represented in terms of flow capacity - storage capacity ($F - \Phi$) diagrams. (Note that for the sake of consistency in notations with the literature, for this section we refer to Φ as the storage capacity and not to be confused with POD basis). This is equivalent to plotting fractional flow versus saturation for a 1D flow displacement. As defined in [106], the storage capacity Φ is defined as the cumulative pore volume as fraction of total travel time τ, i.e., $\Phi(\tau) = \int_0^\tau \phi(x(\tau))d\tau$. The flow capacity is defined as the cumulative flux for increasing travel time. For an incompressible flow, $F(\tau) = \int_0^\tau \frac{\phi(x(\tau))}{\tau} d\tau$, since the pore volume equals the product of the flux and the total travel volume. Lorenz coefficient which is the measure of displacement heterogeneity is defined by measuring how much the flow capacity deviates from the ideal piston like displacement and hence can be written as:

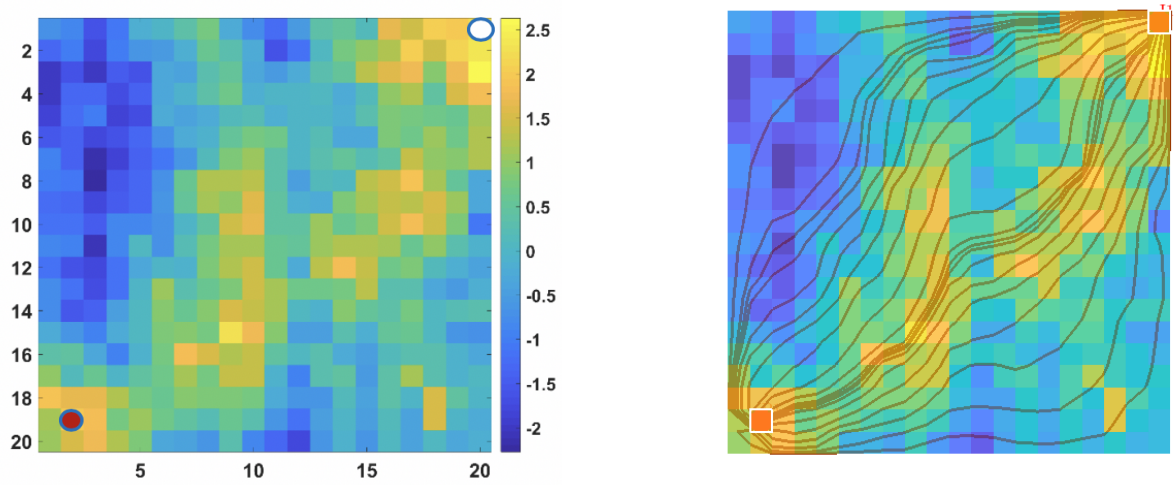
$$L_c = 2\int_0^1 (F(\Phi) - \Phi)d\Phi \tag{4.18}$$

which is twice the area under the $F - \Phi$ curve and above the line $F = \Phi$. Thus, it is 0 which corresponds to homogeneous displacement and unity for infinitely heterogeneous displacement. For a detailed explanation on calculating F and Φ, the reader is referred to [106] for streamline based calculations and [103] for finite volume based diagnostics.

Flow diagnostic features - Example

We now show an example of the flow diagnostic features for one of the well configurations. Consider the same heterogeneous permeability case as was shown in (4.3a) with wells at

location $(2, 19)$. We use the MRST flow diagnostics module that computes TOF and Lorenz coefficient based on solving equations as shown above (4.16 - 4.18).



(a) Normalized Permeability Field

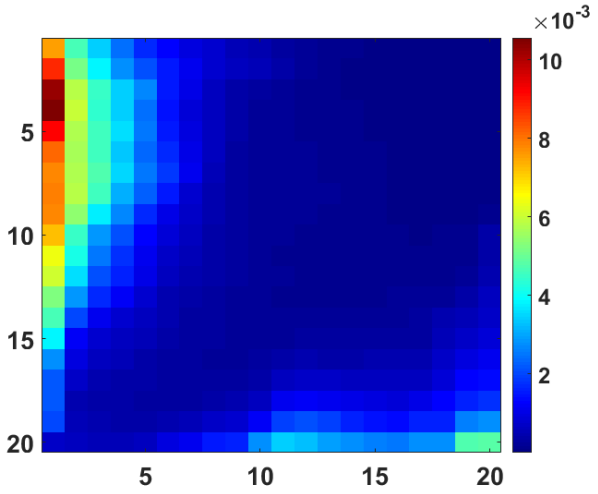
(b) Streamlines

Figure 4.9: (a) Heterogeneous channelized permeability field with one producer in red and one injector in white (b) Streamlines imposed on the permeability field

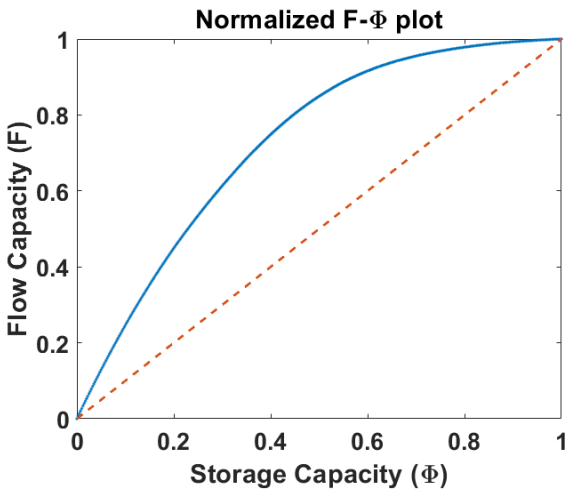
It can be seen that the low perm area in the reservoir that is not well connected to the injector has a higher travel time and the producer location as shown above has a very low travel time as it is located on a high permeability gridblock and connected to the injector via a channel. Thus, while considering the feature to be added for ML procedure, we use the TOF at the producer well location corresponding to the basis well and the new well. Note that since we are interested in finding time of flight at producer well location, it is sufficient to compute forward time of flight since backward time of flight will be negligible.

4.5.3 Feature Selection

The important step before training a ML model is to remove any redundant and correlated features. For this procedure we use the Wrapper approach, the procedure of which is briefly



(a) Total travel time



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

Figure 4.10: (a) The total travel time of the particle injected at injector to any point in the reservoir (b) $F - \Phi$ plot to compute Lorenz coefficient as the area under the solid $F - \Phi$ curve and the dashed line $F = \Phi$

Physics based feature set for local basis (single well)
ΔK – Permeability difference between basis well and new well locations
TOF_o - Total time of flight at basis well location
TOF_n - Total time of flight at new well location
LC_o - Lorenz coefficient for basis well location
LC_n - Lorenz coefficient at new well location

Table 4.3: Physics based features corresponding to the basis well (a single well) and new wells for error map prediction. Features corresponding to basis wells correspond to $\zeta_{\mathbb{D}}$ and those for new wells correspond to ζ^* in equations (4.2) and (4.3)

outlined below. The wrapper approach sequentially selects subset of feature space and evaluate its performance with respect to an induction algorithm (black box model) using cross-validation and subsequently adds or deletes features based on the search criteria. The search engine we use is Best First Search (BFS) using Forward selection which is more robust than the greedy hill-climbing search. Additional details about the wrapper approach can be found in [107]. The procedure is explained in the Figure 4.11. This step becomes

computationally expensive as the number of features grows. But, what we observe with the feature set that we will propose for the well locations is that, the BFS methodology sometimes selects all the features, depending on the data set and the black-box model used. This suggests than no features in this case are correlated. So while training the models, we use all the features and then for feature importance, we try to tune better the hyperparameters and the regularization parameters instead of spending computational effort using the wrapper approach.

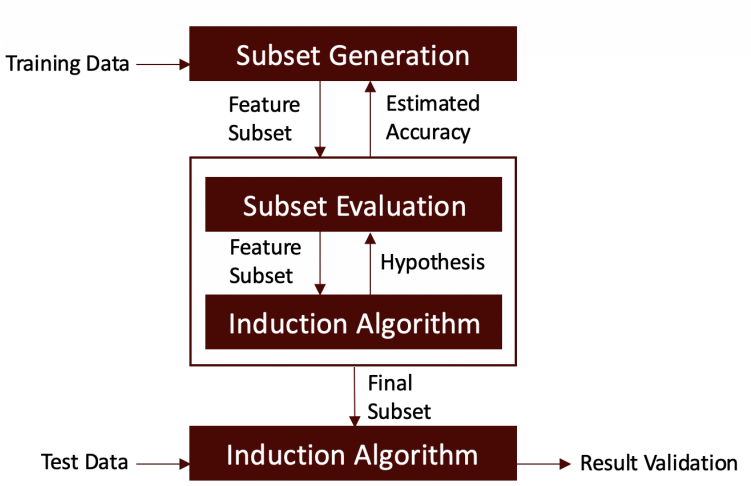


Figure 4.11: Wrapper approach

4.6 Case Study

Based on the observations and feature selection described above, we now apply the ML techniques to predict the error maps for different cases. We use two different strategies for error prediction as mentioned in the section (4.3). First is, posing the problem as the standard regression problem for quantitatively predicting the ROM error values at each well location. The second strategy is posing the problem as a classification problem for qualitative error description. Depending on the application, it may sometimes not be necessary to predict exact value of errors for different parameters (especially no need to predict exact values of

large error). So, for the examples below, we also show the classification results, where the threshold error is chosen as 5%. Thus, the classification models will predict the locations where local ROBs, corresponding to well locations in $\mathbb{D}$, will produce less than 5% error in the oil production rates.

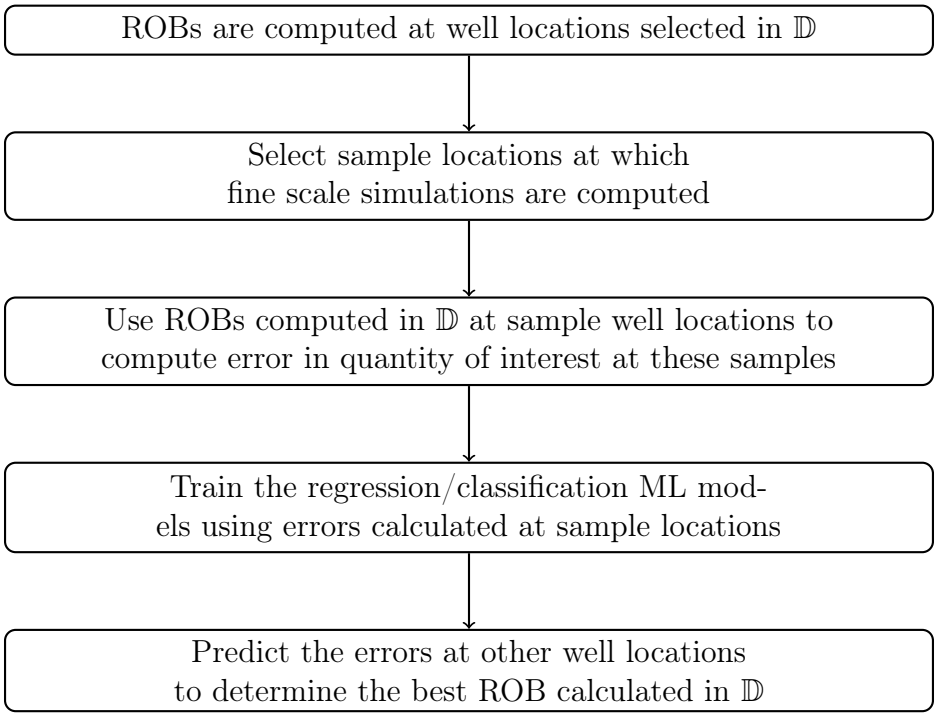


Figure 4.12: Workflow of the ML based local PMOR procedure

The workflow of the process is outlined in Figure 4.12. During the well location optimization procedure, for a new well location at each iteration, error is predicted using the trained ML model for all pre-existing ROBs and the best ROB for this well location is selected. This prediction process and finding the ROB with minimum error is computationally very cheap.

4.6.1 Case 1: Error maps of basis obtained by single well location

Reservoir model 1

Here, we show the results of predicting the error maps for different basis. The reservoir model is the same 20×20 grid block channelized permeability field as before (Figure 4.13). The relative permeability is defined by Corey-type curves with exponent 2 for both the fluids. The flow is considered incompressible for all the cases shown. Thus, we can use the MRST flow diagnostics module for TOF and Lorenz coefficient calculation which is developed for incompressible fluids. The permeability field is isotropic and the porosity is set to 0.2. The initial reservoir pressure is 4200 psi and the injector and producer are BHP controlled at a constant pressures of 7000 psi and 2500 psi respectively. The irreducible water saturation is 0. For all the well locations we consider, the BHP control profile is set the same for each. We note that we are just looking at the single parameter i.e. producer well location. But the workflow can be extended for Closed Loop Field Development where we need both BHP or rate control and well location optimization, by including the controls as additional parameter to the feature space.

Well Location	(L_p, L_s)
(4,19)	(1,4)
(2,19)	(2,8)
(18,19)	(1,6)
(20,20)	(1,4)

Table 4.4: Dictionary of basis used for prediction

The well location positions in $\mathbb{D}$, to get the dictionary of ROBs are set to (4,19), (2,19), (18,19), (20,20), that includes a range of different permeabilities. The sets of basis dimensions for each well are shown in Table 4.4. Thus, we have a dictionary of 4 ROBs to choose from for other well locations. The dimension pairs are chosen by 99.9% energy criteria of the singular

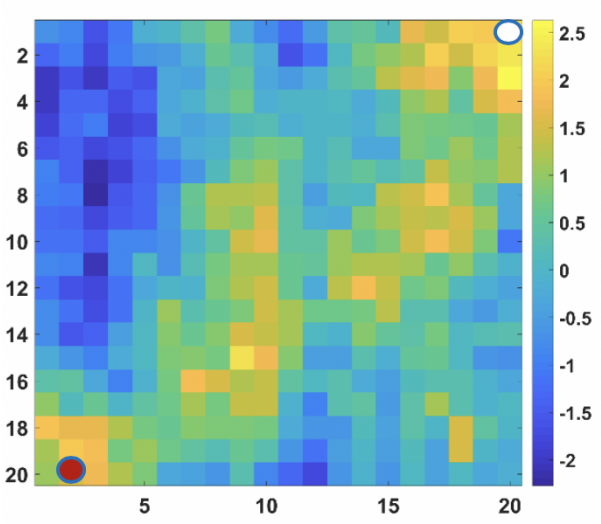
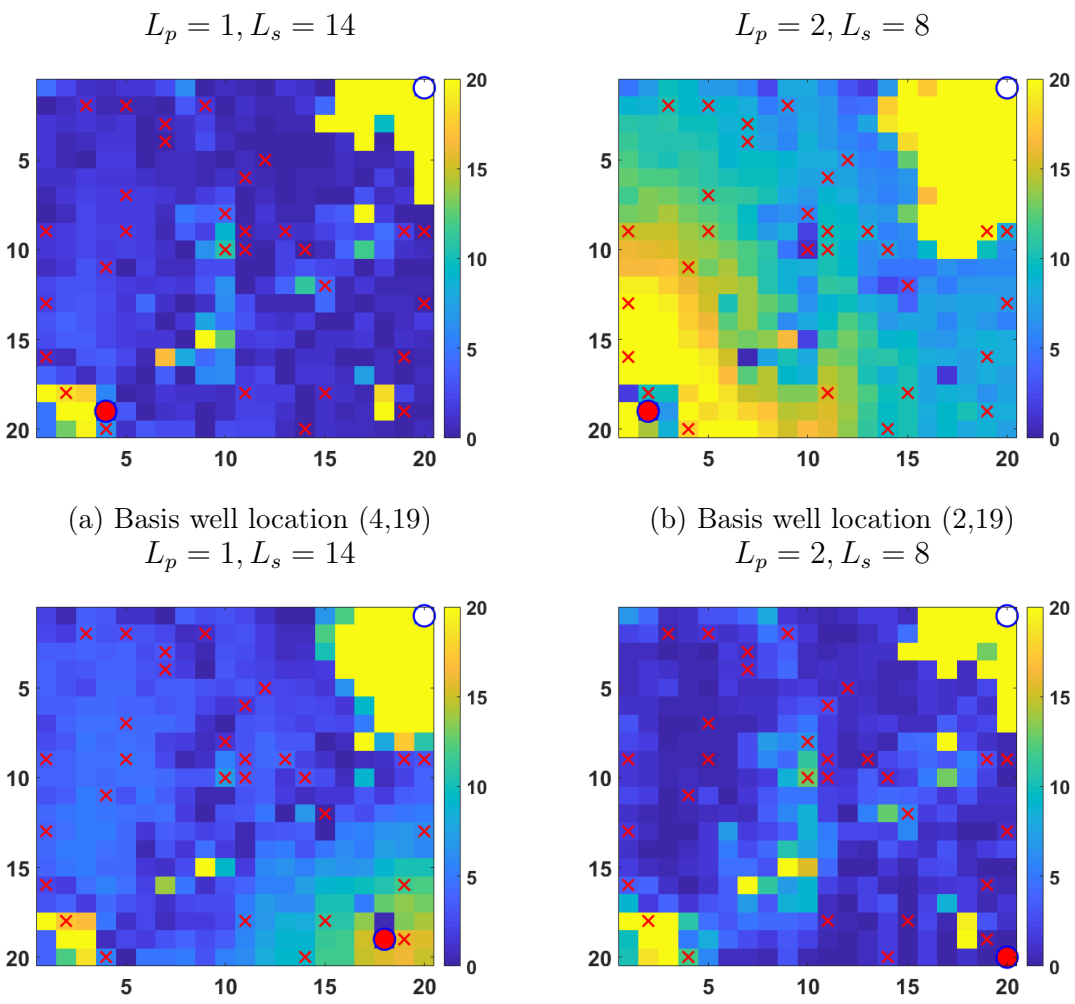


Figure 4.13: Normal scale k_{norm} of permeability field. The values of permeability are chosen as $5^{k_{norm}} mD$ to exhibit low contrast. Red indicates the producer well and white is the injector. The producer well changes location at any location in the reservoir

values. For an incompressible flow, we did not observe any significant changes in the error maps for different set of basis dimensions. Hence we just consider one set of pressure and saturation basis for each well location. Thus, for this case, the basis dimension will not be an input feature for ML models. The first step of our workflow is to generate a sample dataset from the error maps. We randomly choose 30 samples from each of the 4 cases. However, the sample locations in each case are the same. By doing so, we just need 30 fine scale simulations at each sample point (run in parallel on a 30 node high performance computer and hence just a single run required), corresponding to producer wells at these locations. Then, for each sample, we run the reduced model at these locations corresponding to 4 pre-existing ROBs. Figure 4.14 shows these samples on true error maps for regression ML model and Figure 4.15 shows samples on 2 of the true error maps for classification ML models.

The next step is feature selection to remove any redundant and correlated features. But as mentioned before, for many cases, all the features were selected by the BFS algorithm. Thus, we include all the features while training the model and spend more efforts on tuning



(a) Basis well location (4,19)

(b) Basis well location (2,19)

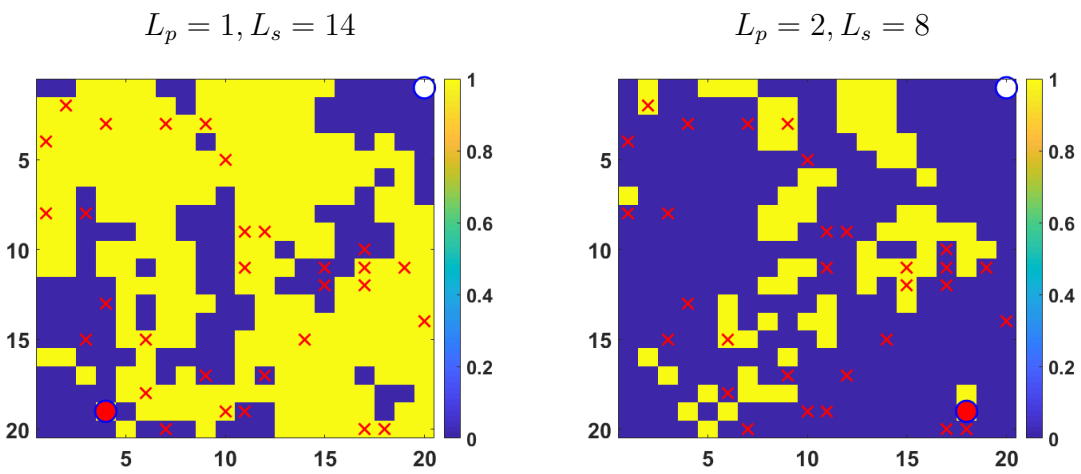
(c) Basis well location (1,6)

(d) Basis well location (1,4)

Figure 4.14: Samples collected from 4 error maps as shown by red $\times$ for training the regression ML models

the regularization parameter for feature importance. We use 5-fold cross validation on the data set to tune the regularization parameters and hyperparameters.

The last step is to train the model. We use both ANN and RF to compare the results for each algorithm. For this case example, we have 15 features input to the ML models as listed in Tables 4.2 and 4.3. The tuning parameters we choose for ANN are, number of nodes in the hidden layers (N_{nodes}) and the regularization parameter λ. For RF, we use number of features to search for split (N_{fmax}) and minimum number of samples at each leaf node (N_l)



(a) Basis well location (4,19)

(b) Basis well location (2,19)

Figure 4.15: Samples collected from 2 of the 4 error maps as shown by red $\times$ for training the classification ML models

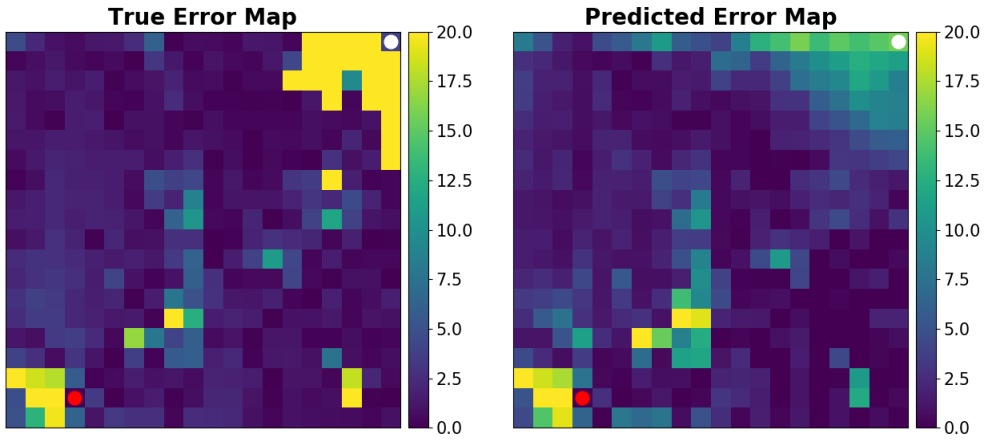
as the tuning parameters. ANN used here has 4 hidden layers which is fixed. We show the predicted results of ANN and RF using both regression and classification strategies.

	Regression	**Classification**
ANN	N_{nodes}=14, λ=0.04	N_{nodes}=18, λ=0.8
RF	N_{fmax}=8, N_l=2	N_{fmax}=8, N_l=2

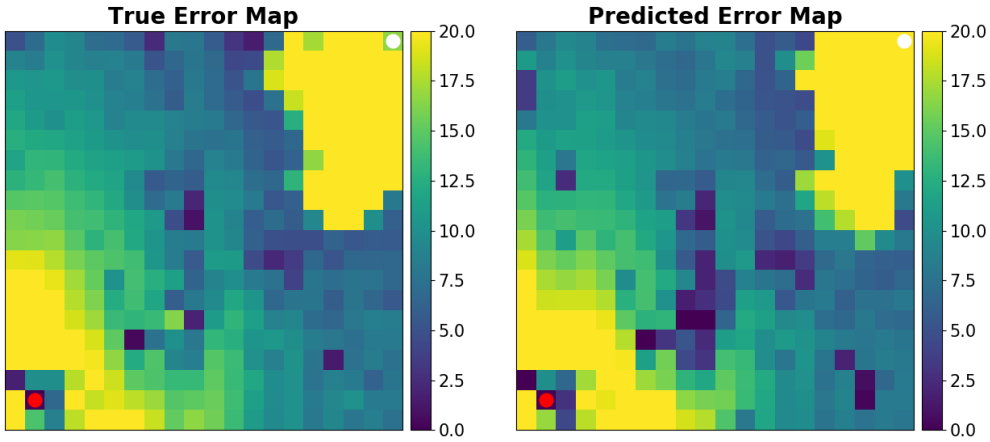
Table 4.5: Regularization parameters and hyperparameters chosen by 5-fold Cross Validation

Table 4.5 shows the optimal regularization parameters selected for the models. We expect the regression accuracy to be low because of the larger discrepancies at high error points. But, we are more interested in finding the locations with low errors. Thus, we also use Reg-Class accuracy measure shown in Table 4.6, which is obtained by converting the regression predicted map to a binary map (0 - 1) based on the 2% threshold and compare it with the true binary error map, to get an idea about the prediction of low error locations.

Figure 4.16 and Figure 4.17 shows the quantitative error maps predicted by ANN and



(a) Error map comparison using ANN for basis well at $(4, 19)$

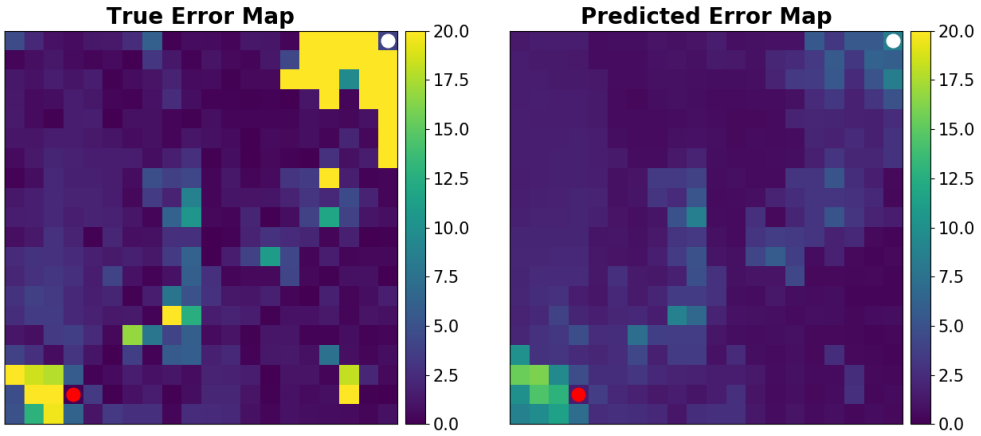


(b) Error map comparison using ANN for basis well at $(2, 19)$

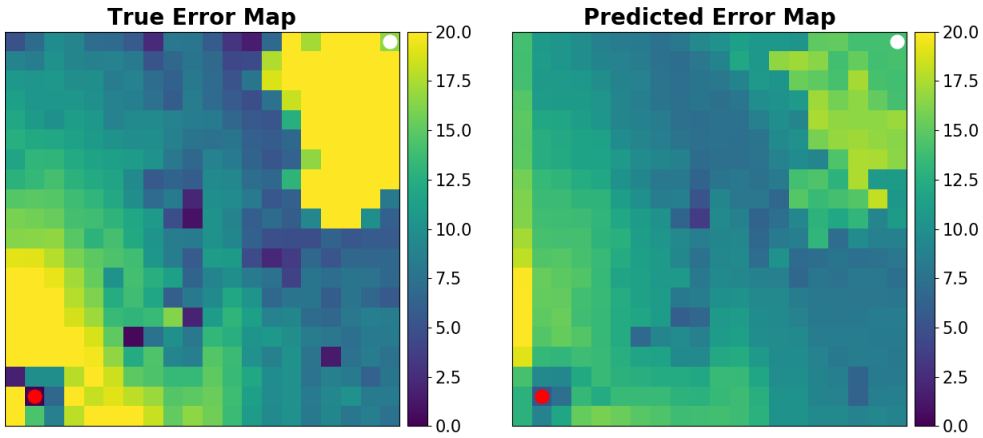
Figure 4.16: ANN regression prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) and (b) $L_p = 2, L_s = 8$ at (2,19) gridblocks

RF respectively for 2 of the 4 ROBs in dictionary given the sample distribution as shown in Figure 4.12.

Figures 4.18 and 4.19 show the qualitative error maps predicted by ANN and RF classification models for 2 out of the 4 basis respectively given the sample distribution as shown in Figure 4.13. To clarify the idea of classification map, we show an example in Figure 4.20. It shows POD predicted oil production rate at well location (17,14) which is classified as 1 (shown by yellow block in Figure 4.18) by the basis well location at (4,19) and the true oil production from the fine scale simulation. Thus, if the well is moved to location (17,14), we



(a) Error map comparison using RF for basis well at $(4, 19)$



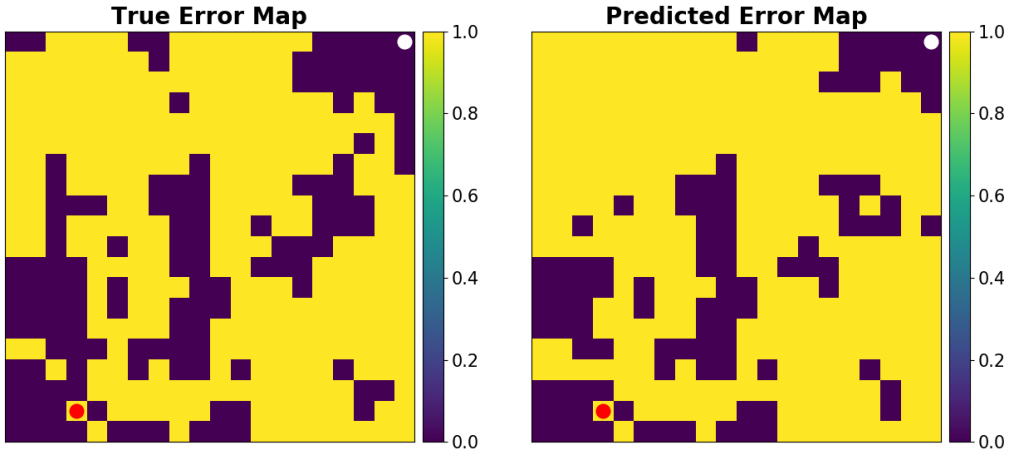
(b) Error map comparison using RF for basis well at $(2, 19)$

Figure 4.17: RF regression prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) and (b) $L_p = 2, L_s = 8$ at (2,19) gridblocks

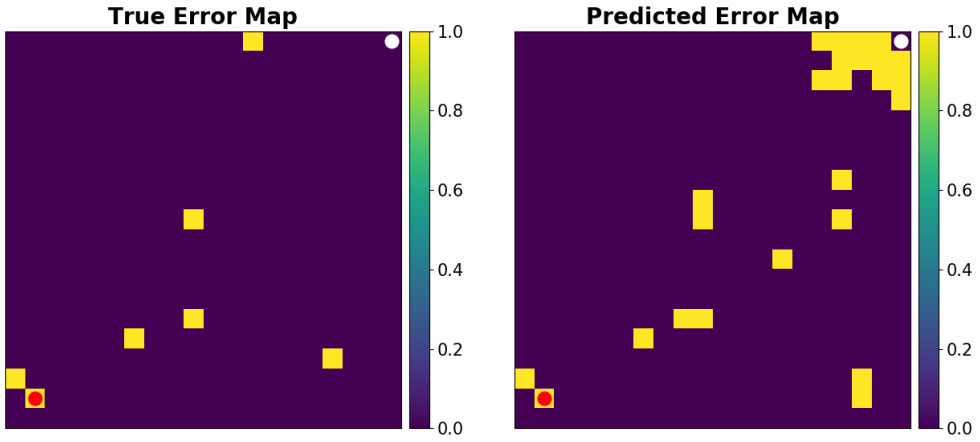
can use the POD basis from well at (4,19) to get less than 2% error in the oil production rate as shown in Figure 4.20.

Table 4.6 and 4.7 shows the accuracy of prediction for all the 4 error maps using ANN and RF for regression and classification problems respectively for the sample distribution as shown. For regression, it is the norm relative accuracy and for classification, it is the percentage of locations where error is correctly classified.

This was the result considering one of the random sample distributions. We performed the same test for 10 different random distributions and observed similar results. Figures

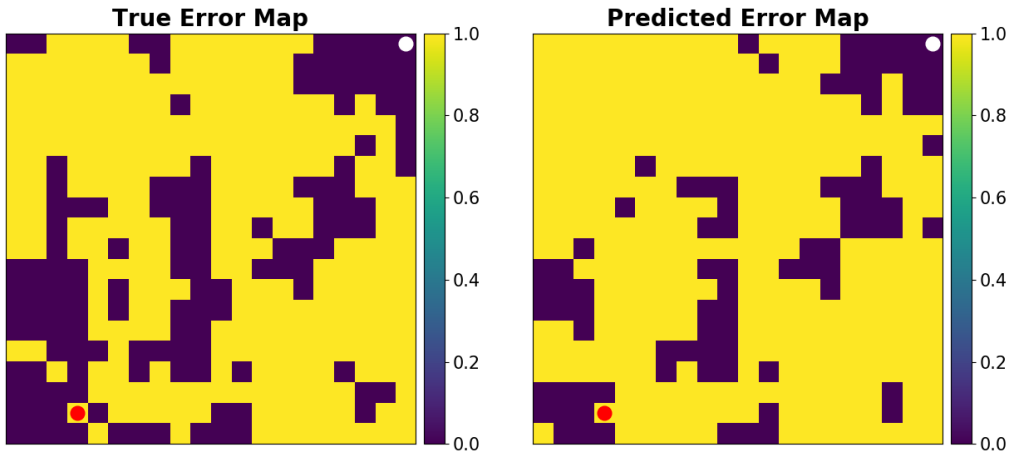


(a) Classification error map comparison using ANN for basis well at $(4, 19)$

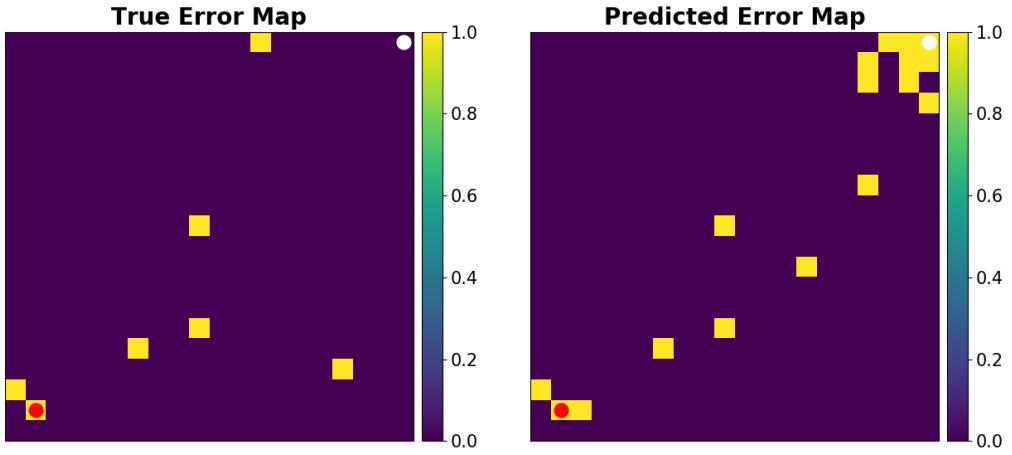


(b) Classification error map comparison using ANN for basis well at $(2, 19)$

Figure 4.18: ANN classification prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) gridblock and (b) $L_p = 2, L_s = 8$ for basis well at (2,19) gridblock



(a) Classification error map comparison using RF for basis well at $(4, 19)$



(b) Classification error map comparison using RF for basis well at $(2, 19)$

Figure 4.19: RF classification prediction for an error map corresponding to (a) $L_p = 1, L_s = 4$ for basis well at (4,19) gridblock and (b) $L_p = 2, L_s = 8$ for basis well at (2,19) gridblock

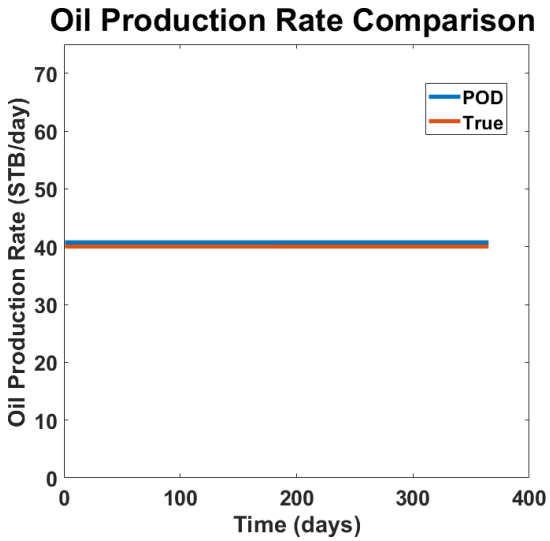


Figure 4.20: Oil production rate comparison at well location (17,14) between the true solution and that predicted by POD using a basis from well location (4,19)

	Regression		Reg- Class	
	Train	Test	Train	Test
ANN	99.9	88.1	99.9	92
RF	94.7	72.9	98.9	88.9

Table 4.6: Error map prediction accuracy for ANN and RF regression problem

4.21 and 4.22 shows the accuracy box plot of regression and classification problems for these samples respectively.

As we can see from the results, posing the prediction problem as a classification problem gets a better accuracy in determining an appropriate basis for new well location. For problems like well placement optimization, we believe, using a small threshold error, and thus posing the formulation as a classification problem, for selecting a ROB from the database should be good enough rather than very accurate reduced models.

ANN proved better than RF for all the other cases of sample distributions for regression problem and proved better for most of the cases for classification problem by looking at the median for both kind of models in the box plots.

	Classification	
	Train	Test
ANN	99.9	92.93
RF	100	90.69

Table 4.7: Error map prediction accuracy for ANN and RF classification problem

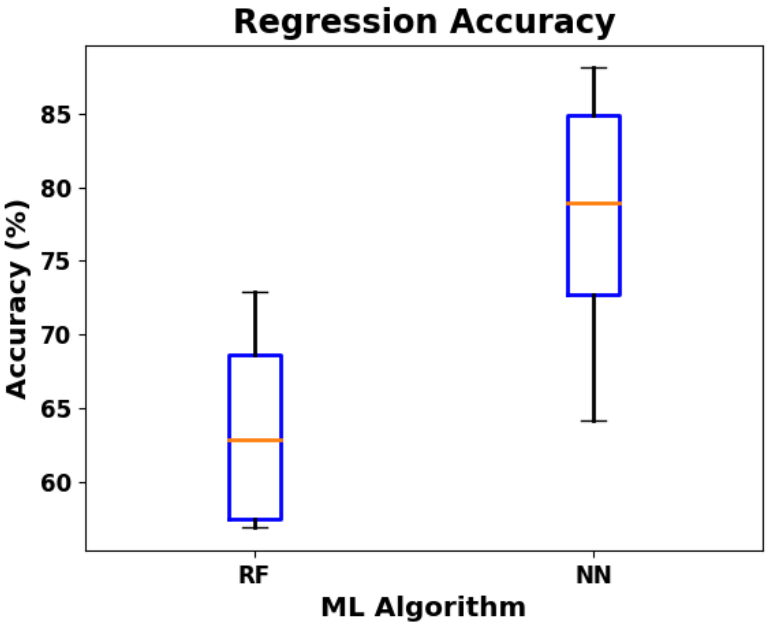


Figure 4.21: Box plot of accuracy for regression results using RF and NN on 10 sample datasets

Reservoir model 2

We now apply the same procedure on a bigger, more heterogeneous reservoir. The reservoir model is a 51×51 grid block heterogeneous permeability field as shown in Figure 4.23 with two phase oil-water flow. The relative permeability is defined by Corey-type curves with exponent 2 for both the fluids. The flow is considered incompressible and we neglect the effects of gravity and capillary pressure. Again, for all the well locations we consider, the BHP control profile is set the same for each.

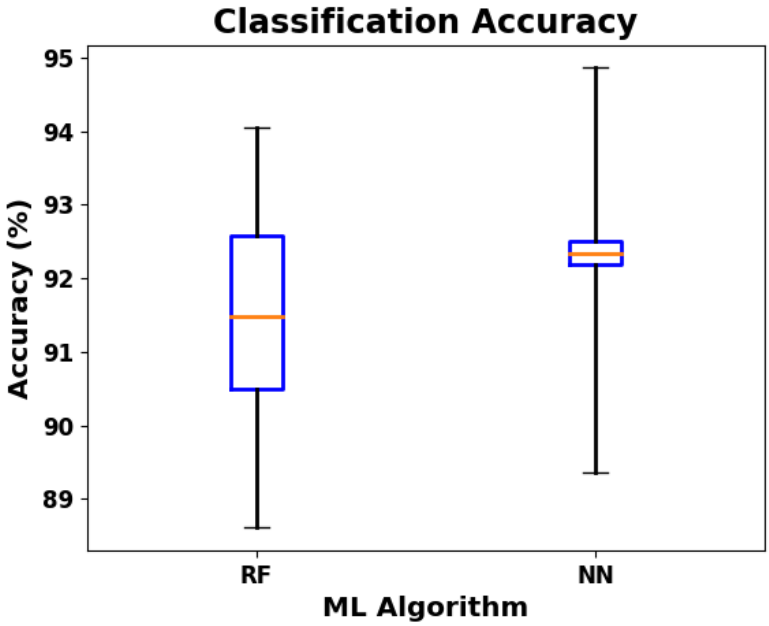


Figure 4.22: Box plot of accuracy for classification results using RF and NN on 10 sample datasets

We now use the same procedure as above to predict the error maps associated with each basis in the database. The database of well locations and the corresponding basis dimensions used to construct the error maps is shown in the Table 4.8. The basis dimensions here are chosen based on 99.99% energy criteria of singular values. The database of basis well locations chosen represent the permeability range from low values to high, since permeability is found to be the most important feature that has a strong correlation to the error in QoI. We randomly choose the basis wells in different regions of the reservoir. From the results in previous case study, we show this case only for classification problem which is expected to show much better accuracy. Here, the threshold of error is chosen to be 10% and hence all the gridblocks marked by yellow color are the low error cells for the corresponding basis. Figure 4.24 shows samples locations selected to train the ML models for predicting the classification error maps.

Both ANN and RF models are trained to predict the error maps and compare their performances. The hyperparameters associated with the trained models are shown in Table

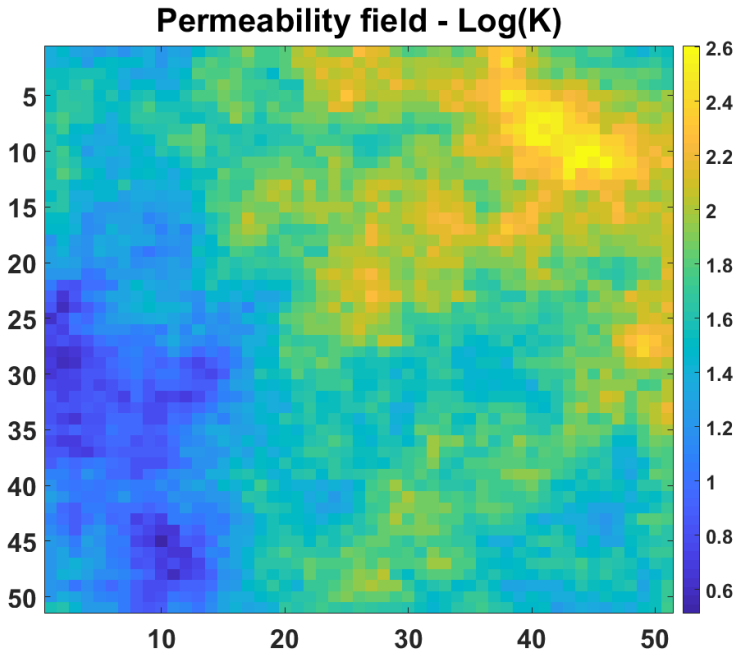
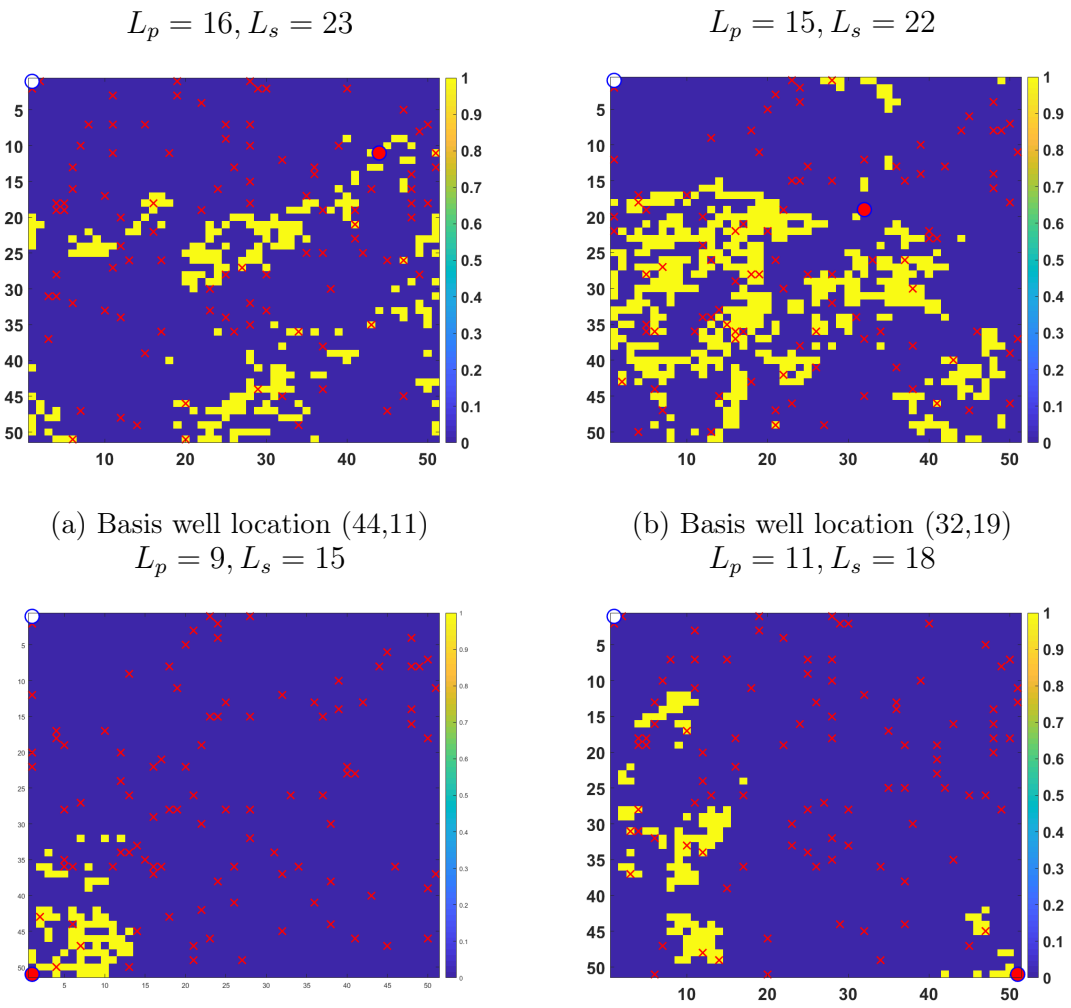


Figure 4.23: Heterogeneous permeability field in log scale

Well Location	(L_p, L_s)
(44,11)	(16,23)
(32,19)	(15,22)
(1,51)	(9,15)
(51,51)	(11,18)

Table 4.8: Dictionary of basis used for prediction

4.9. Figures 4.25 and 4.26 are the error maps as predicted by ANN and RF models for two of the basis wells. From the accuracy and the error maps predicted, it is clear that artificial neural network does a much better job in identifying the low error regions as compared to the random forest model which is the same observation as previous case.



(a) Basis well location (44,11)
$L_p = 9, L_s = 15$

(b) Basis well location (32,19)
$L_p = 11, L_s = 18$

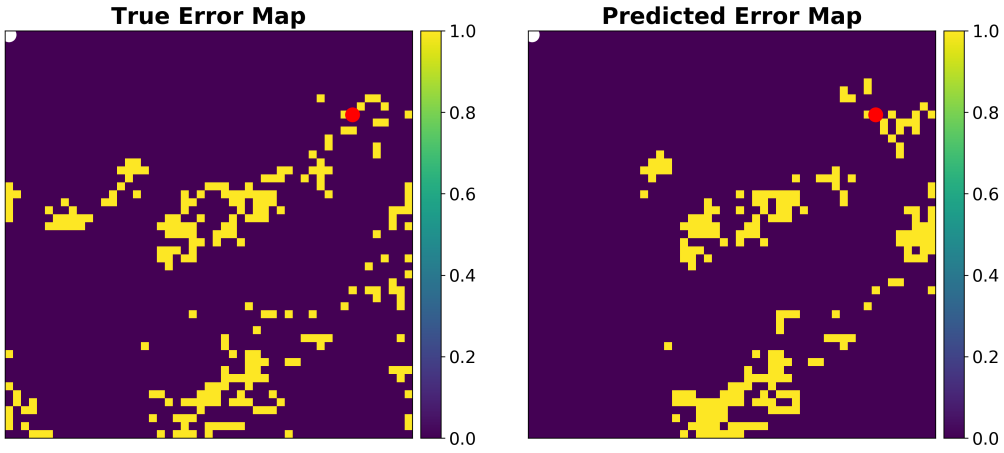
(c) Basis well location (1,51)

(d) Basis well location (51,51)

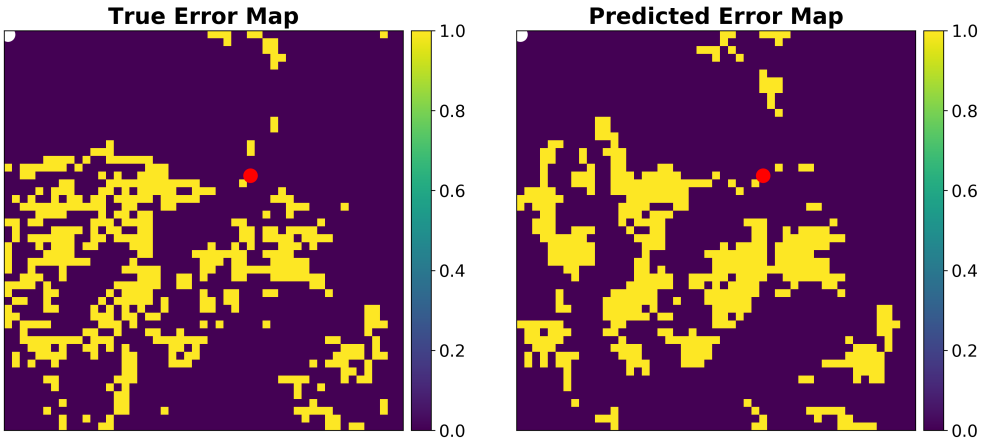
Figure 4.24: Samples collected from 4 error maps as shown by red $\times$ for training the classification ML models

	Classification
ANN	N_{nodes}=10, λ=0.02
RF	N_{fmax}=8, N_l=3

Table 4.9: Regularization parameters and hyperparameters chosen by 5-fold Cross Validation



(a) Classification error map comparison using ANN for basis well at $(4, 19)$

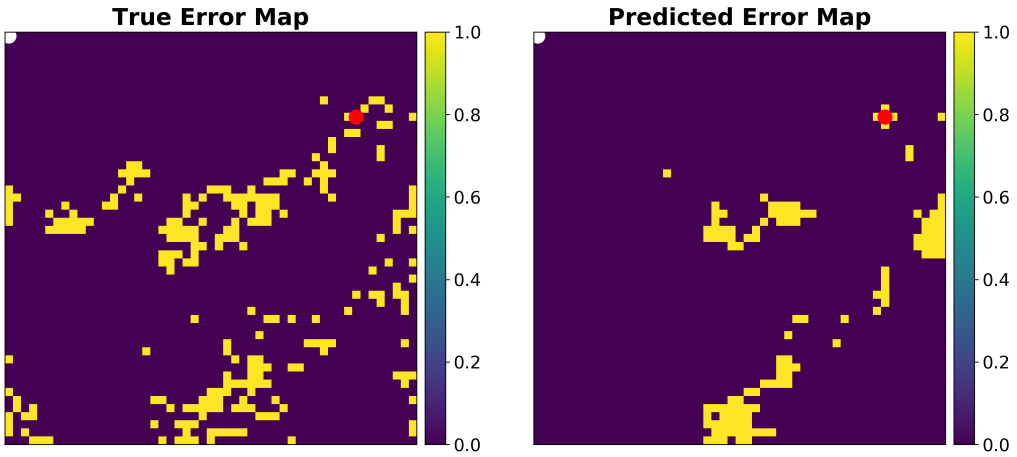


(b) Classification error map comparison using ANN for basis well at $(2, 19)$

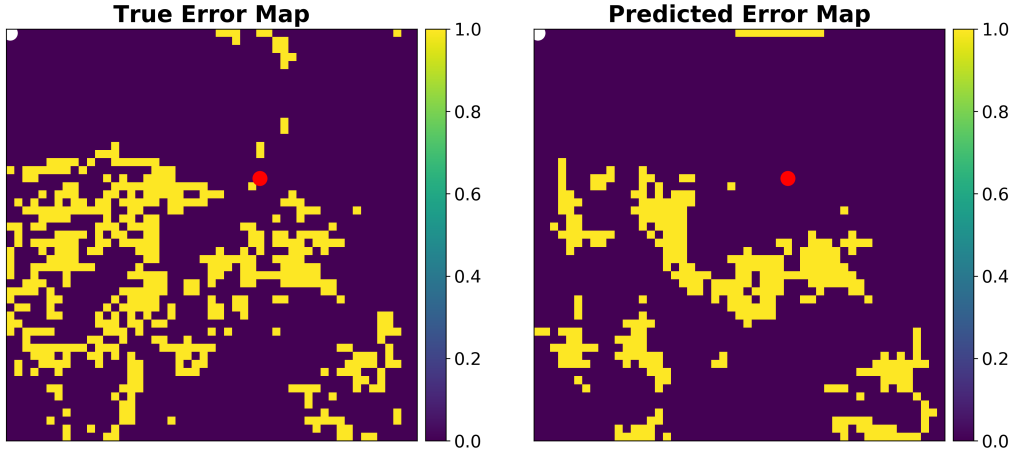
Figure 4.25: ANN classification prediction for an error map corresponding to (a) $L_p = 16, L_s = 23$ for basis well at (44,11) gridblock and (b) $L_p = 15, L_s = 22$ for basis well at (32,19) gridblock

	Classification	
	Train	Test
ANN	100	93.8
RF	100	91.3

Table 4.10: Error map prediction accuracy for ANN and RF classification problem



(a) Classification error map comparison using RF for basis well at $(4, 19)$



(b) Classification error map comparison using RF for basis well at $(2, 19)$

Figure 4.26: RF classification prediction for an error map corresponding to (a) $L_p = 16, L_s = 23$ for basis well at (44,11) gridblock and (b) $L_p = 15, L_s = 22$ for basis well at (32,19) gridblock

4.6.2 Case 2: Error maps of basis obtained by snapshot concatenation of two well locations

As we saw in the observation section (4.2), a basis obtained by snapshot concatenation can provide a good reduced subspace for a wide range of parameters, although the dimensions of the basis can be much larger than a local basis. But if there is no local ROB in the database that is accurate enough for a new well location, we hope to get good accuracy by using a basis from concatenation of snapshots at different parameter values. So in this case, we try to predict the error maps for basis obtained by snapshot concatenation from two well locations by using the proposed methodology. We use the same reservoir properties as Case 1. For the feature set now, we also add the features corresponding to a second well location as shown in Table 4.2. The dictionary of ROBs used for this case is shown in the Table 4.11. The error map is obtained using basis from snapshot concatenation at two different well locations and also considering a dictionary of two sets of basis dimensions. The basis dimensions used here are obtained by using 95% and 99.99% energy criteria. As we consider the incompressible flow case, we observe that the basis dimension change has only a little impact on the error map. We show both regression and classification results for this example. Here, we are interested in mapping the relation, $[\zeta^*, \zeta_{\mathbb{D}}^{well_1}, \zeta_{\mathbb{D}}^{well_2}, L_X] \rightarrow C(\eta_{HF})$. Note, that this formulation can be used for a single set of well configurations used for snapshot concatenation. Since we add features corresponding to two training well locations, we cannot determine the order for features corresponding to two basis wells. Hence, for each basis obtained from different pairs of well configurations, we should have separate ML models.

Well Location	(L_p, L_s)
$(4, 19) + (2, 19)$	(2,3), (7,11)

Table 4.11: Dictionary of ROBs used for predicting basis (obtained by snapshot concatenation) error maps

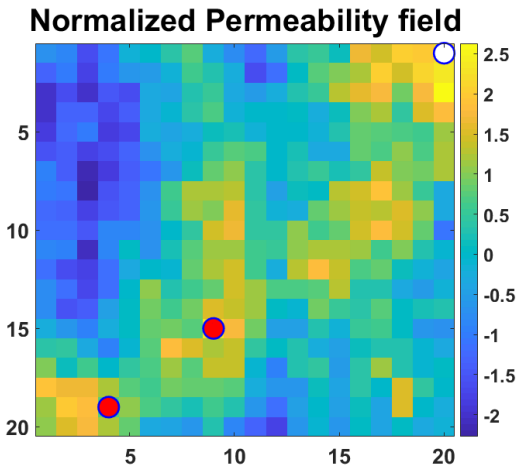


Figure 4.27: Normal scale k_{norm} of permeability field. Two producer well location used for snapshot concatenation

Figures 4.28 and 4.29 show the comparison of predicted error maps by regression NN and RF models for the basis dimension set $(2, 3)$ and $7, 11$ respectively. Similarly, Figures 4.30 and 4.31 show the comparison of predicted error maps by classification NN and RF models for the basis dimension set $(2, 3)$ and $7, 11$ respectively. Table 4.12 shows the prediction accuracy for RF and ANN models for the regression models with optimal hyperparameters and Table 4.13 shows the prediction accuracy for classification models with optimal hyperparameters.

	Regression		Reg- Class		Hyperparameters
	Train	Test	Train	Test	
ANN	99.9	76.75	99.9	90.1	$N_{nodes} = 12, \lambda = 0.07$
RF	95.2	53.8	99.2	88.1	$N_{fmax} = 6, N_l = 3$

Table 4.12: Snapshot concatenation based basis - error map prediction accuracy for ANN and RF regression problem and corresponding optimal hyperparameters

Again, ANN proves to be better than RF for predicting the error maps given highly non-linear complex behavior between the features and output for both regression and classification problems. The classification accuracy was found to better than the regression based

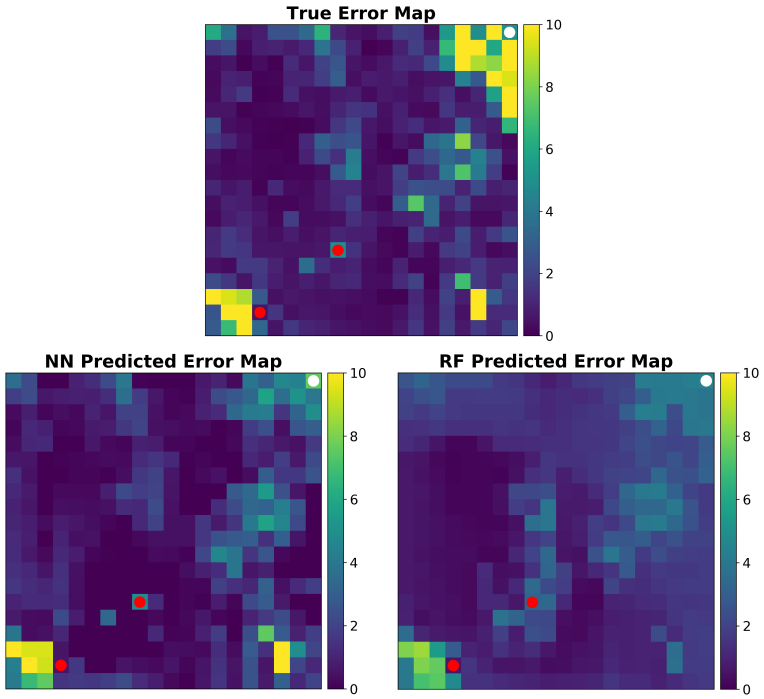


Figure 4.28: Error map prediction (regression) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 2, L_s = 3$ for basis wells at (4,19) and (9,15) gridblock

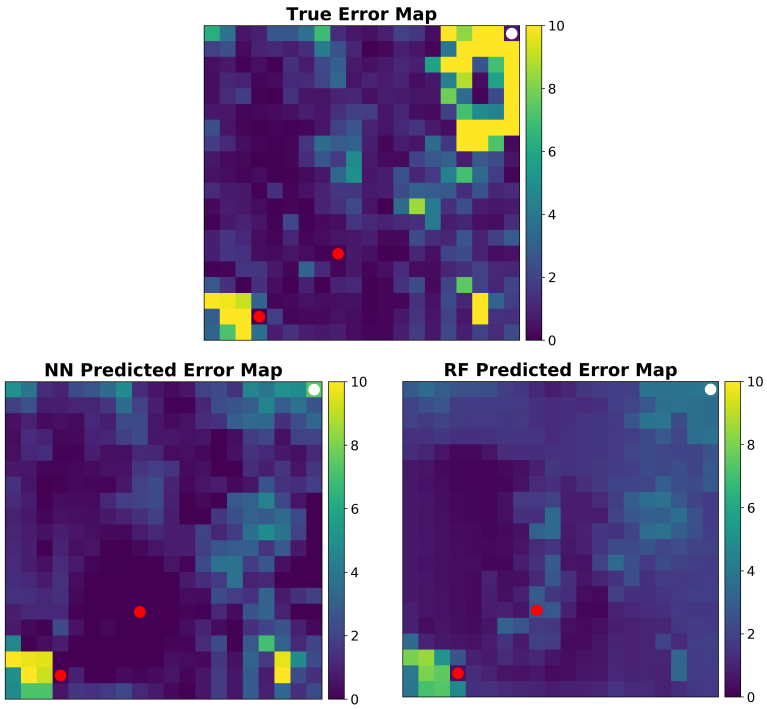


Figure 4.29: Error map prediction (regression) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock

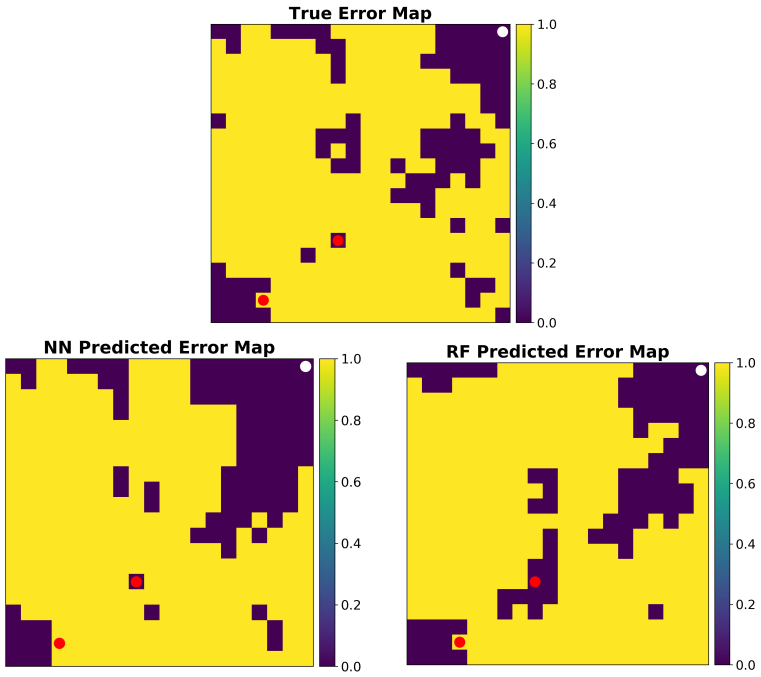


Figure 4.30: Error map prediction (classification) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock

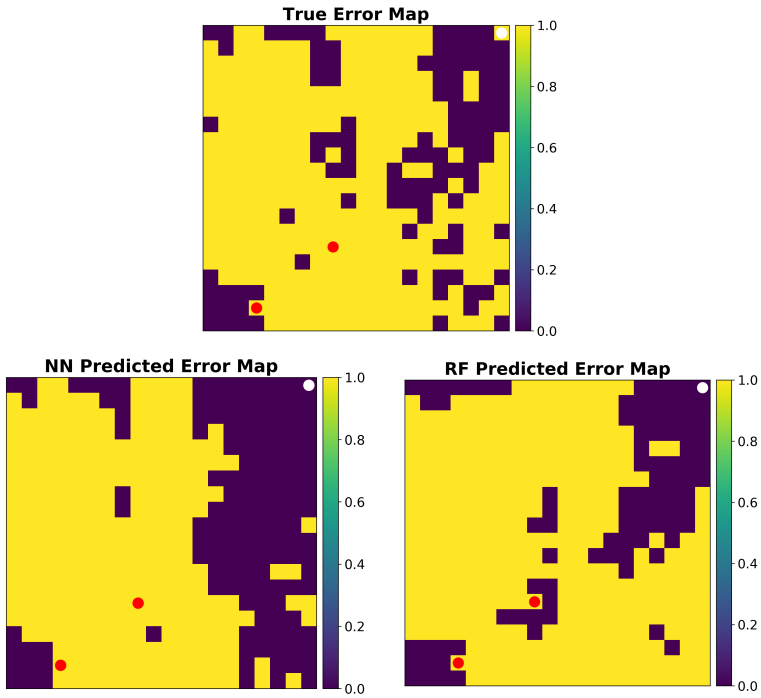


Figure 4.31: Error map prediction (classification) for snapshot concatenation cases using ANN and RF corresponding to $L_p = 7, L_s = 11$ for basis wells at (4,19) and (9,15) gridblock

	Classification		Hyperparameters
	Train	Test	
ANN	100	90.8	$N_{nodes} = 18, \lambda = 0.01$
RF	96.8	88.8	$N_{fmax} = 2, N_l = 3$

Table 4.13: Snapshot concatenation based basis - error map prediction accuracy for ANN and RF classification problem and corresponding optimal hyperparameters

classification accuracy for most of the cases we ran and hence proves to be a better model for adaptively selecting the basis for a new well location. The regression model prediction is useful to estimate the exact error corresponding to a basis for new well locations.

We only show the results for small case here to show that the proposed formulation can also be applied for the cases of snapshot concatenation. But it is difficult to make an a priori selection of wells to concatenate the solutions such that it produces a good quality basis for many new well locations. Also, it requires obtaining separate ML models for separate pair of wells used for basis. Hence, more research work is required in the future before application to bigger reservoir models, with better understanding of basis well locations to be used for solution concatenation and a formulation that can predict error maps for pairs of different well configurations.

4.7 Computational Complexity

The workflow proposed here requires a little higher computational cost in the offline stage corresponding to the samples collected for the dictionary of ROBs. Most of the Local/Global PROM methods developed in the literature have a high computational cost associated in the offline stage to get a good database for a range of parameters. However, with the advent of cloud computing and parallel processing, the offline cost can be significantly reduced given that the offline process involves independent computations. Our examples were run on High Performance Computing (HPC) machine (Texas A&M High Performance Research Computing facility using Ada IBM NeXtScale Cluster operated with Linux, CentOS 6.6,

using 30 nodes with 64 GB, DDR3 1866 MHz). Thus, fine scale simulations required for 30 samples for each error map were computed only once with a single but parallel forward runs. Since, the sample locations in the error maps is the same, we do not need more fine scale simulations for other error maps. After this, we only need to run reduced order models corresponding to each ROB in the database at this 30 locations, also ran in parallel. Training ANN while tuning the parameters is expensive as compared to RF at a cost of better accuracy, which is again run in parallel. Since a smaller case is considered here, the training is computationally cheap but becomes expensive with bigger model architecture and more samples. The online computation cost to predict error at new well location and finding the best basis is very cheap (fraction of a second). Thus, the main computational cost is involved in the offline stage. However, future part of the research will involve better sampling strategies that might require less number of samples with better predictive performance.

4.8 Conclusion and Future Work

A local parametric model order reduction technique using a Machine Learning framework has been proposed for well placement optimization problems. The motivation behind this strategy was the observation that a local basis can act as a good lower dimensional subspace of the state space solutions for a wide range of parameters. A selection criteria for feature set, that includes geometric features and physics based features inspired by flow diagnostics concepts, used for representing the well location performs reasonably well to predict the error maps associated with each local basis using high dimensional regression techniques. The problem has also been proposed as a classification problem, owing to the fact that a relaxed form of basis selection may be sufficient for many applications, which also shows better accuracy compared to predicting exact errors. Results show that regularized Artificial Neural Network show much better accuracy in error prediction as compared to Random Forest for both regression and classification problems. Based on the studies, our first preference would be to choose a local basis if it provides a good accuracy at a new well location, due to its lower dimensions. If a local ROB is not accurate for the new parameter, we may use

snapshot concatenation methods at different well locations that might prove to be a better basis for a wide range of new well locations. In the case of using multiple parameters for basis construction, we construct ML model for each set of multiple parameters used for training because of the way the input features are defined that showed promising results using ANN as compared to RF. Thus, using above workflow, during the well location optimization procedure, the error at a new well location is predicted using the trained ML model for a given ROB and the ROB that produces the least error is used for that well location.

For the future work, we notice that as basis dimensions play a very important role in defining the ROB dictionary for slightly compressible flow, an understanding on finding better prior information about the effect of basis dimensions on new parameters need to be developed. Since we show the cases for incompressible flow, the effect of basis dimensions was not prominent. The examples shown here include the cases where water front does not reach most parts of the reservoir. This method is more data driven rather than physics based as it is based on predicting error in the output quantity of interest without considering the state behavior and the ROBs used for new well location do not honor the controllability properties at that location. This is the main reason this method is difficult to be implemented for the cases when water front reaches the potential producer gridblock, where we have more complex physics of flow and thus higher error in QoI will be observed. In the future, using local in time ROBs and multiple training parameters for a more representative parameter space should be considered to solve this problem. This also includes proposing better sampling strategy using adaptive methods in the parameter space rather than randomly chosen samples as in this paper. This can lead to significant reduction in the number of samples required in the database. In the next section, we propose a global non intrusive PMOR technique that addresses the limitations of proposed local PMOR methodology.

5. NON-INTRUSIVE GLOBAL MODEL ORDER REDUCTION FOR CHANGING WELL LOCATIONS DURING WELL PLACEMENT OPTIMIZATION

In this section, we continue developing methodologies with an aim to develop robust and accurate reduced order models for well location changes during well placement optimization. We again note that we do not solve a well placement optimization problem, rather develop a precursor that aims at MOR construction for changing well locations, which can be eventually used for optimization in the future.

Here, we build on the limitations of the local PMOR strategy mentioned in Section 4 where we use ML techniques to choose an appropriate basis from a dictionary of pre-existing ROBs. However, as we saw, this method is still limited to use for the cases when there is no water cut observed at the producers and for less permeability contrast reservoirs. We seek to develop a method that is not limited by these restrictions and propose a non-intrusive global PMOR technique based on the motivation from observations as shown in the section (4.2) of Section 4. This work aims at establishing a connection between the concepts of model order reduction and machine learning while seeking for the best of the two domains. We begin the section discussing the motivation behind using non-intrusive global PMOR. The main idea is to represent the entire parameter space with a single ROB calculated using fine scale solutions from some representative parameters and then in the online stage, express the solution for a new parameter as a linear combination of the global basis functions. Here, we discuss about parameterizing the states of the reservoir using POD-based model order reduction followed by a way of addressing the physical constraints in the POD method. In the next section, we formulate the non-intrusive PMOR problem using machine learning methods with a brief discussion on the ML methods used. This formulation is then validated using some case studies for prediction of pressure and saturation states at new well locations. Errors in the well block states are then modeled using ML algorithms to correct for the errors in the output quantities of interest (QoI) like oil, water production rates and water-cut. This

error modeling strategy is shown to result in much better accuracy using some case studies in the last section of this section.

5.1 Motivation for non-intrusive global PMOR using machine learning

In this section, we discuss about the motivation behind developing the non-intrusive global PMOR method using machine learning techniques.

In Section 4, we ran some simple case studies in section (4.2), that showed that the local ROBs obtained from single well location are good candidates for ROB at a new well location. However, this was true only for the case of small simulation run times where the water did not flood a significant area in the reservoir. This is expected from the controllability results as was shown in Section 2 that the controllable subspace corresponding to a well location does not preserve the controllability properties at new well locations. Hence, it becomes very difficult to produce accurate results using a reduced subspace of one well location. Also, the observations in section (4.2) show that the ROB corresponding to one well location is only a valid basis for fewer locations when compared to the ROB obtained by concatenating solutions from different well locations. This is the reason behind considering the global ROB strategy for well location problem similar to that used for well control optimization. The global ROB can be obtained as discussed before by random sampling in the parameter space or using greedy algorithms. For the former case, fine scale simulations for randomly distributed parameters are computed and using POD, reduced basis is obtained. It is highly likely that the distribution contains many unnecessary samples or neglects some important samples. Whereas, using greedy algorithms, the samples are carefully chosen using an optimality criteria and usually lead to much better samples. However, the computational expense involved with a greedy sampling procedure for certain kind of problems, especially, highly non-linear dynamical system like reservoir simulation, make it infeasible to use. Thus, for this work, we just focus on random sampling of the candidate well locations.

Most of the PMOR methods discussed above for well control optimization and well location optimization that have been published relies on the simulator source code. Methods

like POD-TPWL [24] are intrusive to some extent as it requires access to Jacobian and residual matrices, that is not easily available for commercial simulators and POD-DEIM [108] is highly intrusive to the source code. One of the non-intrusive methods that has been used for changing well controls is DMD [13]. We propose a new non-intrusive PMOR technique that has been applied in [56, 55] for problems that are time independent or steady state systems. We extend this technique to time dependent problems which is the case for reservoir simulation where states evolve over time. One of the other main advantages of proposing this strategy is it is extremely fast compared to if the global ROB is used within a simulator to project non-linear functions. As discussed before, the reason is, as more parameters are used for concatenating solutions for a global ROB, we expect the basis dimension to increase monotonically and hence projection of non-linear functions become slower. The proposed PMOR technique does not involve such non-linear function projections and can make efficient use of parallel computational facilities as will be discussed in the later section (5.2).

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

5.1.1 Parameterizing states of reservoir using POD

We begin by introducing the parameterization of the reservoir states using proper orthogonal decomposition. POD method was described in section (2.3) of Section 2. Thus, we

do not go into the same details here, but talk about the relevant information needed for the new formulation.

We use the same notations as used in Section 2. Let us consider the states $\mathbf{x}(t,\zeta) \in R^{N_d}$ at time $t \in \mathbb{T}$ and $\zeta \in \mathbb{P} \subset R^{N_\zeta}$, a vector of input parameters. Each column in $\mathbf{x}$ collected over time is also referred to as snapshots. $\mathbf{x}$ represents the pressure and saturation states for our case, i.e. $x = [p, s]$ and ζ corresponds to the well location. In order to construct the global ROB, we collect the states snapshots from different well locations and hence can be written as:

$$S = \{\mathbf{x}(t_i, \zeta_j) | i = 1, ..., N_t, j = 1, ..., N_\zeta\} \tag{5.1}$$

Thus, we have a total of $N_t N_\zeta$ snapshots, with each of the N_ζ parameter has N_t snapshots all concatenated in a single snapshot matrix S. The N_ζ parameters can be selected using experimental design sampling methods like Lattice Hypercube Sampling (LHS), full factorial design etc. or randomly. Thus, POD entails obtaining a reduced order basis (ROB) by taking a singular value decomposition of the snapshot matrix S and then selecting first few vector r (reduced dimension) of the left singular matrix by certain energy criteria on the singular values. This ROB is represented by $\mathbf{\Phi}$. The detailed procedure is explained in Section 2. $\mathbf{\Phi} = \{\phi_1(x), \phi_2(x), ..., \phi_r(x)\}$, where $\phi_i(x), i = 1, ..., r$, are called the reduced basis functions and we seek solutions at any parameter that are a linear combination of these basis functions. We assume that these basis functions span a space Φ_{rb} called the reduced basis space:

$$\Phi_{rb} = span\{\phi_1(x), \phi_2(x), ..., \phi_r(x)\} \tag{5.2}$$

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

$$\tilde{\mathbf{x}}(t,\zeta) = \sum_{i=1}^{r} c_{\mathbf{x}i}(t,\zeta)\phi_{\mathbf{x}i}(x) \tag{5.3}$$

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

$$\tilde{\mathbf{p}}(t,\zeta) = \sum_{i=1}^{r_p} c_{pi}(t,\zeta)\phi_{pi}(x) \tag{5.4a}$$

$$\tilde{\mathbf{s}}(t,\zeta) = \sum_{i=1}^{r_s} c_{si}(t,\zeta)\phi_{si}(x) \tag{5.4b}$$

$$\tilde{\mathbf{p}}(t,\zeta) = \bar{\mathbf{p}} + \sum_{i=1}^{r_p} c_{pi}(t,\zeta)\phi_{pi}(x) \tag{5.5a}$$

$$\tilde{\mathbf{s}}(t,\zeta) = \bar{\mathbf{s}} + \sum_{i=1}^{r_s} c_{si}(t,\zeta)\phi_{si}(x) \tag{5.5b}$$

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

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

$$\mathbf{c}_{\mathbf{x}}(\zeta) = \Phi_{\mathbf{x}}^T \mathbf{x}(\zeta) \in R^{r\times t}, \tag{5.6}$$

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

5.1.2 Addressing physical constraints in POD

In this method, we also enforce physical constraints in the form of POD representation. This is done by writing the equation (5.3) in a different way as:

$$\tilde{\mathbf{x}}(t,\zeta) = \bar{\mathbf{x}} + \sum_{i=1}^{r} c_{\mathbf{x}i}(t,\zeta)\bar{\phi}_{\mathbf{x}i}(x) \tag{5.7}$$

Here, $\bar{\mathbf{x}}$ is called the particular solution which enforces certain characteristics of the spatial behavior of states during POD-based prediction. For our case, this particular solution is defined as the mean of all the snapshots corresponding to all the parameters in snapshot matrix. By doing so, as expected, the accuracy of state prediction increased using the formulation proposed in the next section (5.2). The procedure to calculate the global basis functions remain the same as above, except now, the POD basis $\bar{\Phi} = \{\bar{\phi}_1(x), \bar{\phi}_2(x), ..., \bar{\phi}_r(x)\}$ is computed on the mean subtracted snapshot matrix:

$$\tilde{S} = \{\mathbf{x}(t_i,\zeta_j) - \bar{\mathbf{x}} | i = 1, ..., N_t, j = 1, ..., N_\zeta\} \tag{5.8}$$

After, the basis coefficients are obtained, the states at new parameters and a given time are computed as equation (5.8).

5.2 Global PMOR problem formulation

Once the global ROB is obtained using the method shown above, a traditional way of solving such system is solving the non-linear reservoir simulation equation online and getting the coefficients of the basis for a new parameter by projecting the non-linear Jacobian and residual functions. However, it has been shown several times in the literature that this technique shows modest improvements in the computational speedups as it requires computing the fine scale non-linear functions during each Newton iteration. Also, for well location changes, implementing the simulator intrusive POD-based online procedure caused lot of stability issues and it is very challenging to decide on the suitable dimensions of

global basis. Changing well location as the parameter for PMOR is observed as much more challenging as compared to well control changes as even a global PMOR technique is found difficult to represent the controllability properties of a new well location. On way to address these issues is using non-intrusive reduced basis methods, where the basis coefficients are obtained by interpolating the ROBs or ROMs over the parameter domain. But, the reduced bases belong to non-linear, matrix manifolds and hence the standard interpolation can fail in preserving the constraints characterizing those manifolds and requires having a large dataset [47, 21]. We present an alternative way to overcome this issue by multidimensional mapping of input parameters to the basis coefficients given a global ROB. We use ML techniques that have capabilities of mapping complex non-linear relationships and specially suitable for interpolation of basis coefficients where the parameters have a non-affine dependence. Similar approach was applied to steady state cases in [56, 55]. We extend this technique to time dependent problems. Thus, the ML model $\mathscr{F}$ is trained to learn the relation:

$$\mathscr{F} : [\zeta_j, t_k] \rightarrow [c_{\mathbf{x}1}(\zeta_j, t_k), c_{\mathbf{x}2}(\zeta_j, t_k), ..., c_{\mathbf{x}r}(\zeta_j, t_k)], \qquad j = 1, ..., N_\zeta \;\; and \;\; k = 1, ..., N_t \tag{5.9}$$

$$[c_{\mathbf{x}i}(\zeta_j, t_k)]_{i=1}^{r} = \Phi_{\mathbf{x}}^T \mathbf{x}(\zeta_j, t_k), \;\; \mathbf{x} = [p, s]^T \tag{5.10}$$

Note that, the basis $\Phi_{\mathbf{x}}$ is obtained from the mean subtracted snapshot matrix and the basis dimensions of pressure and saturation are different shown by r here. ζ_j are the parameters representing the well location used for training which are the features of the ML model. In order to consider the temporal evolution of the coefficients, we also add time as one of the features and the coefficients as a function of time as the outputs of the ML model.

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

$$\mathscr{F}(\zeta^*, t_k) = [c_{\mathbf{x}i}(\zeta^*, t_k)]_{i=1}^{r} = \mathbf{c}_{\mathbf{x}}(\zeta^*, t_k) \tag{5.11}$$

$$\tilde{\mathbf{x}}(\zeta^*, t_k) = \sum_{i=1}^{r} c_{\mathbf{x}i}(t_k, \zeta^*)\phi_{\mathbf{x}i} + \bar{\mathbf{x}} = \Phi_{\mathbf{x}} c_{\mathbf{x}}(\zeta^*, t_k) + \bar{\mathbf{x}} \tag{5.12}$$

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

5.2.1 Machine Learning and feature selection

As mentioned in the section (5.2), we are interested in understanding the relationship between the input parameters representing well locations and the basis coefficients of pressure and saturation. The inputs representing the well locations are defined in high dimensional spaces as will be shown later, and the outputs, which are the POD coefficients, also are expected to be very high dimensional based on the number of basis functions chosen that increase with increasing number of training well locations and size of the reservoir. Such high dimensional input-output relation and a highly non-linear relation that is very difficult to define explicitly calls for machine learning strategies. However, for engineering applications like the ones considered, it becomes very challenging to generate a large data set for the machine learning algorithms to capture the underlying dynamics. Thus, there is trade-off that needs to be considered between getting a good training data set and the least computational expense at building a good predictor. Thus sometimes, specially for such applications as developing MOR for changing well locations in the reservoir, it is just beneficial to use simpler ML algorithms that are faster to train and can still capture the underlying non-linearity. Thus, we consider using Random Forest (RF) regressors as our algorithm of choice, since they are much faster to train and capable of mapping complex input-output relationships while using concepts like bootstrap aggregation to avoid overfitting. The results are compared to those obtained by Neural Networks (NN) which are widely used for mapping complex high dimensional input-output relations. They have been used for non-intrusive ROM in [55]. According to many applications, NN is found to be a feasible method where we have access to huge data sets and spending a lot of time in training is a viable option. We however, analyze its performance as compared to that obtained by a fast algorithm like Random Forests.

Other faster and simpler methods like kNN and multi polynomial regression were also considered but their performance was very poor for such complex problem and hence not shown here. The hyperparameters for the random forest regressor model are again chosen to be the minimum number of splits at the leaf node (N_l) and the number of features to search for split (N_{fmax}). For NN model, we use number of nodes in the hidden layers (N_{nodes}) and the regularization parameter (λ) from equation (4.8) as the tuning parameters. For all the cases shown in this section, the best model was found through cross validation.

For input features of the machine learning model, we consider the same observations for feature construction as that used in error map modeling. However, here we do not have a local basis corresponding to a parameter that needs to be selected and hence for the current problem, the new set of features used are listed in Table 5.1. Note that the features correspond to information that we think best describes the well location and also includes time information as discussed previously to account for understanding the temporal behavior of the POD basis coefficients.

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

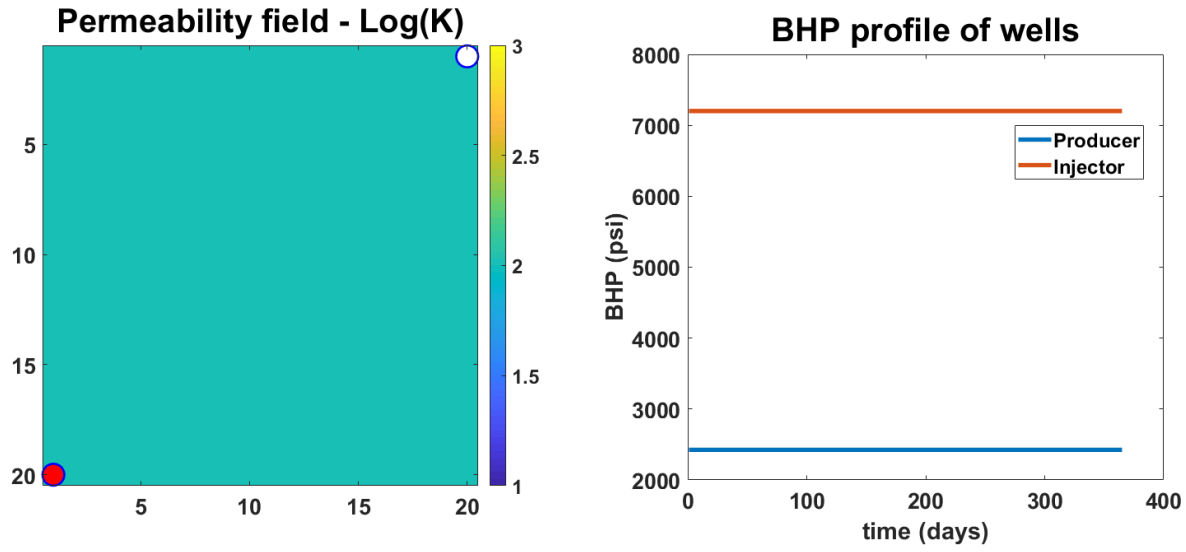
Table 5.1: Geometric and physics based features for ML model construction corresponding to the well configuration

5.2.2 Remarks

As we have formulated the method, we now demonstrate its performance with some case studies. However, before that, we point out some specifics of these cases. The injector for all the cases shown here is at a fixed location and the producer is considered as the parameter i.e. it can change the location in the reservoir. All the new producer well locations have the same BHPs to consider only the well location as a system parameter. The workflow can be extended in the future to also include varying well BHPs as the feature but not in the current scope of work. The simulations are run long enough to observe significant water cut at most of the well locations in order to introduce complexity to the problem which was a limitation in the local ROB based workflow discussed in Section 4. For the cases shown below, we use random sampling in the parameter space to train the ML models. Experimental design sampling techniques or greedy sampling procedures as listed before can be used for sampling the parameter space for better representative parameters, but, for simplicity and fast sampling we choose the parameters randomly for training purpose. However, the number of random samples chosen is an open challenge that needs to be addressed in the future. With random sampling we usually expect a lot more samples than something like greedy sampling procedures, in efforts to capture representative samples for global ROB. The main aim here is to show the validity of proposed methodology given the high complexity of the problem compared to that of changing well control problem and hence we use relatively large training sample sizes here.

5.3 Case Study

In this section, we present the numerical results for the proposed non-intrusive global PMOR technique. We start with a simple example of a homogeneous reservoir model. The model has 20×20 gridblocks and has one injector and one producer well as shown in Figure 5.1a. The porosity is set constant to 0.2 and the relative permeability model is defined by Corey function of degree 2. The permeability of the field is considered to be $100mD$. The



(a) (b)

Figure 5.1: (a) Homogeneous permeability reservoir with 1 injector (white) and 1 producer (red). Producer is the parameter and can change location anywhere in the reservoir (b) BHP profiles of injector and producer that is set constant for all well configurations considered

2-phase flow is considered incompressible neglecting the capillary and gravitational effects. All the producers are produced with a constant BHP of 2425 psi and injector injects water a constant BHP of 7200 psi. We look to change the producer locations only and the injector location is fixed at $(20, 1)$ gridblock. The simulations are ran for a period of 1 year with which we observe watercut at all the producer locations.

The problem is setup as follows. First, in order to construct a global ROB, we randomly sample the well locations. Fine scale simulations are run for each well location and the snapshots are collected in a single snapshot matrix as shown in equation (5.1). We then follow the procedure as described before to train the ML model. Since, we have 400 grid blocks, there are 399 locations we consider where the producer can move (1 injector block). So we randomly sample 100 cases to train the ML models and the rest 299 cases are used as test cases.

5.3.1 Energy of eigenvalues - Global PMOR

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

PMOR for a well control usually use a lot more saturation basis as compared to pressure basis for a fixed well configuration as the pressure behavior has a fast moving front and is not expected to change significantly for for a fixed well location as compared to the saturation behavior that has a very slow moving front with major updates occurring at the saturation front. For the problem of changing well locations, at each well location we have the lowest or highest pressure point in the reservoir corresponding to the well BHP and hence there is also significant difference in the pressure behavior observed moving from one well location to the other. This has a significant impact on the number of well configurations used in the training set. A typical energy criteria used for choosing the basis dimensions is above 90% of the energy of eigenvalues from the equation (2.25). As can be seen in Figure 5.2, we begin with $N_\zeta = 10$ where, as usual, we have a higher number of saturation basis than pressure basis satisfying more than 90% energy criteria. However, after adding more parameters, these trend changes, especially after $N_\zeta = 200$. It requires significantly more number of pressure basis to capture to understand the dynamics. Note, that the pressure basis energy plateaus at a point equaling number of training parameter N_ζ. As we add more training parameters, we see that the saturation basis energy profile does not change significantly.

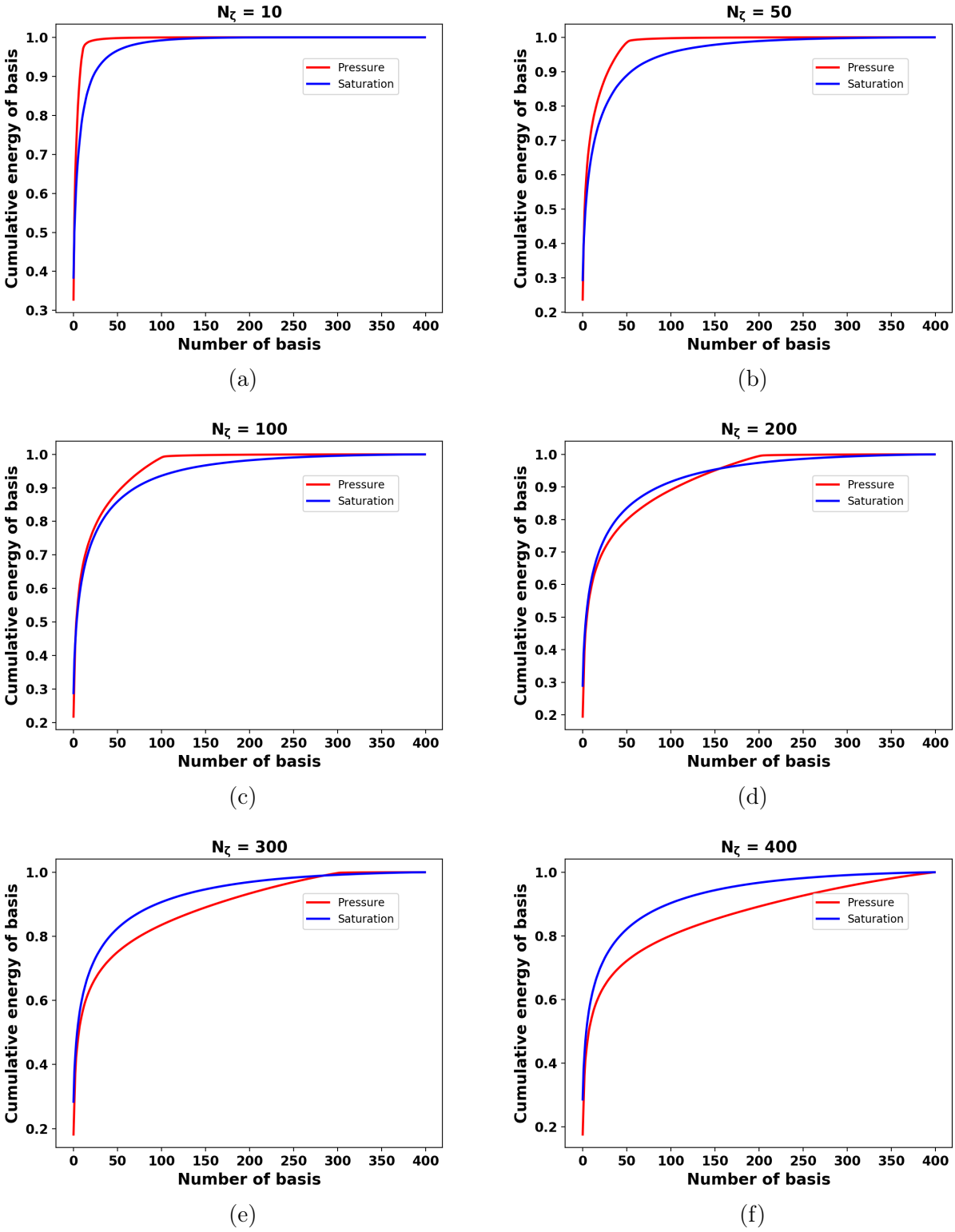
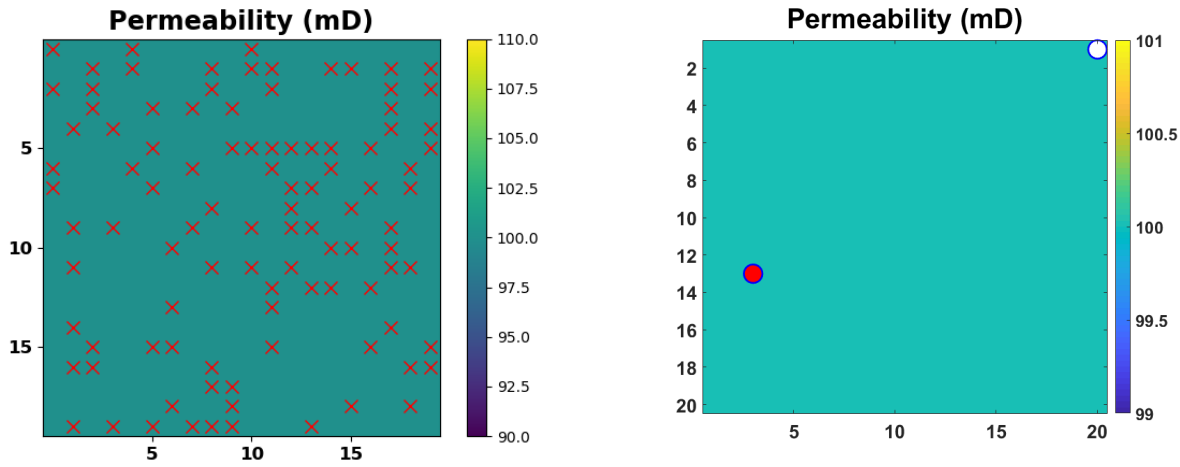


Figure 5.2: Change in cumulative energy of the basis as calculated by the magnitude of eigenvalues for increasing number of parameters (well configurations) in the state snapshot matrix

Thus, for many cases at hand, depending on the training parameters selected, we may observe higher number of pressure basis as compared to saturation basis. This observation also gives a notion on the complexity of the problem as there is a trade-off between choosing the number of training samples and the basis dimension required for a good quality basis over the domain of parameters. For example, we may choose 10 training samples and capture 99.99% energy basis but still may not produce accurate results as each parameter has very different dynamical behavior. So it may sometimes be good to have more training samples and a lower energy criteria at the expense of running more fine scale simulations in the training phase and getting significant speedups in the testing phase.

5.3.2 Results

We now show the prediction results for new well locations not included in the training set. Figure 5.3 shows the training samples of producer well locations each simulated one at time and the new well location in the first test set for which we predict the POD coefficients.



(a) (b)

Figure 5.3: (a) Training samples of producer well locations shown by red crosses (b) Test case 1 well location at (3,13) in homogeneous reservoir

We first use Random Forest regressor to compute the POD basis coefficients. For pressure, the basis dimension is 99 and that for saturation is 113, which corresponds to the output dimension for ML models of pressure and saturation respectively. The outputs are predicted using the relation in equation (5.11). and for this example the optimal tuning parameters for RF regressor found are shown in Table 5.2.

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

Table 5.2: Hyperparameters chosen by 5-fold Cross Validation for Random Forest Regressor using 100 training samples

Figures 5.4 and 5.5)show the comparison between predicted and true states (pressure and saturation) for the new well configuration at two different times. We also show the error in states along time. As can be seen in Figure 5.6, prediction of the overall state behavior is fairly accurate but there is discrepancy in the solution close to the producer well location for pressure, and at the fluid front for saturation.

The quantities of interest that are the outputs of the simulation like fluid production rates, water cut etc. are solely dependent on the state quantities at the well blocks through the Peaceman equation (2.4). Thus, for this prediction results we expect some error in these quantities due to higher prediction error close to the wells. To quantify these discrepancies, we show the pressure and saturation solution comparison at the producer well block in Figure 5.8. As we can see, the predicted solution captures the overall trend of dynamical behavior of the states but has kind of a bias in the solution. Figures 5.9a and 5.9b show the comparison of oil production rate and water cut between the true and predicted solution.

We also consider a second test case with new producer well location as shown in Figure 5.11 to validate the method. Figure 5.11 shows the pressure and saturation comparison

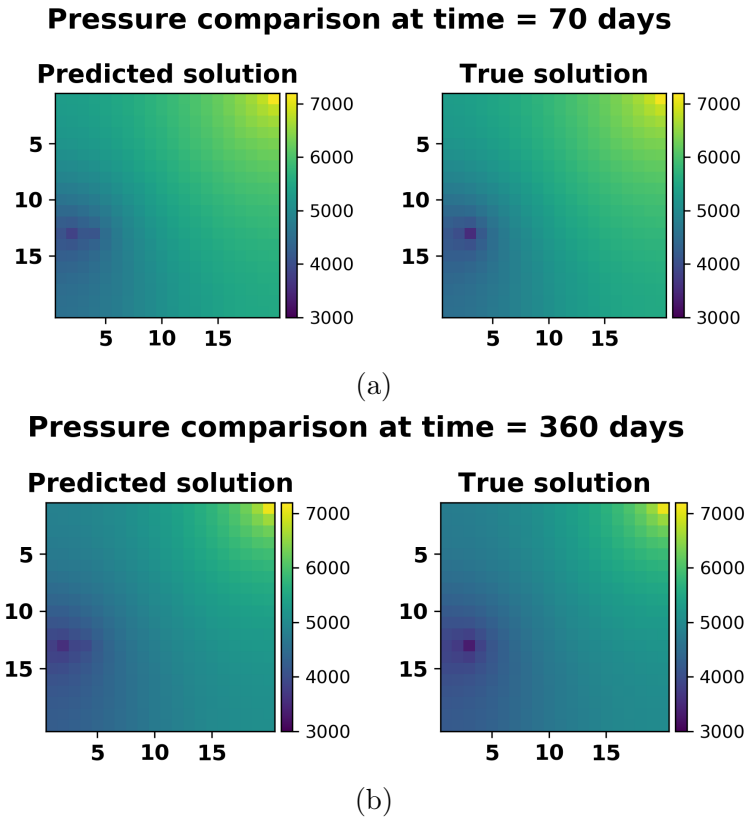
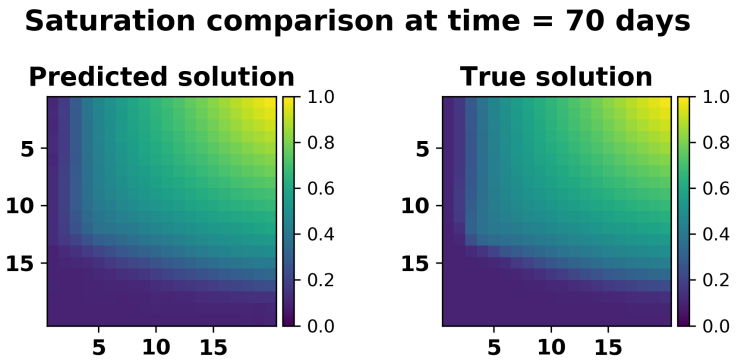


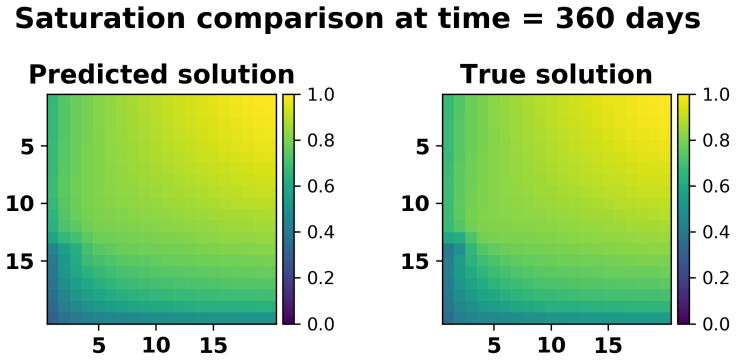
Figure 5.4: Pressure solution comparison at (a) Time = 70 days and (b) Time = 360 days

between ML predicted and true solutions. Both the solutions show a very good agreement visually. For a more detailed analysis, we also plot the well block pressure and saturation in Figure 5.12a and (5.12b). This case shows that ML predicts the solution with a very good accuracy which eventually reflects in the Figure 5.13, where we plot the oil production rate and water cut.

After plotting the results from other test cases, we found that for some cases the prediction is reasonably accurate, similar to that in test case 2. But there are many well configurations that show discrepancies as the test case 1 which overall captures the solution trend but there is some bias associated. So, we need to analyze the reason behind the solution discrepancy which can either be due to the error in machine learning model or due to the quality of global basis. In order to get an intuition about these two factors, we first analyze the

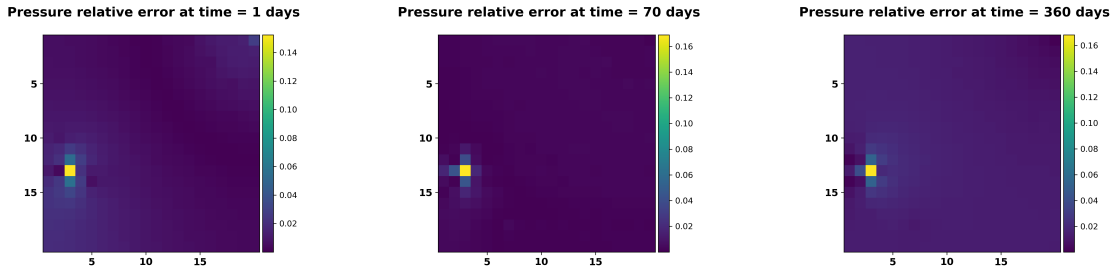


(a)



(b)

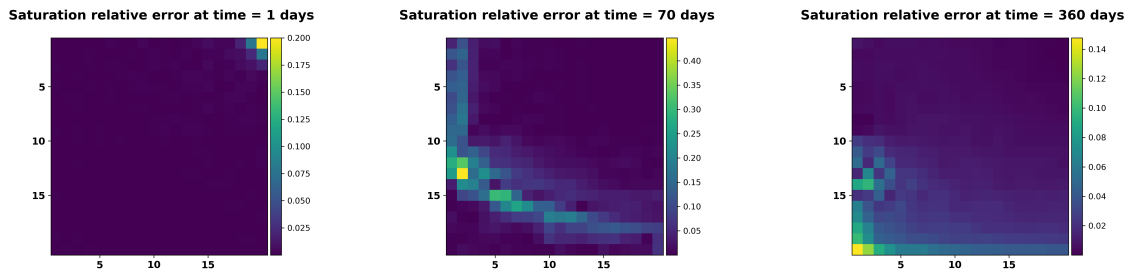
Figure 5.5: Saturation solution comparison at (a) Time = 70 days and (b) Time = 360 days



(a) (b) (c)

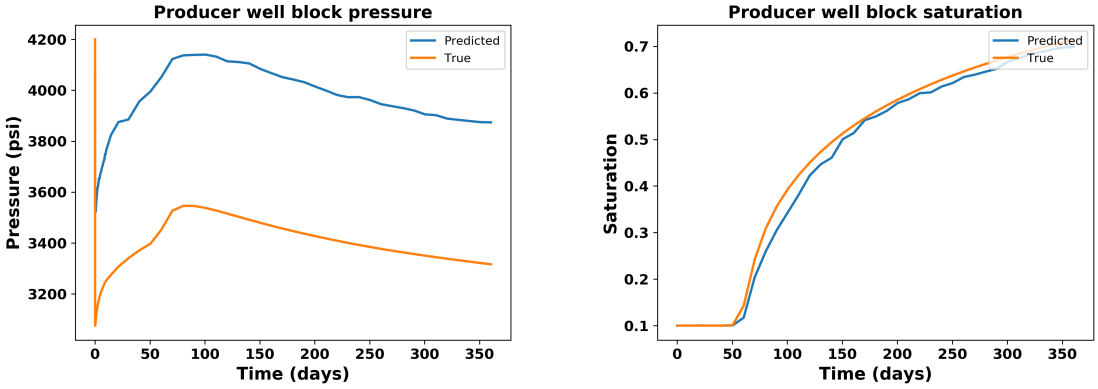
Figure 5.6: Relative error in pressure at time = (a) 1 day , (b) 70 days and (c) 360 days

machine learning model performance. In Figures 5.15 and 5.16, we compare the predicted and true basis coefficients for pressure and saturation respectively for different timesteps for test case 1. These show that the ML model predicts the coefficients with a good accuracy



(a) (b) (c)

Figure 5.7: Relative error in saturation at time = (a) 1 day , (b) 70 days and (c) 360 days

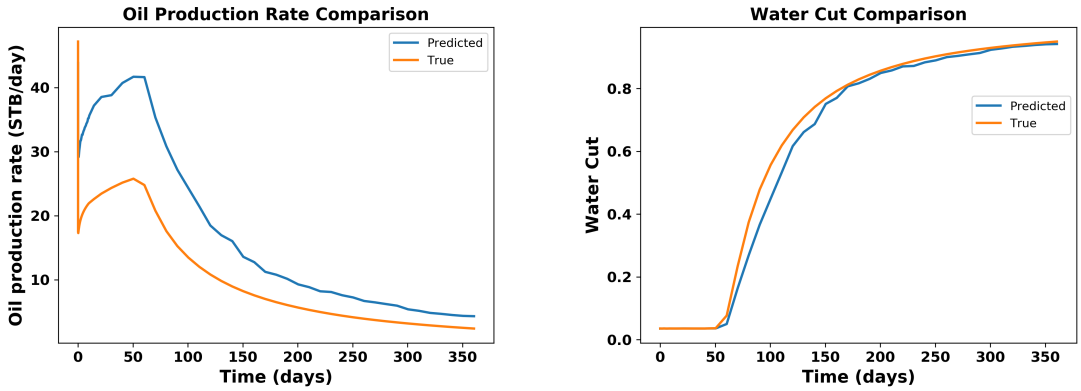


(a) (b)

Figure 5.8: Well block state solution comparison (a) Pressure and (b) Saturation

except for some instances like the first pressure basis at time 360 days. The true coefficients for a new parameter ζ^* are obtained by projecting the fine scale simulation on the subspace spanned by global basis Φ by computing $\Phi_{\mathbf{x}}^T \mathbf{x}(\zeta^*)$.

Thus, the performance of ML model suggests that the discrepancies near the well locations are less likely due to the errors in ML model specially the fact in understanding that the new producer well location should be the lowest pressure point in the reservoir. Thus, now we analyze the quality of the global basis Φ used on test case 1. Here, we look back to Section 2, where we introduced the concept of MOR error that has a orthogonal component which basically results from neglecting the state projection on the orthogonal subspace. At a given



(a) (b)

Figure 5.9: Quantities of Interest comparison (a) Oil production rate and (b) Water cut

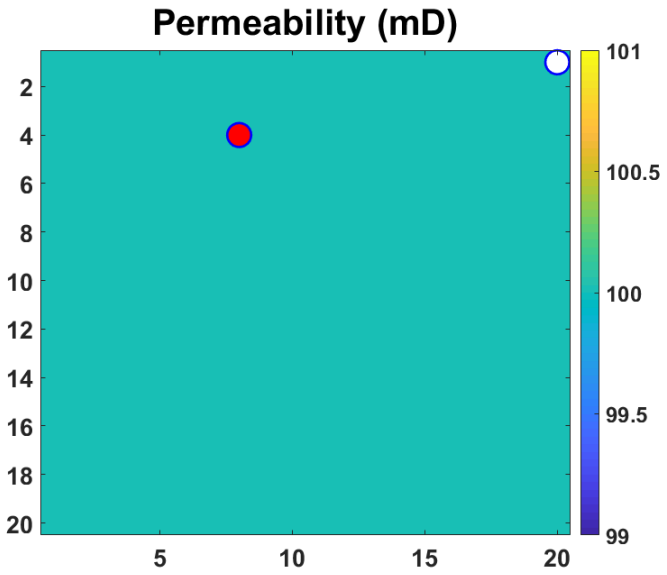


Figure 5.10: Test case 2 with producer at (8,4) grid block and injector at (20,1) gridblock in homogeneous permeability reservoir

time instant for a new parameter ζ^*, it can be written as:

$$\begin{aligned} \epsilon_{\Phi^\perp}(t,\zeta^*) &= (I_n - \Pi_{\Phi,\Phi})\mathbf{x}(t,\zeta^*) \\ \Pi_{\Phi,\Phi} &= \Phi\Phi^T \end{aligned} \tag{5.13}$$

Thus, the deviation of the product $\Phi\Phi^T$ from the identity matrix gives an a priori es-

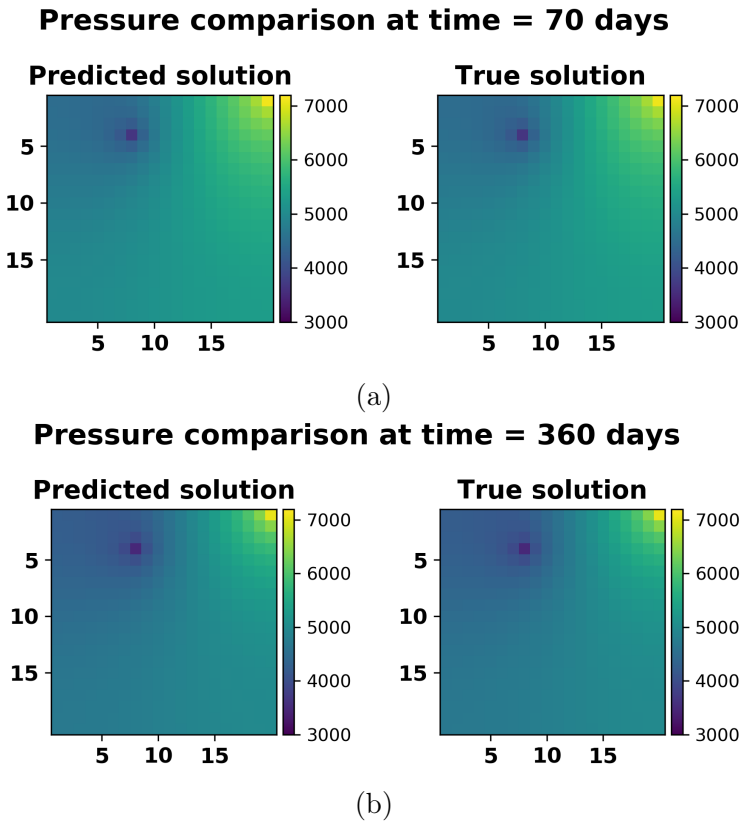
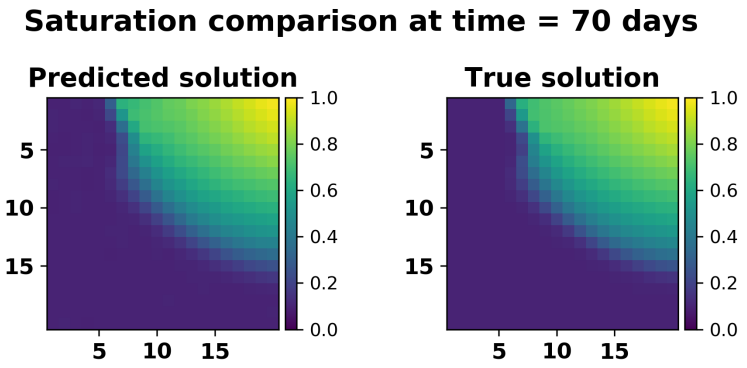
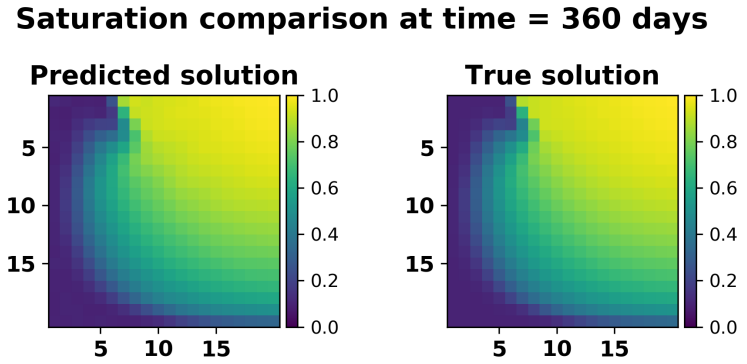


Figure 5.11: Pressure solution comparison at (a) Time = 70 days and (b) Time = 360 days

timation about the quality of basis for the new parameter. To visualize this for our case, we project the true coefficients obtained by $\Phi_{\mathbf{x}}^T \mathbf{x}(\zeta^*)$ back to the fine scale domain. This is basically performing the operation $\Pi_{\Phi,\Phi})\mathbf{x}(t, \zeta^*)$ which we refer to here as the true orthogonal solution. The Figures 5.17a and 5.17b show the predicted orthogonal solution, true orthogonal solution and true fine scale solution for pressure and saturation respectively at the end of simulation for test case 1. As can be seen, the true orthogonal pressure and saturation solutions differ from the true solutions around the producer well. This is clearly visible for the case of pressure solution. However, the predicted orthogonal pressure and saturation solutions are in a very good agreement to their true orthogonal solutions. This is an indication of the quality of basis that is the main reason behind the higher errors around producer well locations. This indicates that even with a global basis of parameters, it is very

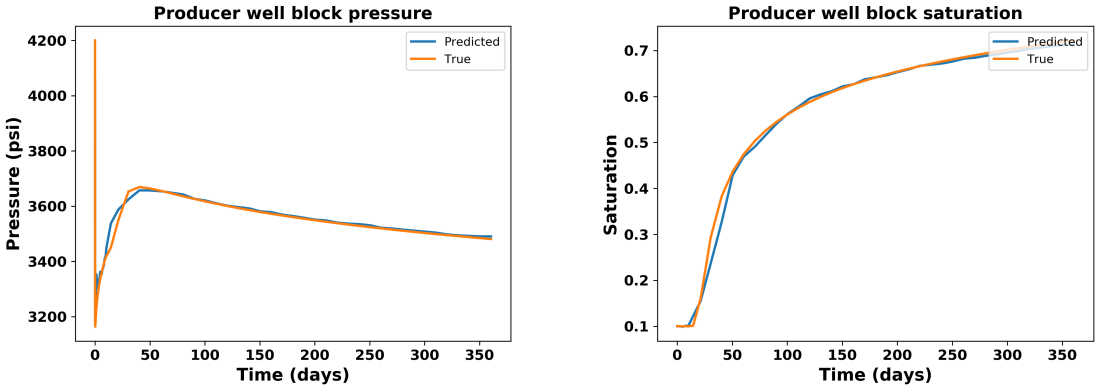


(a)



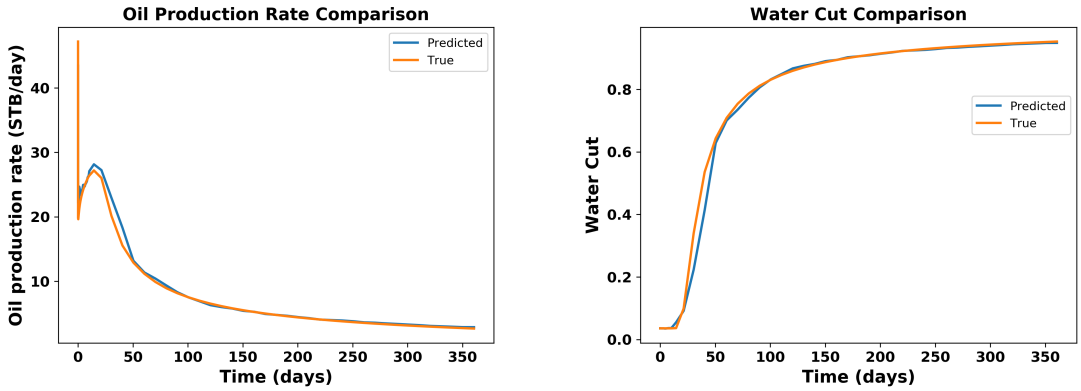
(b)

Figure 5.12: Saturation solution comparison at (a) Time = 70 days and (b) Time = 360 days



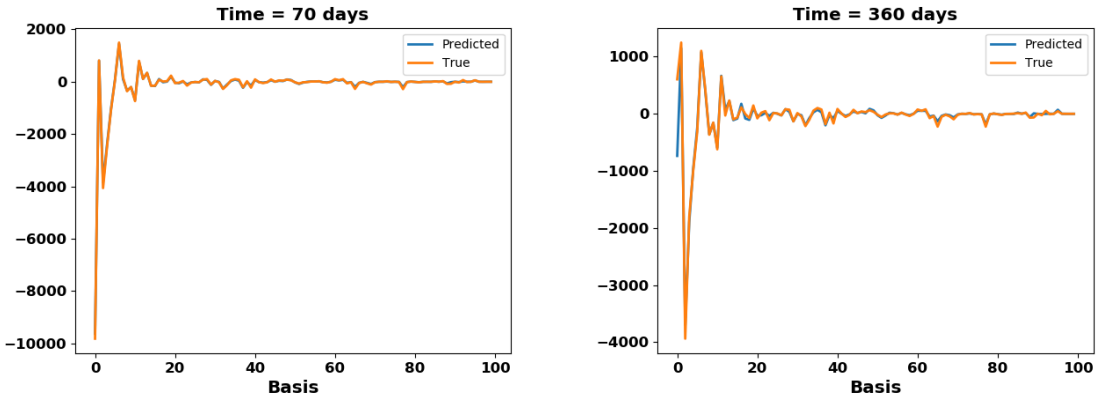
(a) (b)

Figure 5.13: Well block state solution comparison (a) Pressure and (b) Saturation



(a) (b)

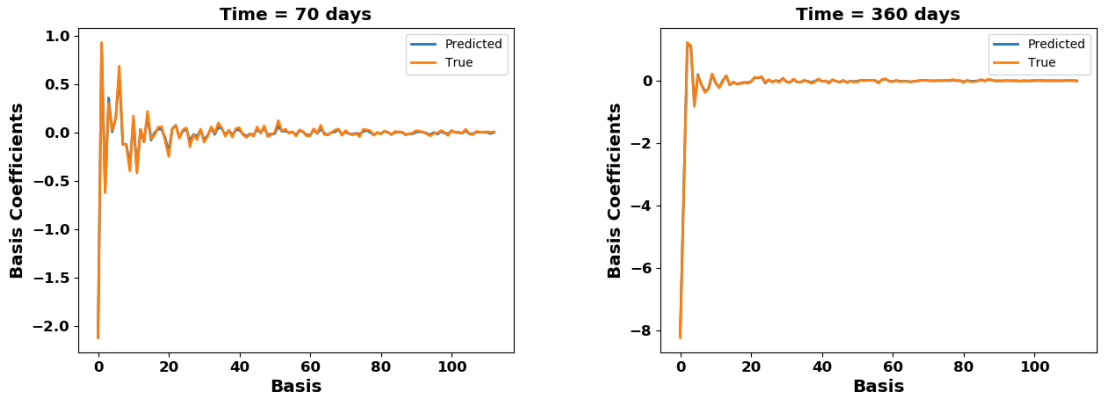
Figure 5.14: Quantities of Interest comparison (a) Oil production rate and (b) Water cut



(a) (b)

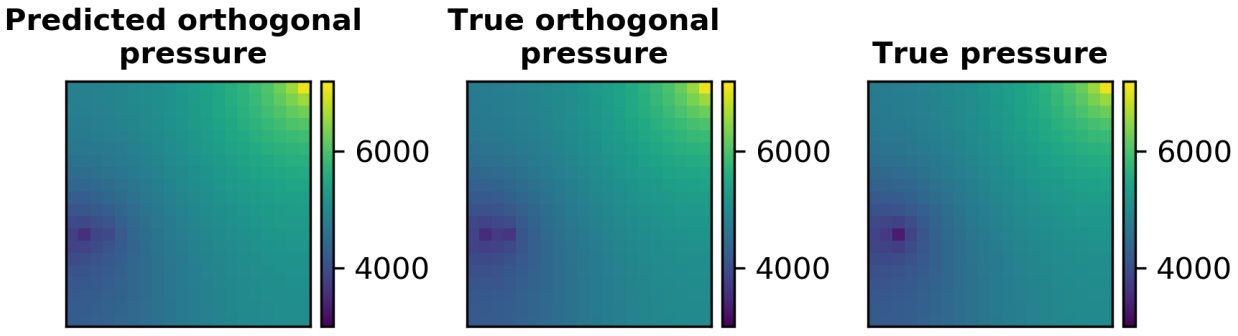
Figure 5.15: True and ML predicted pressure basis coefficient comparison at time = (a) 70 days and (b) 360 days

difficult to preserve the controllability properties for all the new well locations. The global basis here proves to be a good quality basis for some cases like the test case 2. Thus, the best possible way to alleviate this problem is to try getting the most representative set of parameters in the training sample set which can be a future direction of research.

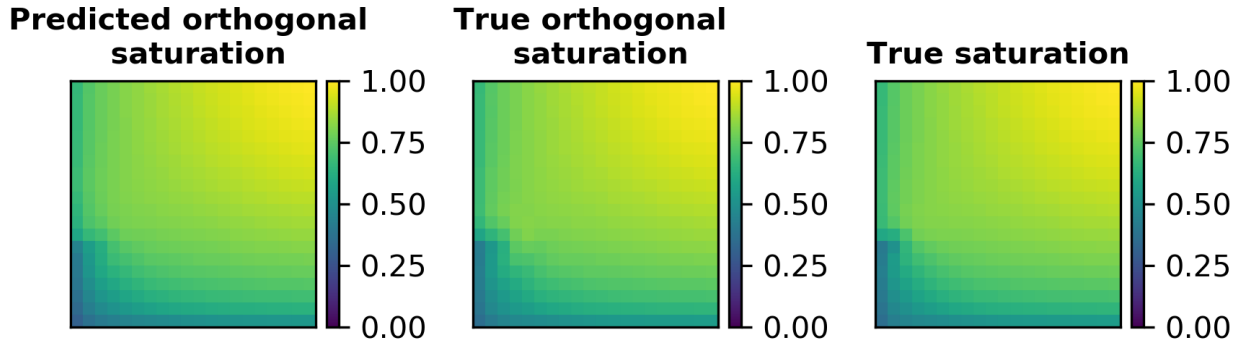


(a) (b)

Figure 5.16: True and ML predicted saturation basis coefficient comparison at time = (a) 70 days and (b) 360 days



(a)



(b)

Figure 5.17: Comparison of predicted orthogonal and true orthogonal solutions with true solution at time = 360 days for (a) Pressure and (b) Saturation

5.4 Modeling error correction

As we saw that finding a good global basis for the entire domain of parameters is very difficult, we get solution discrepancies although it captures the overall trend of the solution. So with the given information about the reduced order model, we propose an error correction method to adjust the solution discrepancies. The error correction model takes into account the information about the reduced order model solution at a new parameter and then predicts the correction required in the quantity of interest. This model is constructed using machine learning techniques to account for high dimensional feature space and mapping the complex non-linear relationship between inputs and outputs. The quantity of interest can be the oil/water production rate, water cut or the well block states. For the current work, we just consider the well block states (pressure and saturation) as the QoI to be corrected as these states directly transfer to the correction in production rates and water cut. At a given instant in time, the error in the well block states is:

$$\begin{aligned} &\Delta \mathbf{x}_{wb}(\zeta, t_k) = (\mathbf{x}_{wb}(\zeta, t_k))_{fine} - (\mathbf{x}_{wb}(\zeta, t_k))_{PMOR} \\ &where, \quad k = 0, 1, ..., N_t \ \ and \ \ \zeta \in \mathbb{P} \end{aligned} \tag{5.14}$$

Thus, we are interested in determining $\Delta \mathbf{x}_{wb}$ for a new well location over time given the reduced model solution obtained in equation (5.12). We also note that, the saturation map comparison at different times during prediction Figure 5.5 and comparison of the predicted injection rates with true injection rates showed good agreement. This information helps us consider the pore volumes injected predicted by the reduced model as a good reduced model information. The formulation of the error correction model is thus given by:

$$\mathscr{E}(\zeta, t_k, \tilde{\mathbf{x}}, PVI_r) \rightarrow \Delta \mathbf{x} \tag{5.15}$$

This model is constructed using ML techniques, where, the ζ correspond to the well location parameter and hence represented by the same features as used in global PMOR formulation.

$\tilde{\mathbf{x}}$ correspond to the PMOR predicted solution at all the gridblocks in the reservoir and PVI_r is the pore volumes injected as predicted by the reduced order model. However, since we are interested in correction of states at the well gridblock only, it is worthwhile considering the relation between the reduced solution at gridblock location $\tilde{\mathbf{x}}_{wb}$ to the correction $\Delta\mathbf{x}_{wb}$. By this assumption, the formulation changes to:

$$\mathscr{E}(\zeta, t_k, \tilde{\mathbf{x}}_{wb}, PVI_r) \rightarrow \Delta\mathbf{x}_{wb} \tag{5.16}$$

This formulation is thus a global error model that is used for the entire parameter space. Similar idea was implemented to construct local error models for POD-TPWL method [31] that showed promising results for well control changes but can be computationally expensive to construct. However, the global error model is used here to avoid further computational complexity to the already expensive PMOR training procedure. Including the reduced order model solution to the input features achieves good accuracy as we take into account the physics of ROMs rather than constructing completely data driven error models as in [7, 6]. Similar to the cases of constructing error maps using ML, we have the single output here corresponding to state error at well gridblock. For such cases, NN proved to be a better model at mapping the complex input-output relation and is faster to train for a single output system rather than using it for the prediction of POD coefficients in global PMOR formulation with high dimensional outputs.

5.5 Case study

5.5.1 Homogeneous reservoir model

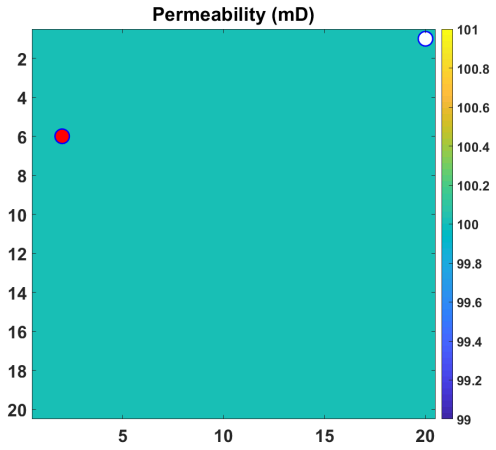
We use the error correction model now in efforts to minimize the solution discrepancies at the well locations. We use the same case study as before for the homogeneous reservoir model and changing producer well locations. In order to construct the error model, we sample randomly other 100 well locations to generate the data set that account for running 100 fine scale simulations. In order for the ML models to capture the underlying behavior of

the system, it requires many data points especially to understand such complex relationships. We realize the computational expense associated with fine scale simulation runs, but better sampling strategies should be the research focus in the future, that can reduce the number of sampling points. Also, the current case is a small model and hence the number of sampling points relative to the size of the problem is high here, however, as we move to the bigger reservoir models, we expect much lower sample points relative to the reservoir size.

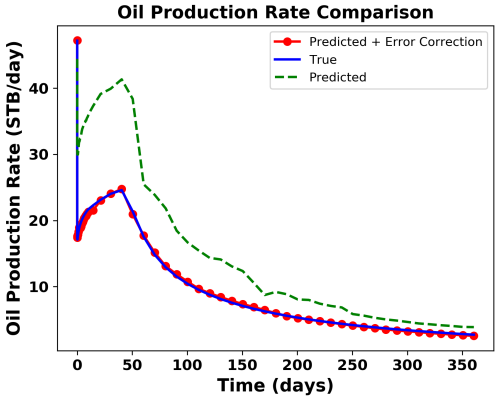
So for the reservoir description as shown in Figure 5.1, we have 100 well configurations used to construct the global PMOR model and 100 other well configurations for training the error correction ML model. Thus, we have 199 test cases in total to test the performance of this methodology. Figures 5.18, 5.19, 5.20 show the comparison of the results obtained by just implementing the non-intrusive global PMOR method, its implementation with the error correction models and the true solution obtained by fine scale simulation.

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

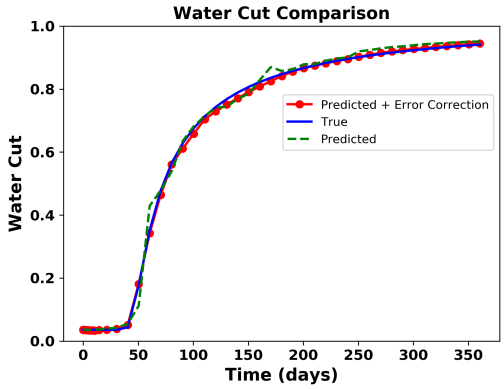
We predicted the solutions for all the test cases and show the errors in prediction for all these cases in Figure 5.21 for oil production rate and water cut. In this figure, the errors shown in red are those obtained by POD coefficient prediction without error correction. These are arranged in increasing order for all the test cases. The blue dots show error after correcting the solutions in the second step for each corresponding red dot. This gives an intuition of the behavior of proposed method for all kinds of test cases. This plot shows that most of the test cases show a very good accuracy after error correction even when the predicted solutions just based on ML estimated POD coefficients have large errors. There are a very few test cases that show errors increasing after correction, that most probably can be attributed to random sampling, and should be solved after efficient sampling techniques.



(a)



(b)



(c)

Figure 5.18: (a) Test case with producer well location at (2,6) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 100 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

The average accuracy for all the test cases can be found in Table 5.3.

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

Table 5.3: Homogeneous reservoir case 1: Average accuracy of oil production rate and water cut for all test samples

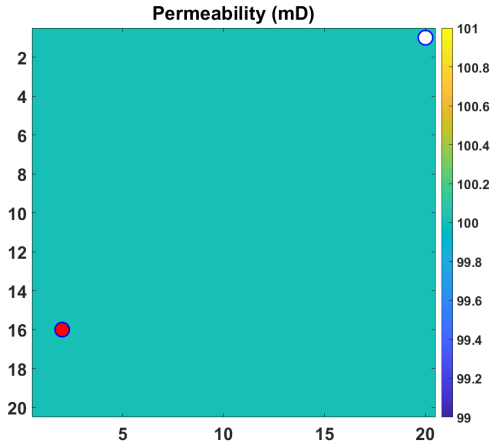
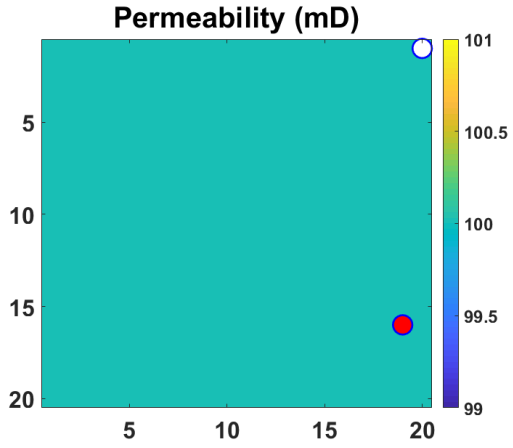
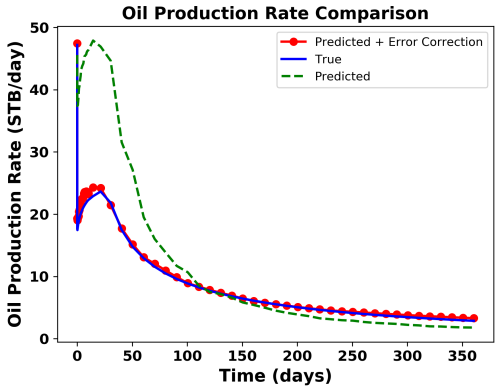


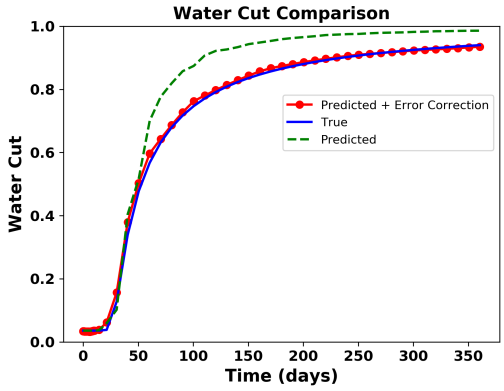
Figure 5.19: (a) Test case with producer well location at (3,16) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)



(a)

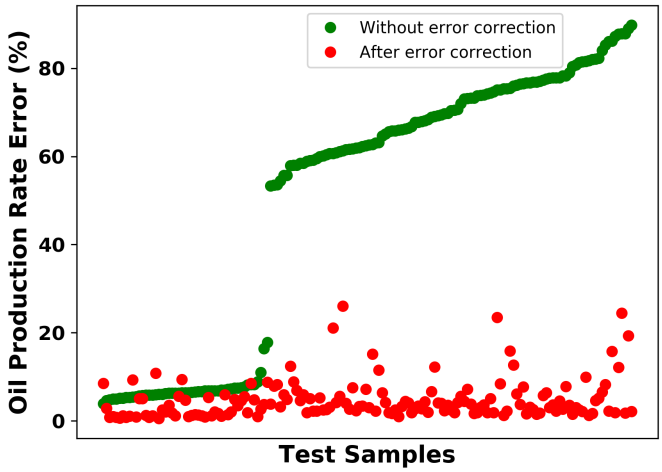


(b)

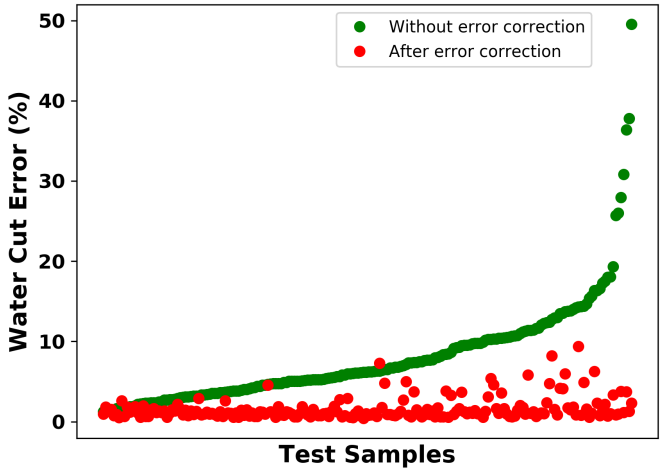


(c)

Figure 5.20: (a) Test case with producer well location at (16,19) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 100 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)



(a)



(b)

Figure 5.21: Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions

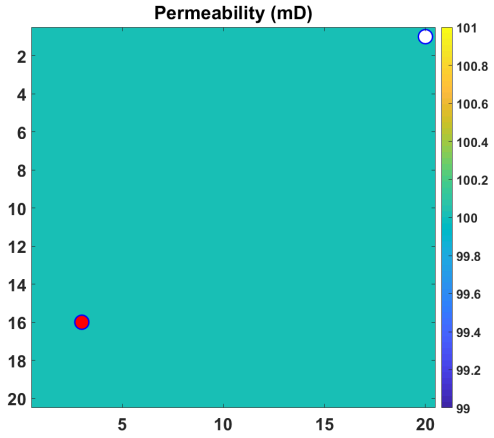
Now, we show the same example, but using lesser number of training sample well locations for predicting basis coefficients and construction of global ROB (this should be a lower quality basis than before) and using same number of training points for error correction model. Since it is challenging to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

Table 5.4 shows the optimum hyperparameters and the corresponding train and test accuracies obtained from the trained Random Forest model to predict the basis coefficients for both pressure and saturation. As can be expected, the train accuracy decreases as the ML model does not have a large data set to learn the physics which eventually also shows a decrease in the test accuracy as compared to the previous test case.

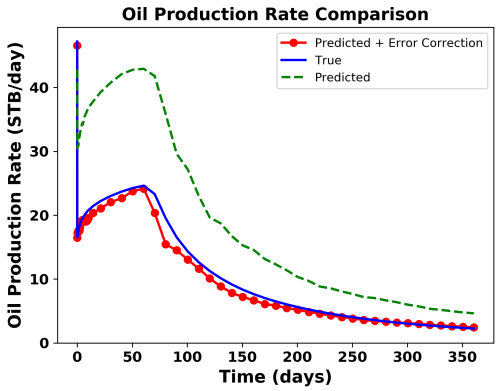
	RF Regression	**Train Accuracy**	**Test Accuracy**
Pressure	N_{fmax}=2, N_l=2	99.55	97.55
Saturation	N_{fmax}=2, N_l=2	98.71	90.79

Table 5.4: Hyperparameters chosen by 5-fold Cross Validation for Random Forest Regressor using 50 training samples

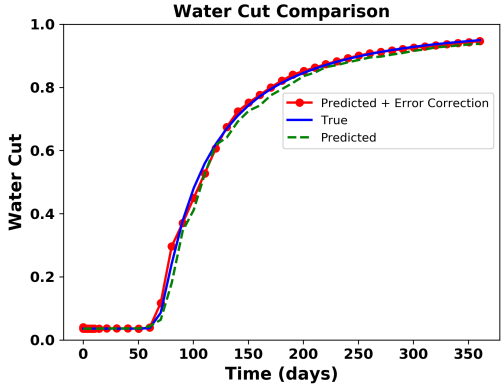
Figures 5.22 and 5.23 show two of the test cases for this example by comparing the oil production rates and water cut. These results show a good agreement in the solutions thus showing the validity of error correction model for the cases when the global basis is not very accurate for the entire parameter domain but still can capture the overall trend of the solutions. However, these observations may not hold true when the quality of global basis is compromised enough to explain the appropriate physics of flow.



(a)



(b)

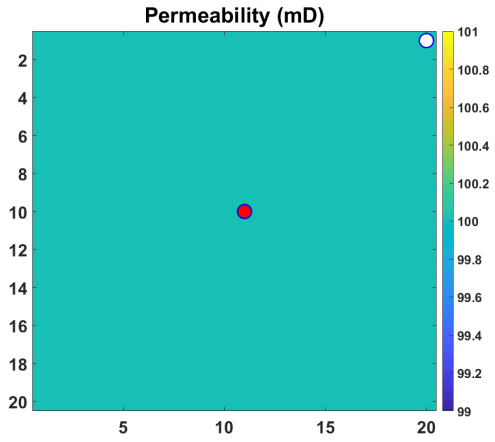


(c)

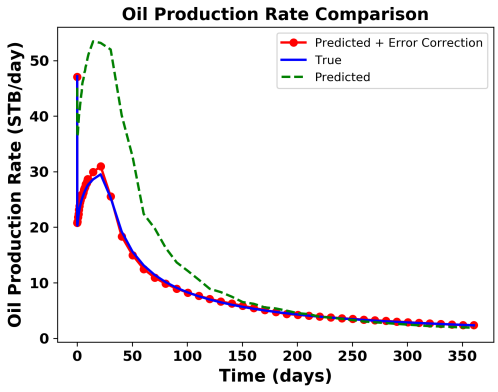
Figure 5.22: (a) Test case with producer well location at (3,16) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 50 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

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

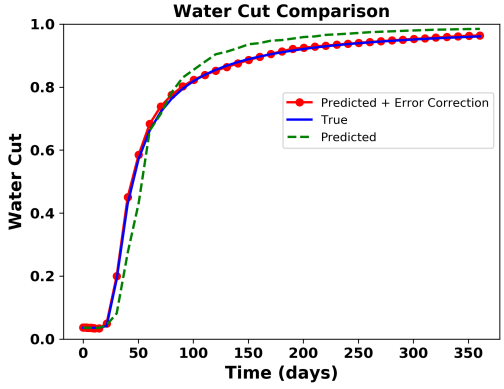
Table 5.5: Homogeneous reservoir case 2: Average accuracy of oil production rate and water cut for all test samples



(a)

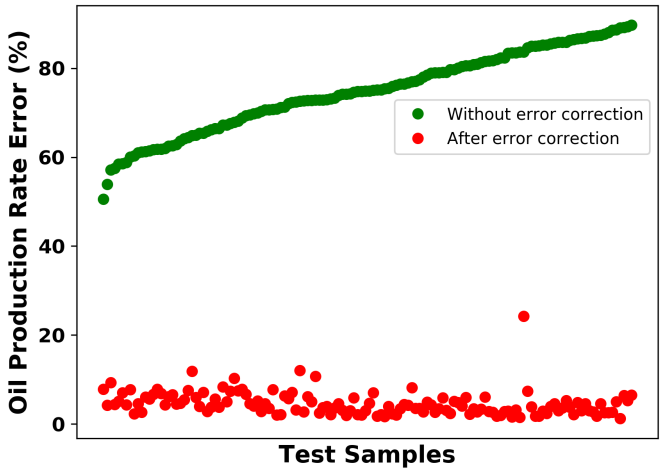


(b)

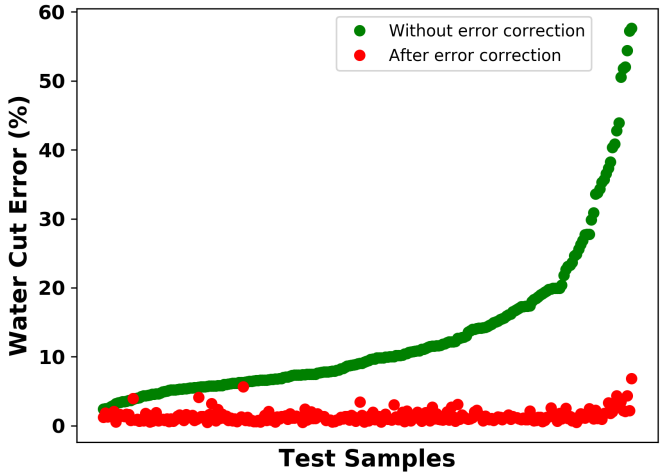


(c)

Figure 5.23: (a) Test case with producer well location at (11,10) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone using 50 samples (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)



(a)



(b)

Figure 5.24: Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions

5.5.2 Heterogeneous reservoir model - single well case

The methodology shows promising results for a small homogeneous reservoir model. Now we apply the same methodology to a heterogeneous reservoir model which is a section of layer 50 of the SPE10 benchmark reservoir. Again, this is a two-phase flow (oil-water) reservoir model with one injector well and one producer well as shown in Figure 5.22. The reservoir model is discretized with a Cartesian grid of size 20 ft $\times$ 20 ft $\times$ 50 ft, and it contains 2500 (50$\times$50) active cells. We neglect the capillary and gravity effects. The initial reservoir pressure is 4200 psi and the initial water saturation is considered 0. The injector and producer are BHP controlled at a constant pressures of 7000 psi and 2500 psi respectively. For the current scope of work, we believe, this model with about 5 orders of range in permeability and an injector and a producer, should be a good case with reasonable complexity to demonstrate the validity of proposed global PMOR strategy. Again we only consider that the producer is the parameter of interest which means the injector location is fixed and the producer changes location in the reservoir. The simulation is run for a duration of 3 years.

For the training of global PMOR model, we consider the producer well locations only on the high permeability gridblocks majority of which lie on the channel connecting the injector and producer. We use all the gridblocks with greater than 10 mD as candidate well locations as shown in Figure 5.23. This is employed to observe the water cut for most of the well locations. Thus, here this global PMOR strategy is employed on a subset of parameter domain. These are around 1000 sample locations out of which we use 200 randomly sampled locations for training the ML based global PMOR model and 100 samples chosen for training the error correction model. The rest of the samples are test cases for which the accuracy of the PMOR model is evaluated.

We show the randomly sampled data set for training the global PMOR ML model in Figure 5.27a. Figure 5.27b shows the cumulative energy of singular values for the given sample set. For the first example, we use 99% energy criteria for pressure basis dimension and 90% energy criteria for saturation basis dimension. Later in the section, we analyze

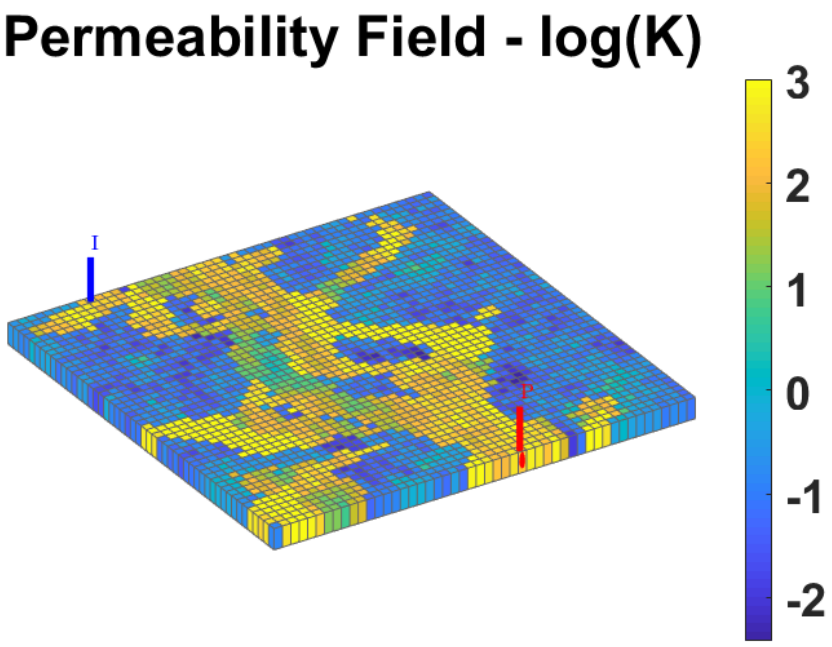


Figure 5.25: Heterogeneous log permeability field (section of SPE10 model layer 50) with one producer and one injector

the accuracy of the method with changing energy criteria for these states. We observe the steep cumulative energy curve for pressure and a more concave curve for saturation. This is expected since all the well locations are just considered on the high permeability area of the reservoir. The observations about energy curves follows the same explanation as that of the homogeneous case if we consider well locations throughout the spatial domain of reservoir.

Since, we consider the heterogeneous case here, an additional feature corresponding to the well block permeability is added which was not included for the homogeneous model before. We first consider a test case as shown in Figure 5.28 with producer well location at (28,50). In Figures 5.29 and 5.30, we show pressure and saturation solutions obtained by predicted POD basis coefficients with the true solutions for two of the test cases at different times. We also show the error in pressure solution in Figure 5.31 to verify the same observations as in the homogeneous case where the maximum error is obtained at producer well locations.

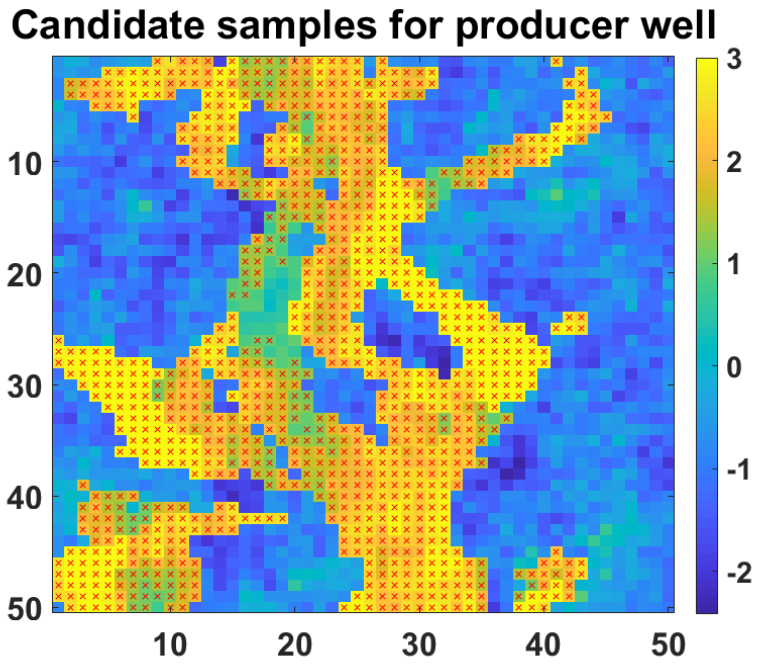
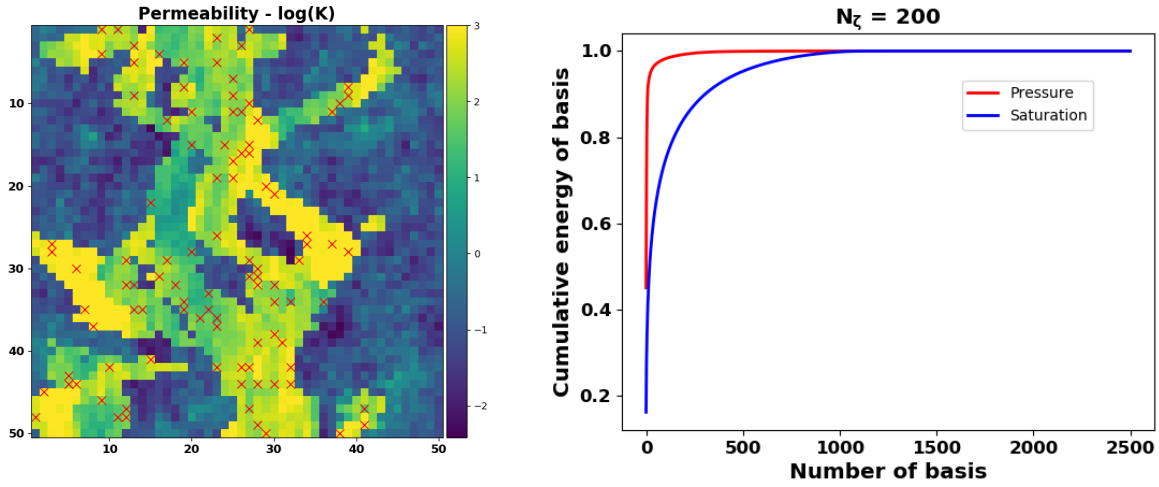


Figure 5.26: Samples considered as potential well locations shown by red crosses. Injector location is fixed at location (10,1) (not shown here) and hence not sampled.

By observing the saturation solutions, we can see that there are small saturation errors near the waterfront. The time at which all the solutions are reported are chosen to be somewhere in the middle and at the end of simulation run time. As can be observed, the error is very small throughout the spatial domain and maximum at the well location which is the same observation as before.

Here, we also show the comparison of basis coefficients for pressure and saturation as predicted by the ML algorithm at the same timesteps to determine the validity of ML models. We can see that the ML model produces a very small error in prediction of the coefficients for some of the pressure and saturation basis. Hence to check if the discrepancy in the predicted solution is due to the ML model or the quality of basis, we also compare the plots of relative error in ML predicted solution and the orthogonal projection error computed using equation (5.13). This comparison for pressure in Figure 5.34 shows that the orthogonal projection error is maximum at the well location. Hence majority of the error in the output quantity



(a) (b)

Figure 5.27: (a) Training samples randomly chosen for basis coefficient prediction (b) Cumulative energy of eigenvalues for pressure (red) and saturation basis (blue)

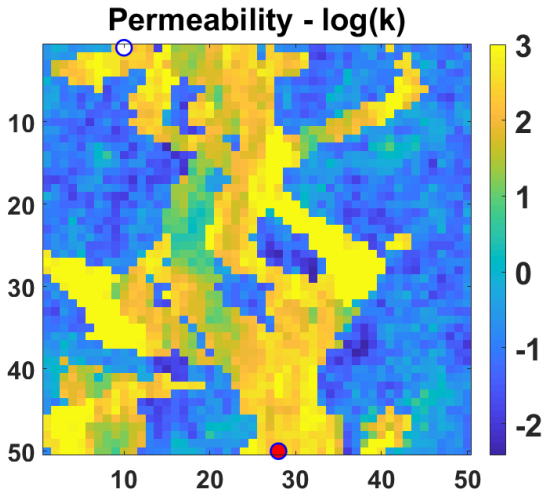
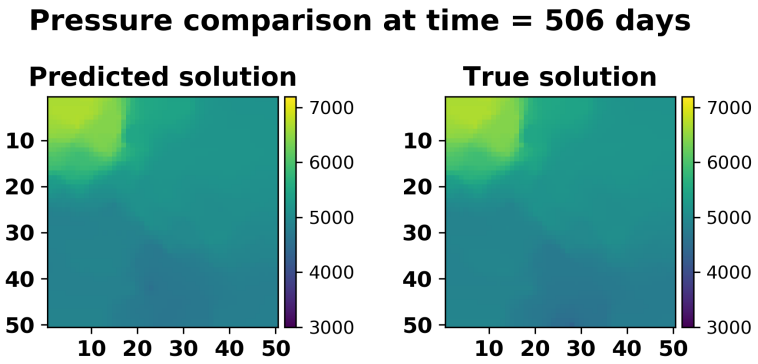
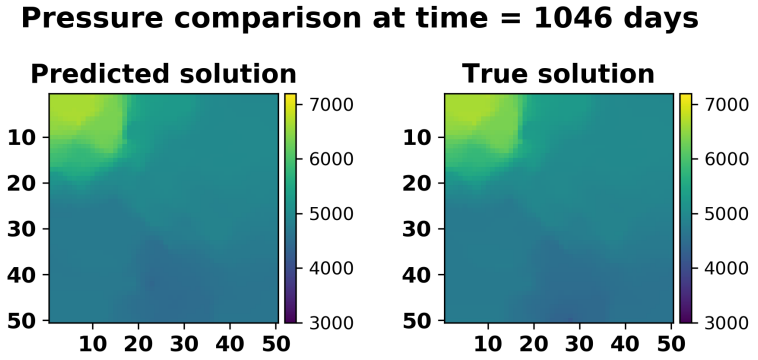


Figure 5.28: Test case with producer well at gridblock (28,50)

of interest that is dependent on well block states is expected due to the quality of basis. Thus, it is in line with our discussion for homogeneous case. As we see that it challenging to find a single global basis that is representative of the controllability properties at all the well locations, we use the error correction formulation to adjust for the error in state solutions



(a)

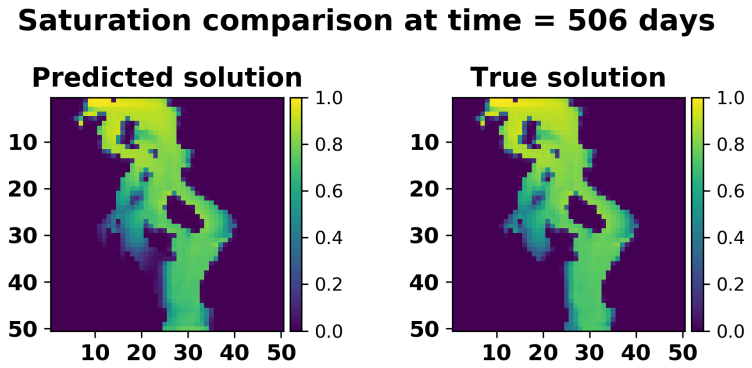


(b)

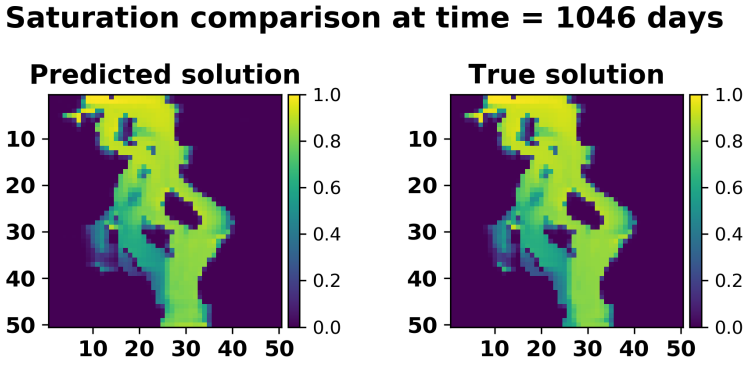
Figure 5.29: Pressure solution comparison at (a) Time = 506 days and (b) Time = 1026 days

at the gridblock. Figures 5.32, 5.33 and 5.34 show the test cases for different well locations comparing the oil production rates and watercut. We can see that the corrected solutions for some cases like test case 2 may not be very accurate but still shows much improvement after error correction. Similarly, for test case 3, where we see a lot of irregularities in the solutions obtained just by coefficient prediction, show significant improvement after error correction.

For a better intuition about the results for this case study, we show a plot of accuracy in prediction for oil production rate and water cut for all the test cases (Figure 5.38). The samples are arranged in increasing order of errors obtained just from predicted basis coefficients and for each sample we show the error after employing the error correction model. It can be observed that after error correction, the accuracy of predictions in oil production rate and water cut increase significantly. There are some test cases that produces higher error after



(a)



(b)

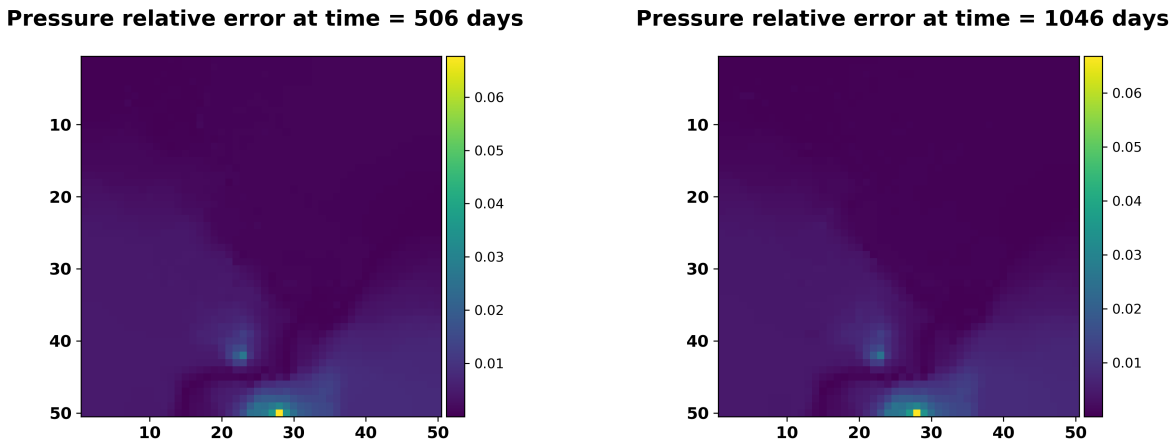
Figure 5.30: Saturation solution comparison at (a) Time = 506 days and (b) Time = 1026 days

correction which were found to be the cases where no water cut was observed and very low oil production rates. A better sampling strategy can solve this issue. The average accuracy for all test cases after error correction is shown in Table 5.6.

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

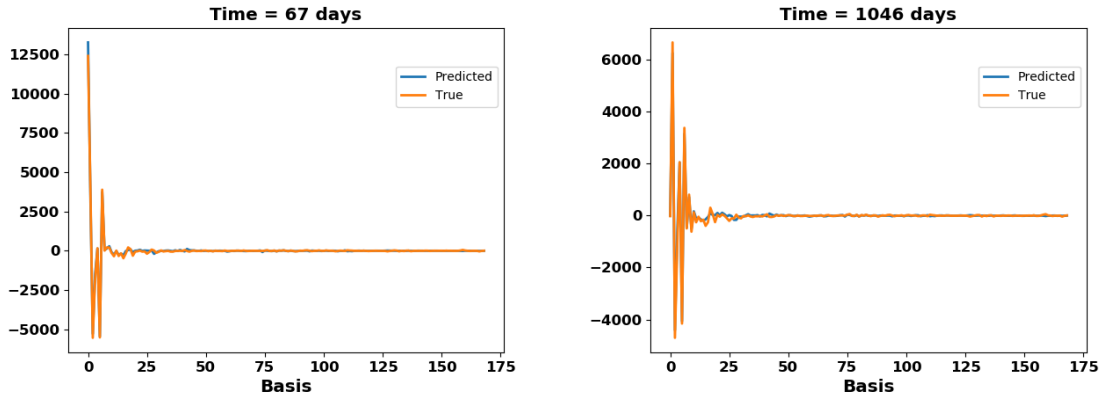
Table 5.6: Heterogeneous reservoir case 1: Average accuracy of oil production rate and water cut for all test samples

Table 5.7 shows the time comparison in seconds for these cases between fine scale simu-



(a) (b)

Figure 5.31: Relative error in pressure at time = (a) 506 days (b) 1026 days

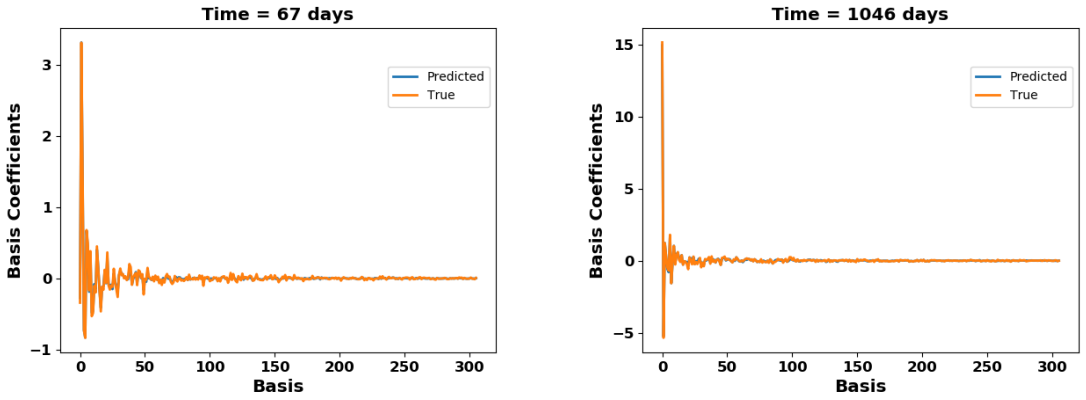


(a) (b)

Figure 5.32: True and ML predicted pressure basis coefficient comparison at time = (a) 506 days and (b) 1026 days

lations run in matlab and the proposed PMOR model with error correction, both run on a local 8 core machine. We see the speedups of about 50× for these test cases. This does not include time required to train the PMOR models.

It is also important to decide on the energy criteria used for choosing the basis dimensions, since it is not trivial to make this decision a priori. As discussed before, there is a trade-off in accuracy and computational time between the decision to select the number of training



(a) (b)

Figure 5.33: True and ML predicted pressure basis coefficient comparison at time = (a) 506 days and (b) 1026 days

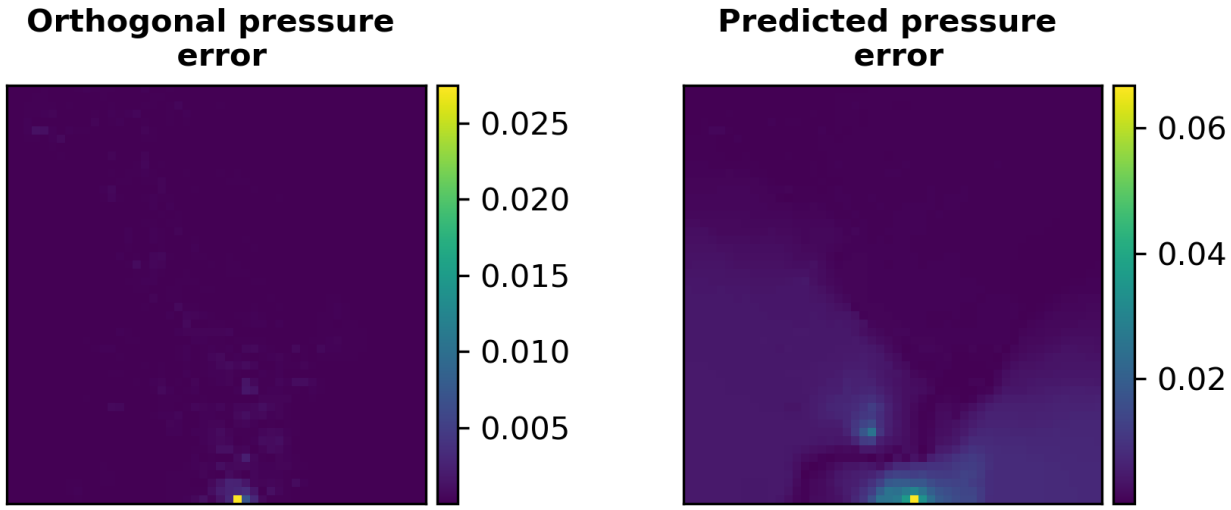
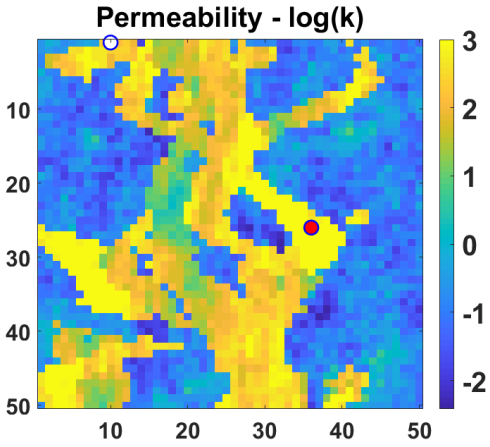
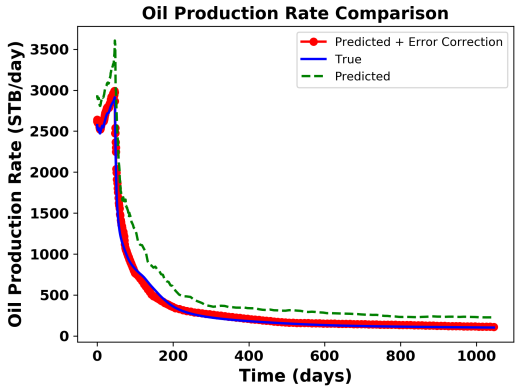


Figure 5.34: Comparison of relative error in pressure due to orthogonal projection (left) and ML predicted solution (right)

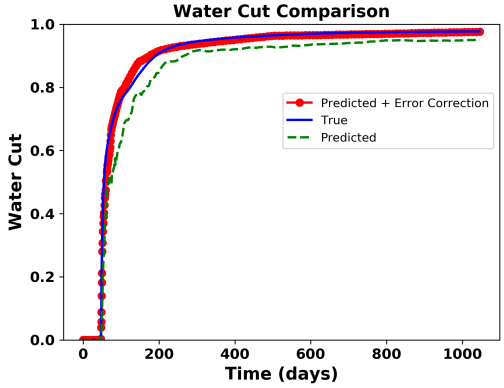
samples and the basis dimensions. To get another perspective, we also show a box plot of the pressure and saturation solution accuracies as predicted from the basis coefficients for a range of energy criteria (Figure 5.39). Each box is constructed with 10 different training sets. As can be expected, with increasing number of basis, we see an increase in the accuracies of solutions. However, we also see a drastic reduction in the number of basis with decreasing



(a)



(b)



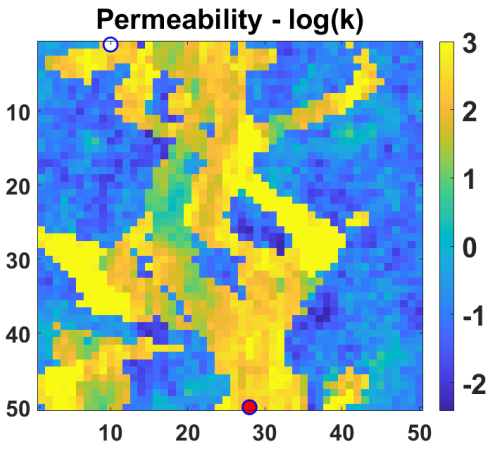
(c)

Figure 5.35: (a) Test case 1 with producer well location at (36,26) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

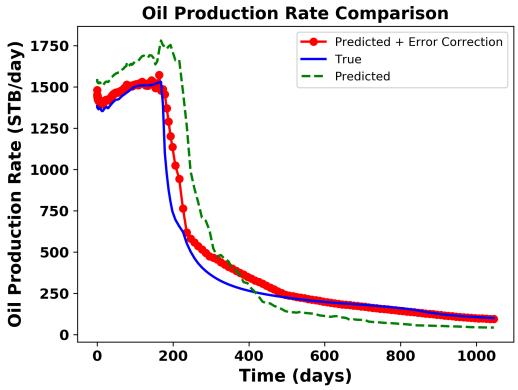
	Fine scale simulation	**PMOR + Error correction**
Test Case 1	65 seconds	1.3 seconds
Test Case 2	52 seconds	1 seconds

Table 5.7: Time (seconds) comparison for the two test cases between fine scale simulation and reduced order model with error correction

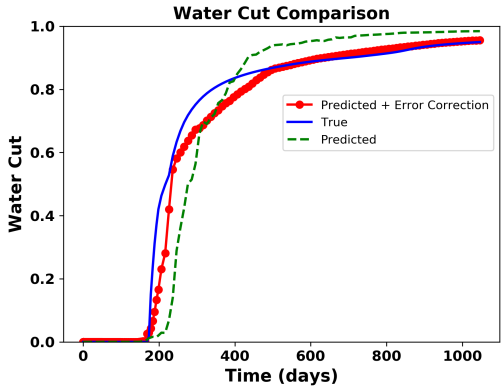
percentage of singular values selected, especially for pressure. The accuracy shown here is the that when compared at all the gridblocks, which is not very different in magnitude between



(a)



(b)

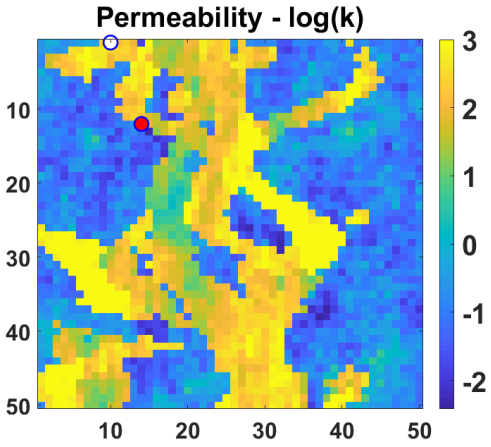


(c)

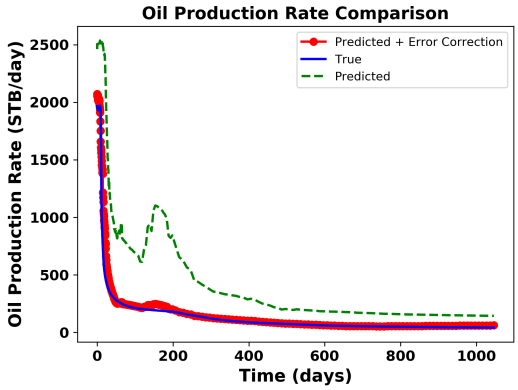
Figure 5.36: (a) Test case 2 with producer well location at (28,50) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

different basis dimensions. But it was observed that the accuracy of solutions at the well gridblock increases significantly with increasing basis dimension.

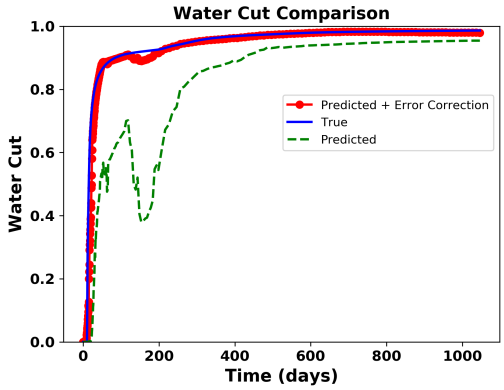
The case shown above correspond to pressure and saturation basis dimension selected using energy criteria of 99% and 90% respectively. Thus, we now evaluate, if the error correction model can also produce reasonable results if lower energy basis dimensions criteria is chosen. For this reason we consider another case with pressure and saturation basis dimension selected using energy criteria of 95% and 80% respectively (Note that these energy



(a)



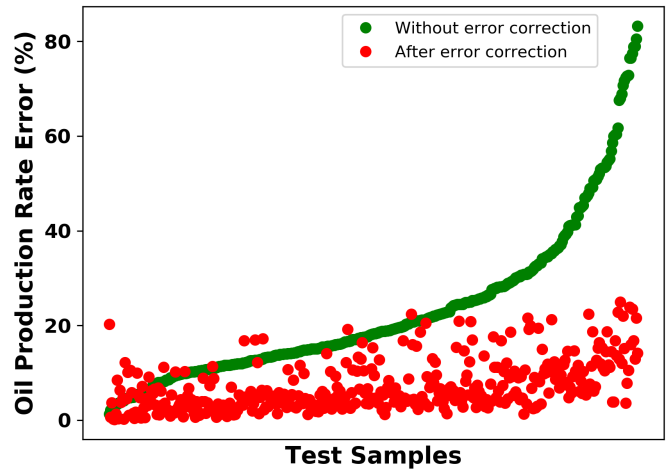
(b)



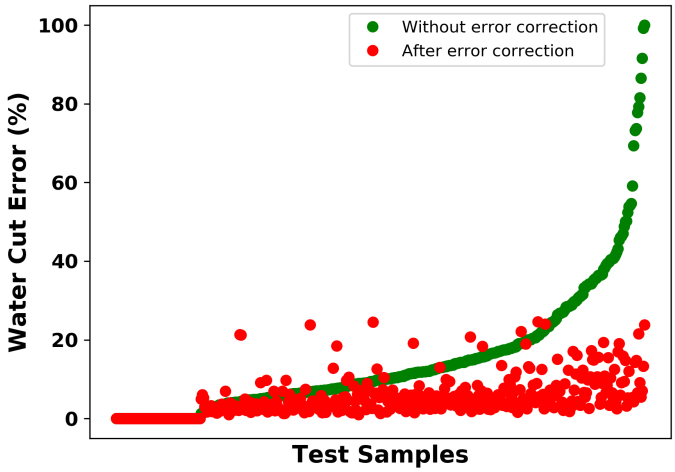
(c)

Figure 5.37: (a) Test case 3 with producer well location at (14,12) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

are chosen such that they capture most of the high energy singular values as shown in Figure 5.27b. If we choose even lower, the predicted basis coefficients are not expected to capture the physics of states behavior adequately and hence lead to inaccurate solutions). Figures 5.40 and 5.41 show two test cases comparing the oil production rate and water cut before and after error correction. In general, the solutions are a good match after error correction. The error in prediction, for all the test samples, before and after error correction is shown in Figures 5.42a and 5.42b. It can be seen that the error from predicted coefficients now increase



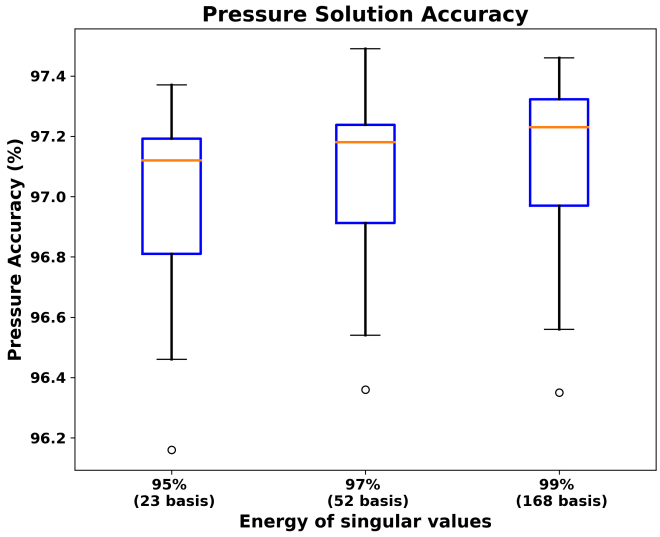
(a)



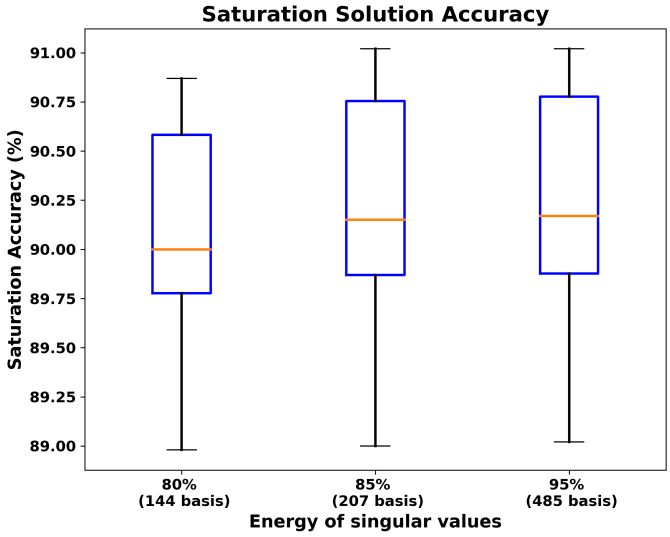
(b)

Figure 5.38: Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions

at a higher rate which is expected due to much lower basis dimensions. However, after error correction, most of the test cases show a much better accuracies. The average accuracy for oil production rate and water cut is shown in Table 5.8. It is lower than that predicted in the case of higher basis dimension (Table 5.6), but still not significantly less. Thus, it is up to one to choose appropriate basis dimensions such that significant computational speed ups



(a)

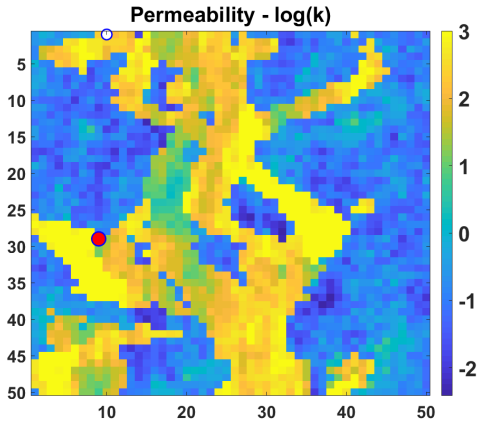


(b)

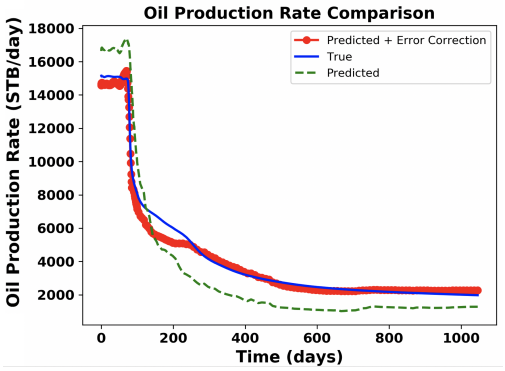
Figure 5.39: Box plot of accuracies in pressure and saturation using predicted POD coefficients for increasing basis dimensions

are obtained at the cost of losing accuracy.

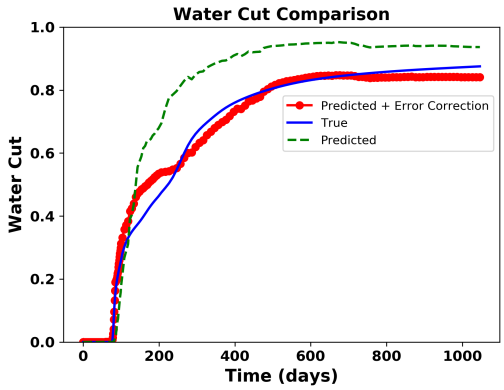
Table 5.9 shows the time comparison in seconds for these cases between fine scale simulations run and the proposed PMOR model with error correction, both run on a local 8 core machine. We see the speedups of about 100× for these test cases since the model outputs



(a)



(b)



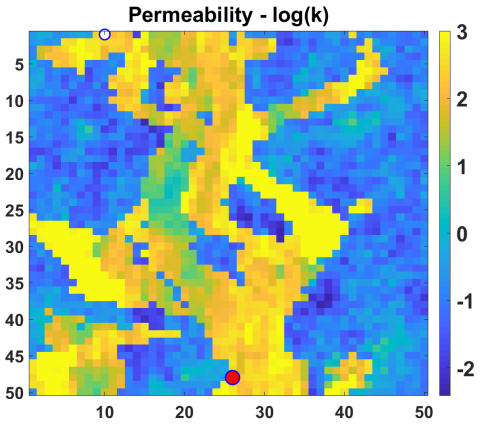
(c)

Figure 5.40: (a) Test case 1 with producer well location at (9,29) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

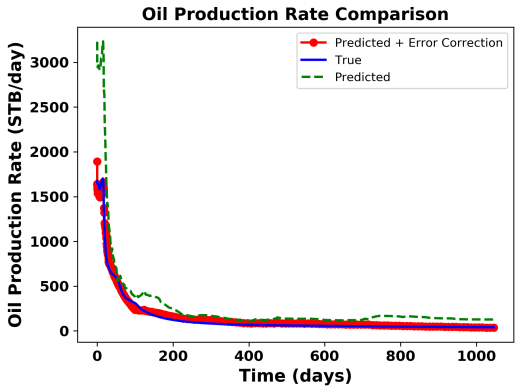
	Average Accuracy (%)
Oil production rate	91.74
Water cut	94.6

Table 5.8: Heterogeneous reservoir case 2: Average accuracy of oil production rate and water cut for all test samples

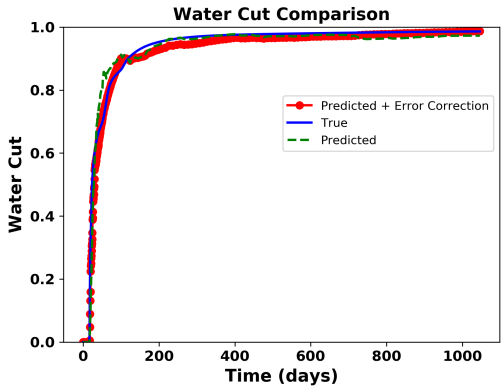
have lower dimensions and hence less complex.



(a)



(b)

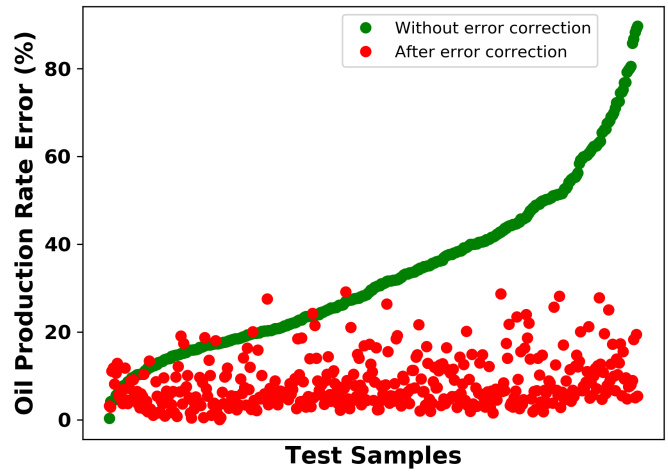


(c)

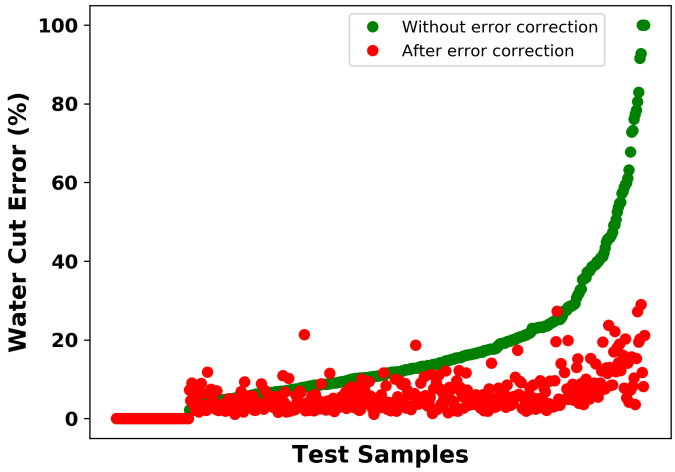
Figure 5.41: (a) Test case 2 with producer well location at (26,48) (b) Comparison of oil production rate and (c) Comparison of water cut, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

	Fine scale simulation	**PMOR + Error correction**
Test Case 1	56 seconds	0.7 seconds
Test Case 2	46 seconds	0.5 seconds

Table 5.9: Time (seconds) comparison for the two test cases (lower dimensional basis) between fine scale simulation and reduced order model with error correction



(a)



(b)

Figure 5.42: Error in prediction of (a) Oil Production Rate (b) Water Cut, for all the test cases before and after the error correction of the solutions

5.5.3 Heterogeneous reservoir models - multiple wells case

The results for a single producer and injector shows the validity of the proposed methodology, but for practical implementation, we need to see if this method shows such promising results. This is mainly the future work that needs to be addressed, but we show a simple

case study as a beginning for future directions. We know from the MOR literature that, MOR techniques when used within optimization workflows, that require many simulation runs corresponding to new parameter sets during each iteration, the training set of parameters for MOR development is designed such that the parameters are not drastically different from those expected to be encountered in the test set. For well locations, as we see from the above results, it is very difficult to determine how different is one location to the other, especially for heterogeneous reservoirs like the section of SPE10 model considered. Also, the parameter domain for well locations increase significantly for large reservoir models, in that each grid block in the reservoir is a potential well location. This becomes even more challenging as we have more wells where we can have different combinations of well locations across all the gridblocks. In these scenarios, it just becomes impractical to devise PMOR strategies that can satisfy all possible combinations of well locations in the reservoir model. Thus, training a PMOR model must be tied to the optimization strategy to be used and training samples must be designed accordingly. For example, well location optimization being a very challenging problem given numerous possible well configurations, researchers resort to quick evaluations or diagnostics 11of the reservoir to rule out many areas in the reservoir as potential well locations [45, 104]. And then, given this information, complex optimization routines using simulation runs can be sought for a systematic decision making process.

The method currently proposed, is suitable for any well location in the reservoir, and hence further adds to the training effort by collecting more samples. But in the future, when this workflow is applied with optimization routines, the training strategy should be designed keeping the optimization strategy in mind. As a preliminary idea, we show a simple case for the same model as considered for single well case, but now we consider 2 producers and 1 injector. We keep the injector location fixed and producers can change locations. With 2500 gridblocks and injector location fixed, we can have $2498 \times 2498 = 6,240,004$ number of producer combinations. If both the producers have the same well properties

and produced with the same BHP schedule, i.e. both the well are equivalent, we have $6,240,004/2 = 3,120,002$ different well configurations. Thus, the optimization strategies are sought such that this possible combinations are reduced a priori by setting some constraints. It is thus impractical to develop the proposed PMOR strategy that is expected to work for all these combinations. We only consider a simplistic case here as a beginning to validate the method for multiple wells. Figure 5.43 shows the reservoir model with an injector and 2 regions (shown in red) where producer 1 and producer 2 locations are randomly sampled. We choose 20 samples in each region and hence we can have 400 combinations of producer 1 and producer 2 pairs. The BHP schedules for both producers are kept the same and have identical well properties. The simulation is ran for a duration of 500 days. ML model is trained to predict the POD basis coefficients for pressure and saturation using 50 sample points randomly chosen out of 400. 100 training samples are randomly chosen for error correction ANN model. Note that, the change here as compared to single well formulation lies in the input featured defined for ML models. Now the inputs include geometric and flow diagnostics based features for both the wells in ML model to predict the basis coefficients and the error correction model has the same features representing both producer wells along with the reduced order states and pore volumes injected.

Figures 5.44 and 5.45 shows two of the test cases with different producer 1 and 2 pairs than used in training. For test case 1, the results show a good accuracy after error correction which shows erratic behavior for producer 2 using just the predicted basis coefficients. Similarly, the test case 2 is an example where even without error correction the results are a good match to the true solutions, which improve further after error correction. This shows that even though the global basis may not be a good quality basis for many parameters, we can employ error correction model to account for the solution discrepancies. Figure 5.46 shows the test accuracies for all the 250 test samples of oil production rates and water cut for both the producers. As we see, the accuracy after error correction improves significantly for many test cases. There are very few test cases again, with higher error after error correction that

we deem as a result of random sampling procedure.

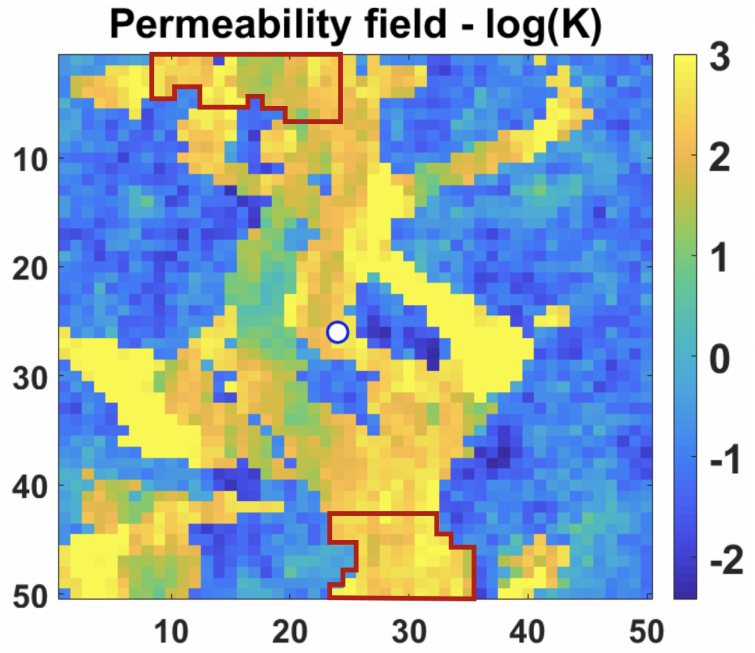


Figure 5.43: Reservoir model with on injector at location (24,26) and red regions showing the regions from where producer 1 (top of the reservoir) and producer 2 (bottom of the reservoir) locations are sampled

Table 5.10 shows the comparison of time between fine scale simulation and the proposed PMOR model with error correction. The speedup obtained for these cases was about 100×. Note that these times are just for the test runs. It does not include the training time, which must be the area of focus for reduction in the future.

	Fine scale simulation	**PMOR + Error correction**
Test Case 1	102 seconds	1 second
Test Case 2	90 seconds	0.8 seconds

Table 5.10: Time (seconds) comparison for the two test cases between fine scale simulation and reduced order model with error correction

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

5.5.4 Advantages and Limitations

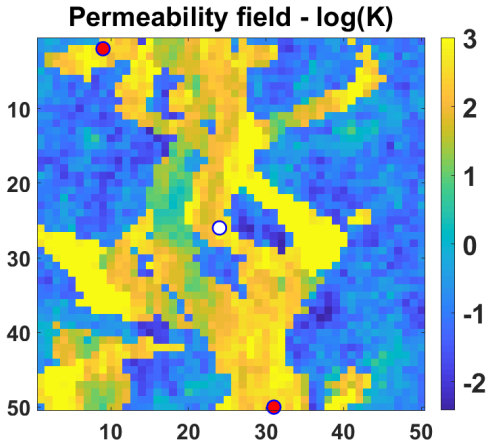
The method described above requires many fine scale simulations to train the ML models but can easily be used with high performance computing facilities. With the advent of parallel computing resources and cloud architecture, the training procedure can be extremely fast. It is true that one may argue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

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

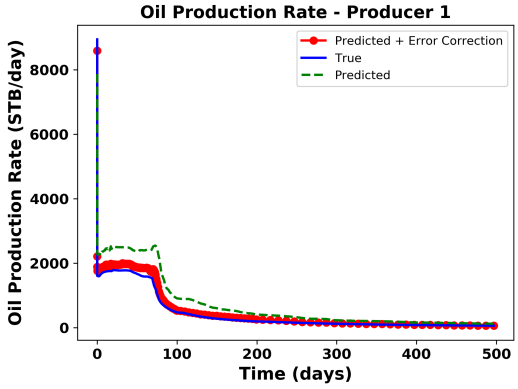
5.6 Summary

In this section, we introduced a non-intrusive global parametric model order reduction method for changing well locations based on machine learning techniques. This method requires just the pressure and saturation states at different time steps and a global basis is constructed from the representative well locations in the reservoir. The machine learning model is trained to learn the coefficients of the POD basis for the well locations as defined by a set of geometric and flow diagnostics based features. We also introduce time as a feature to account for temporal behavior of the coefficients. The trained ML model is then used to predict the POD coefficients of the states for the new well locations. The ML model used here is Random Forest, which is much faster to train compared to other algorithms like ANN especially for large dimensional outputs, and has the ability to define complex input-output

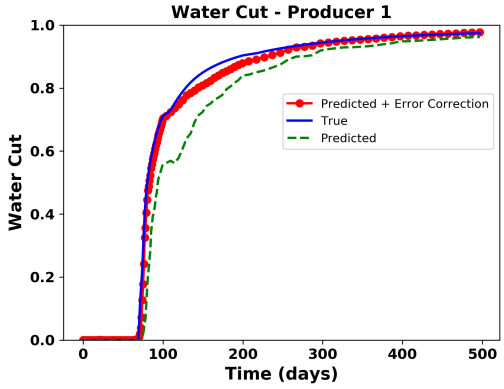
relations. ML model predicts the basis coefficients extremely fast with speed ups of about 50×(for high dimensional basis) - 100× (for low dimensional basis). However, the ML model could capture the overall solution trend but he states predicted at the well block showed a significant bias as can be attributed to the controllability theory. But due to a systematic discrepancy in the solution, in that it captures the solution behavior, we considered fixing this discrepancy through error correction models. These error correction model is a ML based model that captures the state disparity given the reduced model solutions. ANN was used in this case which is faster to train due to this being a single output model and proved to reduced the error of prediction drastically for most test cases. The proposed workflow has a limitation in that it requires many training samples to define the global parameter space. This should be looked in the future research using efficient sampling techniques, that can reduce the number of samples required significantly and using local parameter regions to develop multiple local models in a similar way as proposed. We also note that, as mentioned in the case study for multiple wells, the proposed method must be tied with the optimization strategy used that would help design a better training strategy, as it is impractical to expect the model to be accurate for all possible well configurations in the reservoir.



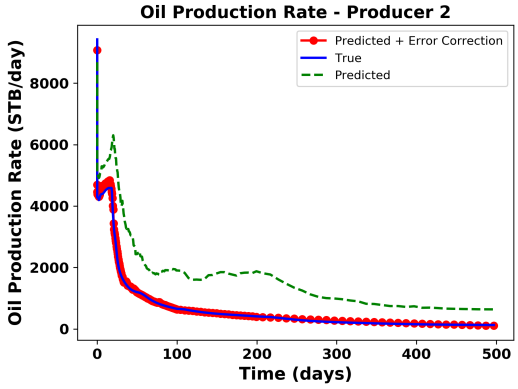
(a)



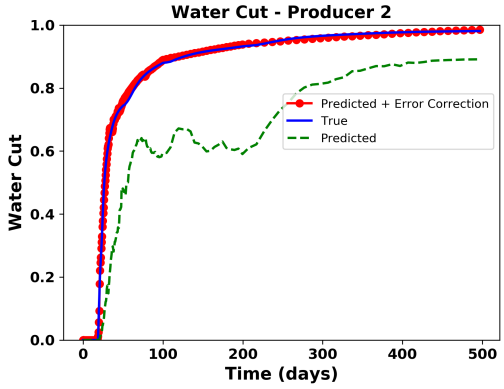
(b)



(c)

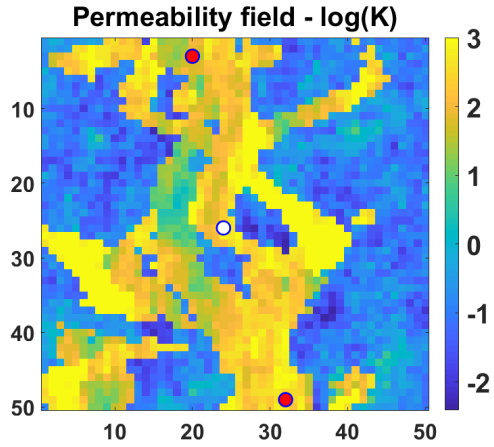


(d)

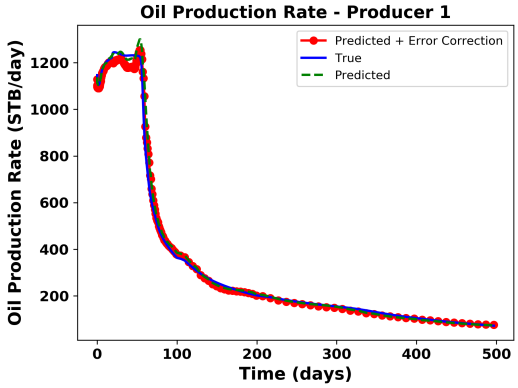


(e)

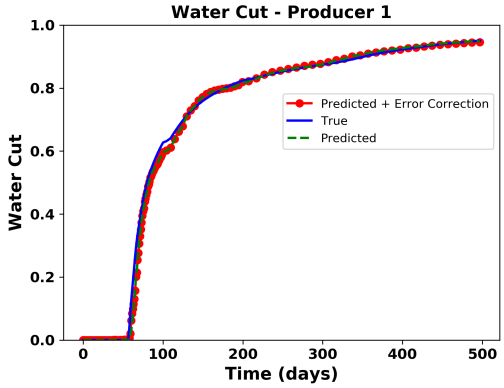
Figure 5.44: (a) Test case 1 with producer 1 at (9,2) and producer 2 at (31,50) (b) Comparison of oil production rate for producer 1, (c) Comparison of water cut for producer 1, (d) Comparison of oil production rate for producer 2, (e) Comparison of water cut for producer 2, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)



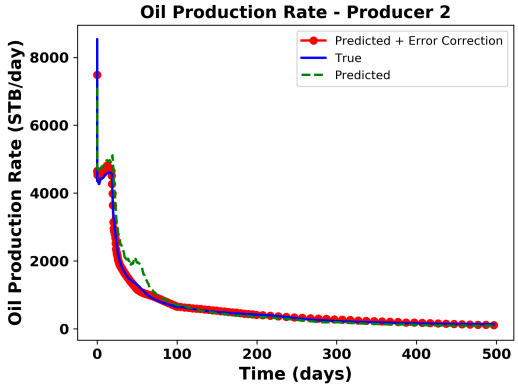
(a)



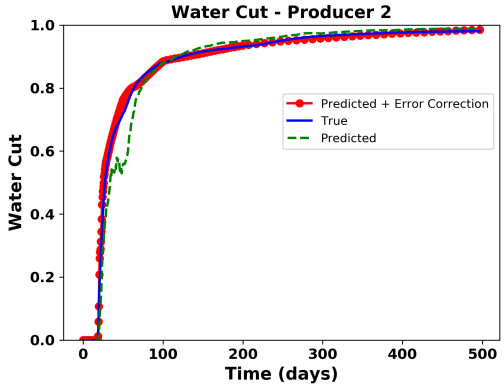
(b)



(c)

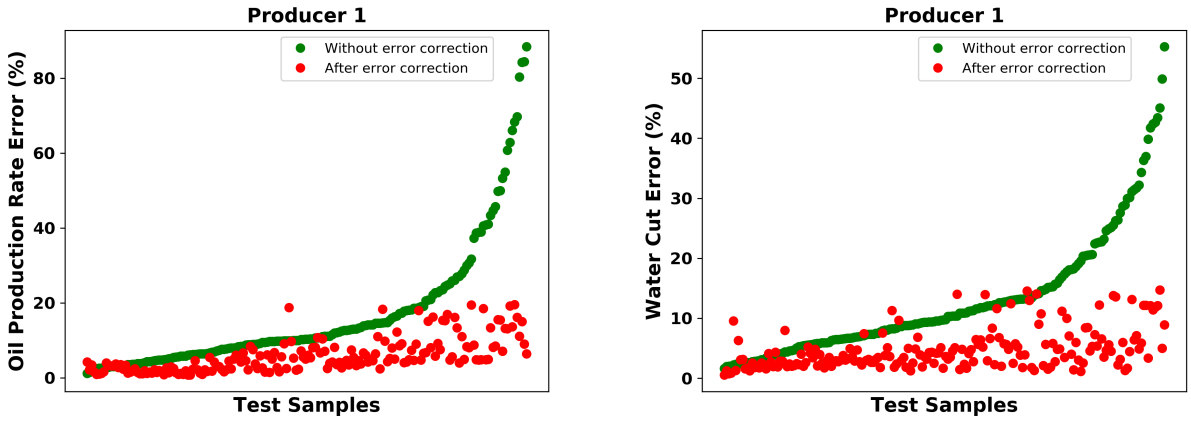


(d)



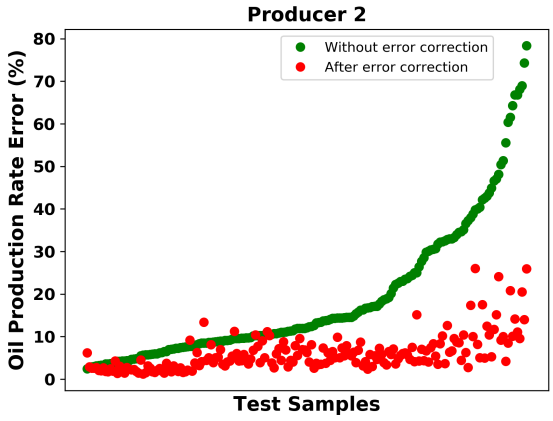
(e)

Figure 5.45: (a) Test case 2 with producer 1 at (20,3) and producer 2 at (32,49) (b) Comparison of oil production rate for producer 1, (c) Comparison of water cut for producer 1, (d) Comparison of oil production rate for producer 2, (e) Comparison of water cut for producer 2, predicted using global PMOR method alone (dotted green line) and after implementation of error correction model (red circled line) with the true solution (blue line)

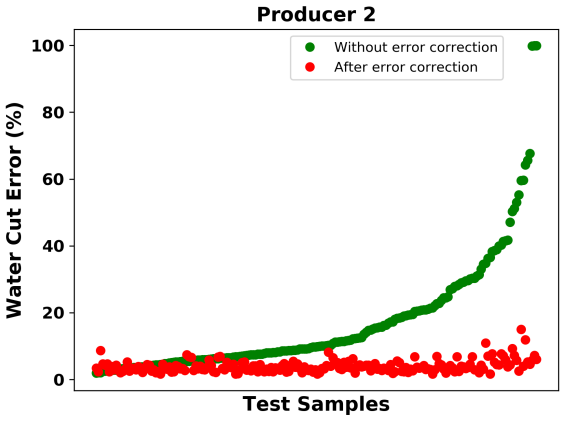


(a)

(b)



(c)



(d)

Figure 5.46: Error in prediction of (a) Oil Production Rate and (b) Water Cut for producer 1, (c) Oil Production Rate and (d) Water Cut for producer 2, for all the test cases before and after the error correction of the solutions

6. CONCLUSIONS AND FUTURE WORK

In this research work, we have tried to address reduced order modeling procedures for the optimization step in reservoir management workflows. This optimization step includes well control and well placement optimization problems, which are computationally very expensive to accomplish using fine scale simulations. Thus, our focus here is the development of fast and reliable projection based reduced order models for quick evaluation of simulations during optimization procedures. The reduced order models are represented as parametric models to account for different system parameters, namely, well controls for well control optimization and well indices for well placement optimization. All the PMOR techniques proposed are based on Proper Orthogonal Decomposition that projects a high dimensional state space model to a subspace of much lower dimensions and thus solves the system in this reduced space much faster. We summarize our findings in this work below.

6.1 PMOR for well control optimization

In this work, we develop a global parametric model order reduction (PMOR) technique using Proper Orthogonal Decomposition - Discrete Empirical Interpolation Method (POD-DEIM), which is applied to a novel control optimization procedure that takes into account various polynomial control strategies. The POD-DEIM workflow for changing well controls was introduced in previous work ([35]) but lacked a full scale implementation in an optimization procedure which was accomplished here. The proposed methodology takes into advantage the optimization procedure itself to train the global reduced order model. Since, this is a global PMOR technique, we just train the model once with representative BHP controls of the wells and use it for all the well controls during optimization.

We begin our work by first introducing the control optimization framework to be used. For that we compare various control parameterization techniques like conventional Piecewise constant (PWC) approximation of control, cubic spline approximations and, newly

introduced here, Chebyshev polynomial controls. The motivation behind using polynomial controls was for a practical implementation of control schedule which is smooth as compared to the bang-bang type control procedure in PWC approximation. For optimization performance evaluation, we use both gradient based interior point optimization (IPOPT) method and gradient free particle swarm optimization (PSO) method. Chebyshev approximation procedure proved to yield the maximum net present value (NPV) for the case study used which we believe is because of the better parameter search obtained with orthogonal basis functions in Chebyshev polynomials.

All these control strategies were then tied with the POD-DEIM model order reduction technique to be implemented during optimization procedure. We introduced a novel training strategy where we use a swarm of BHP controls as the first step in optimization procedure and also to train the reduced model using snapshot of states collected from this swarm. In the next optimization iteration, we use the best swarm for IPOPT based optimization or continue with a new swarm for PSO based optimization, but now the solutions are evaluated using the trained POD-DEIM model. We could achieve a time reduction to 50% for the entire optimization procedure while maintaining the accuracy of fine scale simulations to a good extent. The PMOR strategy proved to be the most accurate with proposed polynomial controls, especially Chebyshev polynomial controls, at providing optimal control profiles and NPV values very close to those obtained using fine scale optimization procedure. Thus, polynomial control strategy, being a practical optimization control procedure, also proved to capture the dynamics of state evolution, for PMOR training, better than traditional control procedures.

We also introduced our work in Bayesian formulation for solving reduced order models within a probabilistic framework for changing well controls. We did some preliminary work to validate the idea of sampling several realizations of basis functions to obtain less accurate but a number of inexpensive solutions to account for uncertainties in reduced model solutions. The posterior formulations is based on residual information and we use Gibbs sampling to

sample the posterior. The proposed method showed promising preliminary results on small case studies and need full scale implementation after addressing some issues.

6.2 PMOR for changing well location during well location optimization

The next part of my research, which was the major focus of my work, was developing PMOR procedure for changing well locations observed during well placement optimization. There has not been any development, to the best of our knowledge, for construction of projection based MOR workflow for well location optimization, which is a computationally very demanding problem using fine scale simulations during optimization procedure. This was the main motivation behind this work. In the first part, a machine learning based local PMOR workflow using POD was introduced. The proposed method was based on the observations that a pre-exisiting reduced order basis (ROB) produces a small error in the quantity of interest, here oil production rate, for a new well location and hence can be used for that location. Thus, a ML based formulation was proposed to adaptively select a pre-existing basis from a dictionary of ROBs that can predict the solution at a new well location within a desired tolerance. The ML formulation also requires defining features corresponding to a new well location. These were defined based on some analysis using simple case studies and are described as geometric and physics based features. The physics based features are derived from the concept of flow diagnostics that evaluate some properties describing flow of fluids like Time of Flight (TOF) and Lorenz coefficient, computed very quickly. The proposed formulation can be set as a regression problem to quantify the QoI errors at new well locations or as a classification problem for qualitative description of a database of basis. The case studies show two phase flow problems for an injector - producer pair with changing producer location. The methodology when applied to such system showed promising results using Artificial Neural Network (ANN) models as compared to Random Forest (RF) models. In general, posing the problem as classification problem showed better accuracy in determining good quality basis selection for new well locations. Since, this method is based on selecting basis obtained from single well location, the basis dimension is

very low and hence lead to very fast online procedure. But a major limitation to this method in its current form was its applicability to smaller simulation times where water front does not reach the producers.

In the next part of research, in order to work on the limitations of the proposed local PMOR technique, we introduced a non-intrusive global PMOR technique based on Machine Learning framework. This technique is based on predicting the POD basis coefficients of pressure and saturation states given a new well location. The global basis is constructed from randomly chosen training sample well locations. Thus, a ML model is trained to learn the relationship between the input features corresponding to new well locations and the multi-dimensional output of basis coefficients. The input features are defined based on the same geometric and flow diagnostic concepts for new well locations. We also introduce time as a feature to account for temporal evolution of basis coefficients, which in intrusive methods is done by projecting the non-linear functions at each simulation time step. The application of this workflow on two phase flow cases, shows that the ROM captures the solution trend but produced a discrepancy in the well block states. This is expected from the controllability property explanation for well locations. In order to minimize this discrepancy, we introduced a error correction model that predicts the error in well block states given the reduced model solution. This showed a significant reduction in the QoI errors for most of the well locations in the reservoir.

Thus, we introduce a novel PMOR procedure for changing well locations, and thereby, try to complete the PMOR workflow for optimization problems during field development.

6.3 Suggestions for future work

Future research work in the areas focused in this dissertation can include the following ideas:

- In PMOR development for well control optimization, we considered a global PMOR method that aims at constructing a global ROB from representative controls, which

can be used for all the BHP profiles during entire optimization run. However, such techniques lead to higher dimensional basis as we add more BHP profiles during the training procedure. One may look for strategies to construct local ROBs in the parameter space for POD to achieve even more speedups. One of the ways to address this can be using QoI error prediction as discussed in Section 4 for changing well locations. The same strategy can also be used considering polynomial coefficients as the system parameter to represent well BHP profiles. DEIM used here for fast computation of non-linear functions here requires large number of selected cells by greedy algorithms since we use the same global DEIM basis for all the BHP profiles. Thus, similar to POD basis, it may lead to significant computational speedups by choosing local DEIM basis in the parameter domain. Also, one may look into time varying DEIM selected cells as. Thus, for a given time interval, a very few gridblocks may be needed to interpolate non-linear terms in reservoir simulation and thereby achieving significant computational advantage. Since POD-DEIM is a very intrusive method, one should also look at the non-intrusive Machine Learning based PMOR method proposed in 5 for well control optimization that has the potential to achieve significantly high speedups in the online stage.

- The preliminary work done in Bayesian formulation for selecting basis functions show promising results with the proposed idea, but, the posterior formulation is based on residual calculations at each time step, which for reservoir simulation is non-linear, and hence spends a lot of time effort in sampling the posterior. Thus, one should devise a faster approximation of residual for practical feasibility of the proposed method. Also, one may think of efficient sampling strategies instead of Gibbs sampling for faster computation such as two-stage MCMC [109].

- The local and global PMOR techniques developed for well location changes rely on Machine Learning methods that require efficient sampling procedures in the parameter

space such that it is representative of the new parameters during testing phase. We use random sampling here which may lead to a large number of unwanted samples. Thus, it is of great interest for one to consider fast and reliable sampling methods for practical implementation of these workflows for well location optimization. As mentioned before, the local PMOR technique based on error map construction, in its present form, is applicable for scenarios where we do not observe significant water fronts i.e. small simulation times. Considering the complexity of the problem, it is difficult for a basis from single well location to represent the physics of a new well location, especially for large water fronts. Hence it is worthwhile to consider local in time basis and concatenating snapshots from different well locations for future work for the proposed error map based formulation.

- Global PMOR method being able to tackle the limitations of local PMOR technique, should focus on reducing the computational effort required in training by efficient sampling procedures. One possible way is to sample the parameter space using experimental design for the most important features like TOF, well block permeability and LC. One can also consider adding more features to represent the well location parameter in addition to geometric and flow diagnostics based features suggested here. Extending this methodology to a full field implementation for multiple wells in the future, as suggested in 5, it may be required to use the most important features for well locations while predicting POD coefficients and then using the full feature set during the error correction procedure for computational savings. As per the ROM literature while using PMOR with optimization methods, it is necessary to tie the PMOR training procedures with strategy used for optimization. For example, it may not be required to collect well location training samples from certain parts of the reservoir deemed unimportant for well placement, which is set as a constraint in optimization routine. Hence, PMOR may be designed without the requirement of it being used for all the gridblocks of reservoir as potential well locations.

REFERENCES

[1] U.S. EIA, "Annual Energy Outlook 2019 with projections to 2050," *Annual Energy Outlook 2019 with projections to 2050*, vol. 44, no. 8, pp. 1–64, 2019.

[2] J.-D. Jansen, R. Brouwer, and S. G. Douma, "Closed Loop Reservoir Management," in *SPE Reservoir Simulation Symposium*, 2009.

[3] M. Christie, "Upscaling for Reservoir Simulation," *Journal of Petroleum Technology*, vol. 48, no. 11, pp. 1004–1010, 2014.

[4] M. J. King, D. G. Macdonald, S. P. Todd, and H. Leung, "SPE 50643-Application of Novel Upscaling Approaches to the Magnus and Andrew Reservoirs," in *European Petroleum Conference*, 1998.

[5] R. Wanderley de Holanda, E. Gildin, and J. L. Jensen, "A generalized framework for Capacitance Resistance Models and a Comparison with Streamline Allocation Factors," *Journal of Petroleum Science and Engineering*, vol. 162, no. November 2017, pp. 260–282, 2018.

[6] M. C. Kennedy and A. O. Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society*, vol. 63, no. Part 3, pp. 425–464, 2001.

[7] D. L. Knill, A. A. Giunta, C. A. Baker, B. Grossman, W. H. Mason, R. T. Haftka, and L. T. Watsonft, "Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design," *Journal of Aircraft*, vol. 36, no. 1, 1999.

[8] A. Antoulas, *Approximation of Large-scale Dynamical Systems*. SIAM Advances in Design and Control, 2005.

[9] M. Hinze and S. Volkwein, *Dimension Reduction of Large-Scale Systems*, vol. 45. 2005.

[10] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[11] E. Grimme, *Krylov Projection Methods for Model Reduction*. PhD thesis, 1997.

[12] A. Noor and J. Peters, "Reduced Basis Technique for Nonlinear Analysis of Structures," *AIAA Journal*, vol. 18, no. 4, pp. 455–462, 2008.

[13] A. Bao and E. Gildin, "Data-Driven Model Reduction Based on Sparsity-Promoting Methods for Multiphase Flow in Porous Media," in *SPE Latin America and Caribbean Petroleum Engineering Conference*, 2017.

[14] M. Ghommem, V. M. Calo, and Y. Efendiev, "Mode Decomposition Methods for Flows in High-Contrast Porous," *arXiv:1301.5735v1*, pp. 1–25, 2013.

[15] T. Lieu, C. Farhat, and M. Lesoinne, "Reduced-order fluid/structure modeling of a complete aircraft configuration," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 41, pp. 5730–5742, 2006.

[16] T. Lieu and C. Farhat, "Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration," *AIAA Journal*, vol. 45, no. 6, pp. 1244–1257, 2007.

[17] M. W. Hess and P. Benner, "A reduced basis method for microwave semiconductor devices with geometric variations," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 33, no. 4, pp. 1071–1081, 2014.

[18] L. Feng, D. Koziol, E. B. Rudnyi, and J. G. Korvink, "Parametric Model Reduction for Fast Simulation of Cyclic Voltammograms," *Sensor Letters*, vol. 4, no. 2, pp. 165–173.

[19] L. H. Feng, E. B. Rudnyi, and J. G. Korvink, "Preserving the film coefficient as a parameter in the compact thermal model for fast electrothermal simulation," *IEEE*

Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 12, pp. 1838–1847, 2005.

[20] M. Ghasemi, Y. Yang, E. Gildin, Y. Efendiev, and V. Calo, "Fast Multiscale Reservoir Simulations using POD-DEIM Model Reduction," in *SPE Reservoir Simulation Symposium*, 2015.

[21] Y. Choi, D. Amsallem, and C. Farhat, "Gradient-based Constrained Optimization Using a Database of Linear Reduced-Order Models," *arXiv preprint arXiv:1506.07849*, pp. 1–28, 2015.

[22] J. F. M. V. Doren and R. Markovinovic, "Reduced-order optimal control of water flooding using proper orthogonal decomposition," *Computational Geosciences*, vol. 10, pp. 137–158, 2006.

[23] M. A. Cardoso, L. J. Durlofsky, and P. Sarma, "Development and application of reduced-order modeling procedures for subsurface flow simulation," *International Journal for Numerical Methods in Engineering*, pp. 1322–1350, 2008.

[24] M. A. Cardoso and L. J. Durlofsky, "Use of Reduced-Order Modeling Procedures for Production Optimization," in *SPE Reservoir Simulation Symposium*, no. February, pp. 2–4, 2009.

[25] Y. J. Yang and K. Y. Shen, "Nonlinear heat-transfer macromodeling for MEMS thermal devices," *Journal of Micromechanics and Microengineering*, vol. 15, no. 2, pp. 408–418, 2005.

[26] B. Bond and L. Daniel, "Parameterized Model Order Reduction of Nonlinear Dynamical Systems," in *IEEE/ACM International Conference on Computer-Aided Design*, 2005.

[27] D. Vasilyev, M. Rewienski, and J. White, "A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS," no. 1, p. 490, 2003.

[28] M. Rewieński 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*, vol. 22, no. 2, pp. 155–170, 2003.

[29] J. He, J. Sætrom, and L. J. Durlofsky, "Enhanced linearized reduced-order models for subsurface flow simulation," *Journal of Computational Physics*, vol. 230, no. 23, pp. 8313–8341, 2011.

[30] S. Trehan and L. J. Durlofsky, "Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to PDE-constrained optimization," *Journal of Computational Physics*, vol. 326, pp. 446–473, 2016.

[31] S. Trehan, L. Durlofsky, and K. Carlberg, "Error Estimation for Surrogate Models of Dynamical Systems Using Machine Learning," *International Journal for Numerical Methods in Engineering*, vol. 00, pp. 1–31, 2016.

[32] S. Chaturantabut and D. Sorensen, "Nonlinear Model Reduction via Discrete Empirical Interpolation," *SIAM Journal on Scientific Computing*, vol. 32, pp. 2737–2764, jan 2010.

[33] R. Ştefânescu and I. M. Navon, "POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model," *Journal of Computational Physics*, vol. 237, pp. 95–114, 2013.

[34] H. Antil, M. Heinkenschloss, and D. C. Sorensen, "Application of the Discrete Empirical Interpolation Method to Reduced Order Modeling of Nonlinear and Parametric Systems BT - Reduced Order Methods for Modeling and Computational Reduction," pp. 101–136, Cham: Springer International Publishing, 2014.

[35] E. Gildin, M. Ghasemi, A. Protasov, and Y. Efendiev, "SPE 163618 Nonlinear Complexity Reduction for Fast Simulation of Flow in Heterogenous Porous Media," in *Reservoir Simulation Symposium*, 2013.

[36] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz, "Localized Discrete Empirical Interpolation Method," *SIAM Journal on Scientific Computing*, vol. 36, pp. A168–A192, jan 2014.

[37] M. Ghasemi, *Model Order Reduction in Porous Media Flow Simulation and Optimization*. PhD thesis, 2015.

[38] X. Tan, E. Gildin, S. Trehan, Y. Yang, and N. Hoda, "Trajectory-Based DEIM TDEIM Model Reduction Applied to Reservoir Simulation," in *SPE Reservoir Simulation Conference*, 2017.

[39] M. Ghasemi, A. Ibrahim, and E. Gildin, "Reduced Order Modeling In Reservoir Simulation Using the Bilinear Approximation Techniques," in *SPE Latin America and Caribbean Petroleum Engineering Conference*, 2014.

[40] M. Ghasemi and E. Gildin, "Model Order Reduction in Porous Media Flow Simulation using Quadratic Bilinear Formulation," *Computational Geosciences*, vol. 20, no. 3, pp. 723–735, 2016.

[41] R. Jiang and L. J. Durlofsky, "Implementation and detailed assessment of a GNAT reduced-order model for subsurface flow simulation," *Journal of Computational Physics*, vol. 379, pp. 192–213, 2019.

[42] M. A. Cardoso and L. J. Durlofsky, "Use of Reduced-Order Modeling Procedures for Production Optimization," *SPE Journal*, vol. 15, no. 2, pp. 426–435, 2010.

[43] J. S. Hesthaven, B. Stamm, and S. Zhang, "Efficient Greedy Algorithms for High-Dimensional Parameter Spaces with Applications to Empirical Interpolation and Reduced Basis Methods," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 48, no. 1, pp. 259–283, 2014.

[44] A. Centilmen, T. Ertekin, and A. S. Grader, "SPE 56433 Applications of Neural Networks in Multiwell Field Development," in *SPE Annual Technical Conference & Exhibition*, 1999.

[45] S. V. Taware, H.-y. Park, A. Datta-Gupta, S. Bhattacharya, A. Tomar, M. Kumar, and H. Rao, "Well Placement Optimization in a Mature Carbonate Waterflood using Streamline-based Quality Maps," in *SPE Oil and Gas India Conference and Exhibition, 28-30 March, Mumbai, India*, 2012.

[46] D. Amsallem and C. Farhat, "An Online Method for Interpolating Linear Parametric Reduced-Order Models," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2169–2198, 2011.

[47] D. Amsallem, J. Cortial, and C. Farhat, "On-Demand CFD-Based Aeroelastic Predictions Using a Database of Reduced-Order Bases and Models," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, no. January, pp. Orlando, Florida, 2009.

[48] A. A. Shah, W. W. Xing, and V. Triantafyllidis, "Reduced-order modelling of parameter-dependent , linear and nonlinear dynamic partial differential equation models Subject Areas :," *The Royal Society Publishing*, no. i, 2017.

[49] M. Drohmann and K. Carlberg, "The ROMES Method for Statistical Modeling of Reduced-Order-Model Error," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3, pp. 116–145, jan 2015.

[50] M. Eldred, A. Giunta, and S. Collis, "Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Multidisciplinary Analysis Optimization Conferences, American Institute of Aeronautics and Astronautics, aug 2004.

[51] S. E. Gano, J. E. Renaud, and B. Sanders, "Hybrid Variable Fidelity Optimization by Using a Kriging-Based Scaling Function," *AIAA Journal*, vol. 43, pp. 2422–2433, nov 2005.

[52] A. Moosavi, R. Ştefănescu, and A. Sandu, "Multivariate Predictions of Local Reduced-Order-Model Errors and Dimensions," *International Journal for Numerical Methods in*

Engineering, vol. 113, no. 3, pp. 512–533, 2018.

[53] F. Casenave, A. Ern, and T. Lelièvre, "A nonintrusive reduced basis method applied to aeroacoustic simulations," *Advances in Computational Mathematics*, vol. 41, no. 5, pp. 961–986, 2015.

[54] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[55] J. S. Hesthaven and S. Ubbiali, "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *Journal of Computational Physics*, vol. 363, pp. 55–78, 2018.

[56] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, "Projection-based model reduction: Formulations for physics-based machine learning," *Computers and Fluids*, vol. 0, pp. 1–14, 2018.

[57] AZIZ and K., "Petroleum Reservoir Simulation," *Applied Science Publishers*, vol. 476, 1979.

[58] T. Ertekin, J. H. Abou-Kassem, and G. R. King, *Basic Applied Reservoir Simulation*. Richardson, Tex.: Society of Petroleum Engineers, 2001.

[59] Z. Chen, G. Huan, and Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics, jan 2006.

[60] K.-a. Lie, "An Introduction to Reservoir Simulation Using MATLAB / GNU Octave User Guide for the Matlab Reservoir Simulation," 2018.

[61] A. Alhuthali, A. Oyerinde, and A. Datta-Gupta, "Optimal Waterflood Management Using Rate Control," *SPE Reservoir Evaluation & Engineering*, vol. 10, no. 05, pp. 539–551, 2007.

[62] J.-D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media - A review*, vol. 46. jul 2011.

[63] T. Kailath, *Linear systems*. Englewood Cliffs, N.J.: Prentice-Hall, 1980.

[64] M. J. Zandvliet, J. F. M. Van Doren, O. H. Bosgra, J. D. Jansen, and P. M. J. Van den Hof, "Controllability, observability and identifiability in single-phase porous media flow," *Computational Geosciences*, vol. 12, no. 4, pp. 605–622, 2008.

[65] J. F. M. Van Doren, P. M. J. Van den Hof, O. H. Bosgra, and J. D. Jansen, "Controllability and observability in two-phase porous media flow," *Computational Geosciences*, vol. 17, no. 5, pp. 773–788, 2013.

[66] S. Lall, J. E. Marsden, and S. Glavaški, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 6, pp. 519–535, 2002.

[67] J. F. V. Ren, *Model Structure Analysis for Model-based Operation of Petroleum Reservoirs*. 2010.

[68] K. Carlberg, M. Barone, and H. Antil, "Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction," *Journal of Computational Physics*, vol. 330, pp. 693–734, 2017.

[69] J. He and L. J. Durlofsky, "Constraint reduction procedures for reduced-order subsurface flow models based on POD–TPWL," *International Journal for Numerical Methods in Engineering*, vol. 103, pp. 1–30, jul 2015.

[70] A. A. Awotunde, "On The Joint Optimization of Well Placement and Control," 2014.

[71] A. Codas, B. Foss, and E. Camponogara, "Output-Constraint Handling and Parallelization for Oil-Reservoir Control Optimization by Means of Multiple Shooting," *SPE Journal*, vol. 20, no. 04, pp. 856–871, 2015.

[72] M. J. Zandvliet, O. H. Bosgra, J. D. Jansen, P. M. J. Van den Hof, and J. Kraaijevanger, "Bang-bang control and singular arcs in reservoir flooding," *Journal of Petroleum Science and Engineering*, vol. 58, no. 1, pp. 186–200, 2007.

[73] P. Sarma, L. J. Durlofsky, K. Aziz, and W. H. Chen, "Efficient real-time reservoir management using adjoint-based optimal control and model updating," *Computational Geosciences*, vol. 10, no. 1, pp. 3–36, 2006.

[74] B. Houska and B. Chachuat, "Global optimization in Hilbert space," *Mathematical Programming*, vol. 173, no. 1-2, pp. 221–249, 2015.

[75] N. Sorek, E. Gildin, F. Boukouvala, B. Beykal, and C. A. Floudas, "Dimensionality reduction for production optimization using polynomial approximations," *Computational Geosciences*, vol. 21, no. 2, pp. 247–266, 2017.

[76] B. Beykal, F. Boukouvala, C. A. Floudas, N. Sorek, H. Zalavadia, and E. Gildin, "Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations," *Computers & Chemical Engineering*, vol. 114, pp. 99–110, 2018.

[77] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*. Pacific Grove, CA, USA: Brooks/Cole Publishing Co., 1991.

[78] O. Devolder, F. Glineur, and Y. Nesterov, "Solving Infinite-dimensional Optimization Problems by Polynomial Approximation BT - Recent Advances in Optimization and its Applications in Engineering," (Berlin, Heidelberg), pp. 31–40, Springer Berlin Heidelberg, 2010.

[79] B. Houska and B. Chachuat, "Branch-and-Lift Algorithm for Deterministic Global Optimization in Nonlinear Optimal Control," *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 208–248, 2014.

[80] G. K. Smyth, "Polynomial Approximation," sep 2014.

[81] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers & Chemical Engineering*, vol. 8, no. 3, pp. 243–247, 1984.

[82] T. Hirmajer, M. Fikar, E. Balsa-Canto, and J. Banga, *Application of a control vector parameterization method using an interior point algorithm.* jan 2008.

[83] J. Vlassenbroeck, "A chebyshev polynomial method for optimal control with state constraints," *Automatica*, vol. 24, no. 4, pp. 499–506, 1988.

[84] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, "Dynamic optimization using adaptive control vector parameterization," *Computers & Chemical Engineering*, vol. 29, no. 8, pp. 1731–1751, 2005.

[85] A. Ghosh, A. Chowdhury, R. Giri, S. Das, and A. Abraham, "A hybrid evolutionary direct search technique for solving Optimal Control problems," in *2010 10th International Conference on Hybrid Intelligent Systems*, pp. 125–130, 2010.

[86] C. de Boor, *A Practical Guide to Spline*, vol. Volume 27. jan 1978.

[87] D. F. Oliveira and A. Reynolds, "An Adaptive Hierarchical Algorithm for Estimation of Optimal Well Controls," 2013.

[88] MathWorks, "Constrained Nonlinear Optimization Algorithms - MATLAB; Simulink," 2019.

[89] MathWorks, "Particle Swarm Optimization Algorithm - MATLAB; Simulink," 2019.

[90] D. Echeverría Ciaurri, O. J. Isebor, and L. J. Durlofsky, "Application of derivative-free methodologies to generally constrained oil production optimization problems," *Procedia Computer Science*, vol. 1, no. 1, pp. 1301–1310, 2010.

[91] D. Amsallem, M. Zahr, Y. Choi, and C. Farhat, "Design optimization using hyper-reduced-order models," *Structural and Multidisciplinary Optimization*, vol. 51, no. 4, pp. 919–940, 2015.

[92] T. Bui-Thanh, K. Willcox, and O. Ghattas, "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space," *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3270–3288, 2008.

[93] Y. Efendiev, W. Leung, S. W. Cheung, N. Guha, V. H. Hoang, and B. Mallickk, "Bayesian Multiscale Finite Element Methods. Modeling missing subgrid information probabilistically," *International Journal for Multiscale Computational Engineering*, vol. 15, no. 2, pp. 175—-197, 2017.

[94] H. Zalavadia and E. Gildin, "Th A2 08 Parametric Model Order Reduction For Adaptive Basis Selection Using Machine Learning Techniques During Well Location Opt," *15th European Conference on the Mathematics of Oil Recovery (ECMOR XV)*, 2018.

[95] A. Bittencourt and R. Horne, "SPE 38895-MS Reservoir Development and Design Optimization," in *SPE Annual Technical Conference & Exhibition*, 1997.

[96] B. L. Beckner and X. Song, "SPE 30650 Field Development Planning Using Simulated Annealing - Optimal Economic Well Scheduling and Placement," in *SPE Annual Technical Conference & Exhibition*, 1995.

[97] P. Sarma and W. H. Chen, "SPE 112257 Efficient Well Placement Optimization with Gradient - Based Algorithms and Adjoint Models," in *SPE Intelligent Energy Conference and Exhibition*, 2008.

[98] F. Forouzanfar, A. C. Reynolds, and G. Li, "Optimization of the well locations and completions for vertical and horizontal wells using a derivative-free optimization algorithm," *Journal of Petroleum Science and Engineering*, vol. 86, pp. 272–288, 2012.

[99] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[100] L. E. O. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[101] A. Datta-Gupta and M. J. King, *Streamline simulation : theory and practice*. Richardson, TX: Society of Petroleum Engineers, 2007.

[102] A. Datta-Gupta and M. J. King, "A semianalytic approach to tracer flow modeling in heterogeneous permeable media," *Advances in Water Resources*, vol. 18, no. 1, pp. 9–24, 1995.

[103] M. Shahvali, B. Mallison, K. Wei, H. Gross, C. Energy, and T. Company, "An Alternative to Streamlines for Flow Diagnostics on Structured and Unstructured Grids," *SPE Journal*, no. September, pp. 768–778, 2012.

[104] O. Møyner, S. Krogstad, and K.-A. Lie, "The Application of Flow Diagnostics for Reservoir Management," *SPE Journal*, vol. 20, no. 02, pp. 306–323, 2015.

[105] J. R. Natvig and K.-A. Lie, "Fast Computation of Multiphase Flow in Porous Media by Implicit Discontinuous Galerkin Schemes With Optimal Ordering of Elements," *Journal of Computational Physics*, vol. 227, no. 24, pp. 10108–10124, 2008.

[106] G. M. Shook and K. M. Mitchell, "A Robust Measure of Heterogeneity for Ranking Earth Models: The F PHI Curve and Dynamic Lorenz Coefficient," 2009.

[107] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[108] N. Sorek, H. Zalavadia, and E. Gildin, "SPE-182652-MS Model Order Reduction and Control Polynomial Approximation for Well-Control Production Optimization," in *SPE Reservoir Simulation Conference*, 2017.

[109] Y. Efendiev, A. Datta-Gupta, V. Ginting, X. Ma, and B. Mallick, "An efficient two-stage Markov chain Monte Carlo method for dynamic data integration," *Water Resources Research*, vol. 41, no. 12, pp. 1–6, 2005.